SIEMENS

VBS pour la création de procédures et d'actions 1

ANSI-C pour la création de fonctions et actions 2

VBA pour la configuration automatisée 3

SIMATIC HMI

WinCC V7.2
WinCC : Scripting (VBS, ANSI-C, VBA)

Manuel système

Impression de l'aide en ligne

02/2013
A5E32315922-AA

Mentions légales

Signalétique d'avertissement

Ce manuel donne des consignes que vous devez respecter pour votre propre sécurité et pour éviter des dommages matériels. Les avertissements servant à votre sécurité personnelle sont accompagnés d'un triangle de danger, les avertissements concernant uniquement des dommages matériels sont dépourvus de ce triangle. Les avertissements sont représentés ci-après par ordre décroissant de niveau de risque.

⚠ DANGER
signifie que la non-application des mesures de sécurité appropriées **entraîne** la mort ou des blessures graves.

⚠ ATTENTION
signifie que la non-application des mesures de sécurité appropriées **peut entraîner** la mort ou des blessures graves.

⚠ PRUDENCE
signifie que la non-application des mesures de sécurité appropriées peut entraîner des blessures légères.

IMPORTANT
signifie que la non-application des mesures de sécurité appropriées peut entraîner un dommage matériel.

En présence de plusieurs niveaux de risque, c'est toujours l'avertissement correspondant au niveau le plus élevé qui est reproduit. Si un avertissement avec triangle de danger prévient des risques de dommages corporels, le même avertissement peut aussi contenir un avis de mise en garde contre des dommages matériels.

Personnes qualifiées

L'appareil/le système décrit dans cette documentation ne doit être manipulé que par du **personnel qualifié** pour chaque tâche spécifique. La documentation relative à cette tâche doit être observée, en particulier les consignes de sécurité et avertissements. Les personnes qualifiées sont, en raison de leur formation et de leur expérience, en mesure de reconnaître les risques liés au maniement de ce produit / système et de les éviter.

Utilisation des produits Siemens conforme à leur destination

Tenez compte des points suivants:

⚠ ATTENTION
Les produits Siemens ne doivent être utilisés que pour les cas d'application prévus dans le catalogue et dans la documentation technique correspondante. S'ils sont utilisés en liaison avec des produits et composants d'autres marques, ceux-ci doivent être recommandés ou agréés par Siemens. Le fonctionnement correct et sûr des produits suppose un transport, un entreposage, une mise en place, un montage, une mise en service, une utilisation et une maintenance dans les règles de l'art. Il faut respecter les conditions d'environnement admissibles ainsi que les indications dans les documentations afférentes.

Marques de fabrique

Toutes les désignations repérées par ® sont des marques déposées de Siemens AG. Les autres désignations dans ce document peuvent être des marques dont l'utilisation par des tiers à leurs propres fins peut enfreindre les droits de leurs propriétaires respectifs.

Exclusion de responsabilité

Nous avons vérifié la conformité du contenu du présent document avec le matériel et le logiciel qui y sont décrits. Ne pouvant toutefois exclure toute divergence, nous ne pouvons pas nous porter garants de la conformité intégrale. Si l'usage de ce manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
ALLEMAGNE

A5E32315922-AA
℗ 02/2013 Sous réserve de modifications techniques

Copyright © Siemens AG 2013.
Tous droits réservés

Sommaire

1 VBS pour la création de procédures et d'actions....11

1.1 VBS pour la création de procédures et d'actions....11

1.2 Utilisation de Visual Basic Script dans WinCC....12

1.3 Modules et procédures....16

1.4 Actions....19

1.5 Procédez comme suit pour utiliser plusieurs fois des procédures et des actions....22

1.6 Interactions avec CrossReference....24

1.7 Utilisation de variables globales dans VBS....26

1.8 Les éditeurs VBScript....28
1.8.1 Les éditeurs VBScript....28
1.8.2 L'éditeur Global Script....29
1.8.3 Travailler dans la fenêtre d'édition....32
1.8.4 Utiliser les barres d'outils....35
1.8.5 Procédez comme suit pour supprimer des actions ou des procédures....38

1.9 Création et édition de procédures....40
1.9.1 Création et édition de procédures....40
1.9.2 Pour créer une nouvelle procédure....44
1.9.3 Pour écrire le code d'une procédure....45
1.9.4 Pour utiliser des procédures de projet et standard....48
1.9.5 Pour ajouter des informations de module supplémentaires....49
1.9.6 Pour protéger un module par un mot de passe....51
1.9.7 Pour enregistrer une procédure....53
1.9.8 Pour renommer une procédure ou un module....55

1.10 Création et édition d'actions....57
1.10.1 Création et édition d'actions....57
1.10.2 Pour créer une nouvelle action....62
1.10.3 Pour éditer une action....63
1.10.4 Pour ajouter des informations supplémentaires à l'action....66
1.10.5 Pour protéger une action par un mot de passe....68
1.10.6 Pour enregistrer une action....69
1.10.7 Déclenchement....71
1.10.7.1 Déclenchement....71
1.10.7.2 Déclenchement d'animation....75
1.10.7.3 Pour ajouter un Trigger du type "Temporisateur"....76
1.10.7.4 Pour ajouter un Trigger du type "Variable"....77
1.10.7.5 Pour modifier un déclenchement....78
1.10.7.6 Pour supprimer un déclenchement....79
1.10.8 Pour renommer une action....80

1.11 Pour activer des actions globales dans Runtime....82

1.12 Diagnostic....83

1.12.1 Diagnostic....83
1.12.2 Diagnostic GSC....84
1.12.2.1 Diagnostic GSC....84
1.12.2.2 Pour insérer dans une vue la fenêtre de diagnostic GSC....85
1.12.2.3 Les attributs de Diagnostic GSC....85
1.12.2.4 La barre d'outils de Diagnostic GSC....86
1.12.3 GSC Runtime....87
1.12.3.1 GSC Runtime....87
1.12.3.2 Pour insérer dans une image la fenêtre GSC Runtime....88
1.12.3.3 Les attributs de GSC Runtime....89
1.12.4 Tester avec Debugger....90
1.12.4.1 Tester avec Debugger....90
1.12.4.2 Pour activer le débogueur....91
1.12.4.3 Principes de base du débogage....93
1.12.4.4 Composants du Microsoft Script Debugger....95
1.12.4.5 Structure des fichiers VBScript....97
1.12.4.6 Nom de l'action et de la procédure dans Debugger....99
1.12.4.7 Pour sélectionner un Script en vue de l'éditer....101
1.12.4.8 Pour traiter les Scripts étape par étape....102
1.12.4.9 Pour définir des points d'arrêt....103
1.12.4.10 Pour supprimer des points d'arrêt....105
1.12.4.11 Pour activer des signets dans le Script....105
1.12.4.12 Pour déterminer et modifier des valeurs de variable ou de propriété....106
1.12.4.13 Pour exécuter des instructions Script....107

1.13 Pour imprimer des VB-Scripts....109

1.14 Référence VBS....110
1.14.1 Référence VBS....110
1.14.2 Objets et listes....114
1.14.2.1 Objets et listes....114
1.14.2.2 Objet d'alarme....117
1.14.2.3 Objet d'alarmes (liste)....118
1.14.2.4 Objet AlarmLogs....119
1.14.2.5 Objet DataItem....120
1.14.2.6 Objet DataLogs....122
1.14.2.7 Objet DataSet (Liste)....123
1.14.2.8 Objet HMIRuntime....125
1.14.2.9 Objet Item....126
1.14.2.10 Objet Layer....127
1.14.2.11 Objet Layers (Liste)....128
1.14.2.12 Objet Logging....129
1.14.2.13 Objet ProcessValue....130
1.14.2.14 Objet ProcessValues (liste)....131
1.14.2.15 Objet Project....131
1.14.2.16 Objet ScreenItem....132
1.14.2.17 Objet ScreenItems (Liste)....136
1.14.2.18 Objet Screen....138
1.14.2.19 Objet Screens (Liste)....141
1.14.2.20 Objet SmartTags....143
1.14.2.21 Objet Tag....144
1.14.2.22 Objet Tags (Liste)....147
1.14.2.23 Objet TagSet (Liste)....149

1.14.3 Type d'objet de l'objet ScreenItem....150
1.14.3.1 Type d'objet de l'objet ScreenItem....150
1.14.3.2 Objets standard....151
1.14.3.3 Objets complexes....179
1.14.3.4 Objets Windows....210
1.14.3.5 Objets tuyaux....222
1.14.3.6 Contrôles....224
1.14.3.7 Objet utilisateur....290
1.14.3.8 Groupe....291
1.14.4 Propriétés....292
1.14.4.1 Propriétés....292
1.14.4.2 A....293
1.14.4.3 B....312
1.14.4.4 C....339
1.14.4.5 D....372
1.14.4.6 e....381
1.14.4.7 f....391
1.14.4.8 G....412
1.14.4.9 h....416
1.14.4.10 I....424
1.14.4.11 L....430
1.14.4.12 M....461
1.14.4.13 N....482
1.14.4.14 O....484
1.14.4.15 P....501
1.14.4.16 Q....518
1.14.4.17 R....519
1.14.4.18 S....530
1.14.4.19 T....567
1.14.4.20 U....643
1.14.4.21 V....652
1.14.4.22 W....668
1.14.4.23 X - Z....672
1.14.5 Méthodes....680
1.14.5.1 Méthodes....680
1.14.5.2 Méthodes A à E....682
1.14.5.3 Méthodes Get....690
1.14.5.4 Méthodes H à M....738
1.14.5.5 Méthodes N à R....749
1.14.5.6 Méthodes S à T....767
1.14.5.7 Méthodes U à Z....781
1.14.6 Annexe....790
1.14.6.1 Messages d'erreur dans la zone Bases de données....790

1.15 Exemples pour VBScript....792
1.15.1 Exemples pour VBScript....792
1.15.2 Exemples dans WinCC....792
1.15.2.1 Exemples dans WinCC....792
1.15.2.2 Exemple : Pour accéder à des objets dans Graphics Designer....793
1.15.2.3 Exemple : Comment déterminer la couleur des objets....794
1.15.2.4 Exemple : Pour configurer un changement de langue....794
1.15.2.5 Exemple : Comment désactiver Runtime....795
1.15.2.6 Exemple : Comment configurer globalement un changement de vue....795

1.15.2.7 Exemple : Comment configurer un changement de vue via Property....796
1.15.2.8 Exemple : Comment configurer l'affichage d'un diagnostic via Trace....796
1.15.2.9 Exemple : Pour écrire les valeurs variable....797
1.15.2.10 Exemple : Pour lire les valeurs variable....799
1.15.2.11 Exemple : Comment écrire les propriétés de l'objet....802
1.15.2.12 Exemple : Pour démarrer une action sur le serveur (objet Logging)....805
1.15.2.13 Dynamisation des contrôles....807
1.15.3 Exemples, généralités....819
1.15.3.1 Exemples d'ordre général pour VBScript....819
1.15.3.2 Exemple : Comment configurer une intégration de la base de données avec VBS....820
1.15.3.3 Exemple : Comment utiliser l'interface MS Automation....822
1.15.3.4 Exemple : Pour démarrer une application tierce....824

1.16 Principes de base de VBScript....825
1.16.1 Principes de base de VBScript....825
1.16.2 VBScript Principes de base....825

2 ANSI-C pour la création de fonctions et actions....827

2.1 Création de fonctions et actions avec ANSI-C....827

2.2 Création de fonctions et d'actions....828

2.3 Caractéristiques de fonctions de projet....831

2.4 Caractéristiques de fonctions standard....832

2.5 Caractéristiques de fonctions internes....834

2.6 Caractéristiques d'actions locales....835

2.7 Caractéristiques d'actions globales....836

2.8 Pour ajouter Global Script Runtime à la liste de démarrage du projet....837

2.9 Utilisation de variables C globales....838

2.10 Utilisation de DLL dans des fonctions et actions....840

2.11 L'éditeur Global Script....842
2.11.1 L'éditeur Global Script....842
2.11.2 Utiliser la fenêtre d'édition....844
2.11.2.1 Utiliser la fenêtre d'édition....844
2.11.2.2 Fonctions d'édition avec le clavier....845
2.11.2.3 Fonctions d'édition avec la souris....846
2.11.3 Utiliser les barres d'outils....847
2.11.4 Pour paramétrer plusieurs vues....849
2.11.5 Pour paramétrer la police....850
2.11.6 Pour utiliser "Enregistrer sous..."....850
2.11.7 Pour supprimer des actions ou des fonctions de projet ou standard....851
2.11.8 Pour générer l'en-tête....852
2.11.9 Pour compiler toutes les fonctions....853
2.11.10 Pour rechercher un texte dans des fichiers....854
2.11.11 Imprimer des fonctions et actions....855
2.11.11.1 Imprimer des fonctions et actions....855
2.11.11.2 Pour définir les paramètres d'impression....855
2.11.11.3 Pour ouvrir l'aperçu avant impression....856
2.11.11.4 Pour imprimer une documentation de projet....856

2.12 Créer et éditer fonctions....857
2.12.1 Créer et éditer fonctions....857
2.12.2 Pour créer une nouvelle fonction....860
2.12.3 Pour écrire le code d'une fonction....861
2.12.4 Pour utiliser des fonctions internes....862
2.12.5 Pour utiliser des fonctions standard et des fonctions de projet....863
2.12.6 Pour ajouter des informations supplémentaires à la fonction....864
2.12.7 Pour protéger une fonction contre les modifications et l'affichage....864
2.12.8 Pour compiler et enregistrer une fonction....865
2.12.9 Pour renommer une fonction....867
2.12.10 Pour utiliser des fonctions empruntées à d'autres sources....868

2.13 Création et édition d'actions....870
2.13.1 Création et édition d'actions....870
2.13.2 Règle d'écriture du code de WinCC....873
2.13.3 Pour créer une nouvelle action....874
2.13.4 Pour éditer une action....875
2.13.5 Pour ajouter des informations supplémentaires à l'action....876
2.13.6 Pour protéger une action contre les modifications et l'affichage....876
2.13.7 Pour compiler et enregistrer une action....877
2.13.8 Déclenchement....879
2.13.8.1 Déclenchement....879
2.13.8.2 Pour ajouter un nouveau déclenchement du type "Temporisateur"....882
2.13.8.3 Pour ajouter un nouveau déclenchement du type "Variable"....883
2.13.8.4 Pour modifier un déclenchement....884
2.13.8.5 Pour supprimer un déclenchement....884
2.13.9 Pour attribuer une autorisation....885
2.13.10 Pour exporter une action....886
2.13.11 Pour importer une action....887
2.13.12 Pour renommer une action....887
2.13.13 Pour utiliser des actions étrangères au projet....888

2.14 Comportement des actions au Runtime....890
2.14.1 Comportement des actions au Runtime....890
2.14.2 GSC-Runtime....891
2.14.2.1 GSC-Runtime....891
2.14.2.2 Pour placer GSC-Runtime dans une vue de process....894
2.14.2.3 Les attributs de GSC-Runtime....895
2.14.2.4 Pour éditer une action....895
2.14.3 Diagnostic GSC....896
2.14.3.1 Diagnostic GSC....896
2.14.3.2 Pour placer Diagnostic GSC dans une vue de process....897
2.14.3.3 Les attributs de Diagnostic GSC....898
2.14.3.4 La barre d'outils du diagnostic GSC....898

2.15 ANSI-C Descriptions de fonctions....899
2.15.1 lpszPictureName....899
2.15.2 Fonctions standard....899
2.15.2.1 Fonctions standard - Description sommaire....899
2.15.2.2 Alarm....900
2.15.2.3 Graphics....905
2.15.2.4 Obsolete functions....910
2.15.2.5 Rapport....968

2.15.2.6 WinCC....970
2.15.2.7 Windows....977
2.15.3 Fonctions internes....978
2.15.3.1 Fonctions internes - Brève description....978
2.15.3.2 allocate....979
2.15.3.3 c_bib....980
2.15.3.4 graphics....1064
2.15.3.5 tag....1393
2.15.3.6 WinCC....1505
2.15.4 Exemples....1513
2.15.4.1 Exemples - A à G....1513
2.15.4.2 Exemples - GetAlarmHigh à GetPropChar....1516
2.15.4.3 Exemples - GetRangeMax à GetWidth....1532
2.15.4.4 Exemples - H à S....1552
2.15.4.5 Exemples - SetAlarmHigh à SetPropChar....1555
2.15.4.6 Exemples - SetRangeMax à SetWidth....1565
2.15.4.7 Exemples - T à Z....1575
2.15.4.8 Exemples de contrôles WinCC....1580
2.15.5 Listes....1582
2.15.5.1 Orientation du bargraphe....1582
2.15.5.2 Mise à l'échelle du bargraphe....1583
2.15.5.3 Fréquences de clignotement....1583
2.15.5.4 Champ d'E/S, Format de sortie....1583
2.15.5.5 Champ E/S, Type de données du contenu du champ....1585
2.15.5.6 Champ E/S, Type de champ....1585
2.15.5.7 Alignement d'élément dans des cases à cocher et des cases d'option....1585
2.15.5.8 Tableau de couleurs....1585
2.15.5.9 Descripteur de format....1586
2.15.5.10 Motif de remplissage....1587
2.15.5.11 Types de trait....1588
2.15.5.12 Fins de trait....1589
2.15.5.13 Types de liste....1589
2.15.5.14 Identificateurs de langue....1589
2.15.5.15 Sens du texte....1591
2.15.5.16 Etats de variable....1591
2.15.6 Définitions de structure....1592
2.15.6.1 Définition de structure CCAPErrorExecute....1592
2.15.6.2 Définition de structure CCAPTime....1593
2.15.6.3 Définition de structure CMN_ERROR....1594
2.15.6.4 Définition de structure DM_TYPEREF....1595
2.15.6.5 Définition de structure DM_VAR_UPDATE_STRUCT....1596
2.15.6.6 Définition de structure DM_VAR_UPDATE_STRUCTEX....1597
2.15.6.7 Définition de structure DM_VARKEY....1598
2.15.6.8 Définition de structure LINKINFO....1598
2.15.6.9 Définition de structure MSG_FILTER_STRUCT....1600
2.15.6.10 Définition de structure MSG_RTDATA_STRUCT....1603

3 VBA pour la configuration automatisée....1605

3.1 Automatisation de la configuration....1605

3.2 Introduction : Utilisation de VBA dans WinCC....1606
3.2.1 Introduction : Utilisation de VBA dans WinCC....1606
3.2.2 Différenciation : Utilisation de VBA....1606

3.2.3 Organisation du code VBA dans un projet WinCC....1607
3.2.4 Pour exporter ou importer un code VBA....1611
3.2.5 Exécution de macros VBA dans le Graphics Designer....1612

3.3 VBA dans le Graphics Designer....1614
3.3.1 VBA dans le Graphics Designer....1614
3.3.2 Adaptation du Graphics Designer avec VBA....1616
3.3.2.1 Adaptation du Graphics Designer avec VBA....1616
3.3.2.2 Configuration multilingue avec VBA....1617
3.3.2.3 Création de menus et de barres d'outils personnalisés....1619
3.3.2.4 Accès à la bibliothèque des modules avec VBA....1640
3.3.3 Edition de vues avec VBA....1647
3.3.3.1 Edition de vues avec VBA....1647
3.3.3.2 Pour créer des menus et des barres d'outils spécifiques à une vue....1648
3.3.3.3 Edition de niveaux avec VBA....1650
3.3.3.4 Edition d'une copie de vue avec VBA....1651
3.3.4 Edition d'objets avec VBA....1653
3.3.4.1 Edition d'objets avec VBA....1653
3.3.4.2 Objets standard, complexes, Windows et tuyaux....1655
3.3.4.3 Objets Groupe....1668
3.3.4.4 Objets utilisateur....1676
3.3.5 Création de dynamisations avec VBA....1680
3.3.5.1 Création de dynamisations avec VBA....1680
3.3.5.2 Dynamisation des propriétés d'une vue ou d'un objet....1682
3.3.5.3 Configuration d'actions commandées par événement avec VBA....1693
3.3.5.4 Edition de déclencheurs....1701
3.3.6 Gestion des événements....1704
3.3.7 Accès aux applications externes avec VBA....1707
3.3.7.1 Accès aux applications externes avec VBA....1707
3.3.7.2 Exemple : Accès à MS Excel avec VBA....1708

3.4 AddIns....1714
3.4.1 AddIns....1714
3.4.2 Ajouts d'addIns....1714
3.4.3 Pour charger un AddIn dans Graphics Designer....1716
3.4.4 Exemple : Création d'addins....1718
3.4.4.1 Exemple : Création d'addins....1718
3.4.4.2 Exemple : Créer un addin avec Visual Basic 6.0....1718

3.5 Référence VBA....1724
3.5.1 Le modèle d'objet du Graphics Designer....1724
3.5.1.1 Référence VBA....1724
3.5.1.2 Référence VBA : ActionDynamic....1726
3.5.1.3 Référence VBA : HMIObjects....1728
3.5.1.4 Référence VBA : Languages....1730
3.5.1.5 Evénements....1730
3.5.1.6 Méthodes....1765
3.5.1.7 Objets et listes....1867
3.5.1.8 Propriétés....2054
3.5.2 VBA dans d'autres éditeurs WinCC....2479
3.5.2.1 VBA dans d'autres éditeurs WinCC....2479
3.5.2.2 VBA dans la gestion des variables....2481
3.5.2.3 VBA dans Tag Logging....2492
3.5.2.4 VBA dans Text Library....2519

3.5.2.5 VBA dans Alarm Logging.....2532
Index.....2547

VBS pour la création de procédures et d'actions 1

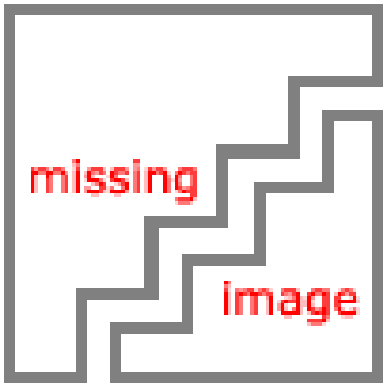
1.1 VBS pour la création de procédures et d'actions

Contenu

WIN CC vous permet de dynamiser votre environnement Runtime avec Visual Basic Script. VBS vous permet non seulement de programmer des actions et des procédures globales, mais également de dynamiser des objets graphiques en Runtime et de déclencher des actions.

Ce chapitre vous explique,

- Comment travailler avec les éditeurs VBScript
- Comment créer et éditer des procédures
- Créer et éditer des actions
- Comment activer des VBScript en Runtime
- Comment effectuer le diagnostic de vos scripts en Runtime
- Le modèle objet du système graphique Runtime
- Exemples détaillées concernant l'utilisation de VBScript



1.2 Utilisation de Visual Basic Script dans WinCC

Introduction

Dans WinCC, la langue de programmation VBScript est également disponible - en plus de C-Script - comme interface de programmation pour dynamiser l'environnement WinCC Runtime.

Groupe cible de la documentation

Cette documentation s'adresse aux configurateurs ayant des connaissances en Visual Basic ou WinCC Scriptings (C).

Possibilités d'emploi

VBScript (VBS) vous permet d'accéder en Runtime aux variables et aux objets du système graphique Runtime, mais aussi d'exécuter des actions ne dépendant pas des vues :

- Variables : Vous pouvez lire et écrire des valeurs variable pour par exemple proposer des valeurs variable pour l'automate, et ce par simple clic de souris sur un bouton.
- Objets : Vous pouvez dynamiser les propriétés de l'objet avec des actions et déclencher des actions via les événements pour objets.
- Actions ne dépendant pas de la vue : Vous pouvez déclencher des actions indépendamment des vues, de façon cyclique ou par le biais de valeurs variable, par exemple la transmission quotidienne de valeurs dans un tableau Excel.

Vous pouvez utiliser VBS aux endroits suivants dans WinCC :

- Dans le Global Script Editor : Vous configurez ici des actions et des actions ne dépendant pas des vues. Les procédures peuvent être utilisées dans des actions dépendant ou non des vues. Plusieurs procédures sont regroupées par thème dans un module.
- Dans Graphics Designer : Vous configurez ici les actions dépendant des vues, avec lesquelles vous dynamiser les propriétés des objets graphiques ou vous permettant de réagir aux événements en mode Runtime.
- Dans les barres d'outils et menus personnalisés : Vous configurez ici les procédures que vous appelez dans Runtime à l'aide des menus et des barres d'outils.

Remarque

Actualiser la configuration modifiée dans Runtime

Un script VB modifié et lié avec "menus et barres d'outils" est mis à jour seulement après un redémarrage de Runtime.

Si vous modifiez les propriétés de "Menus et barres d'outils" dans Runtime, les modifications ne sont acceptées que dans les cas suivants :

- Après un changement de vue, lorsque la modification de la configuration ne concerne pas la vue de base.
- Par le chargement d'un autre fichier de configuration et le nouveau chargement du fichier de configuration modifié.

Variables connectées dans les menus et les barres d'outils

Les variables connectées dans les scripts de "Menus et barres d'outils" restent connectées lors de la désélection de vue. Pour la lecture indirecte de la mémoire image de process, les variables sont connectées et après la désélection de vue de nouveau déconnectées. Dans les scripts de "Menus et barres d'outils", les variables restent toutefois connectées lors de la désélection de vue.

Scénarios d'application

VBS vous permet par exemple d'effectuer en Runtime les opérations suivantes :

- configurer des consignes allouées aux variables pour la commande d'un objet graphique, pour par exemple allouer une valeur à l'automate par simple clic de souris.
- configurer la permutation de la langue Runtime pour la commande d'un objet graphique.
- configurer le changement de couleur, de façon cyclique par exemple (clignotement) ou pour représenter des états (Moteur en marche).

Vous pouvez également utiliser, outre les applications spécifiques à WinCC, les fonctions VBS générales pour adapter votre Windows, par exemple :

- transférer des données dans d'autres applications (Excel p. ex.)
- lancer des applications externes depuis WinCC
- créer des fichiers et des dossiers.

Pour adapter votre environnement Windows, vous disposez des objets d'automation de votre environnement.

Remarque

Vous pouvez intégrer dans votre environnement tous les objets qui sont fournis avec Windows Script Host (WSH) de Microsoft, et ce par la méthode standard VBS CreateObject. De WinCC, VBS ne vous permet toutefois pas d'accéder directement à l'objet WSH.

Aucune garantie, ni WinCC-Support ne peuvent être apportés pour les fonctions VBS permettant d'adapter l'environnement Windows.

Distinction par rapport aux autres langages de programmation dans WinCC

VBS et C

Vous pouvez utiliser VBScript dans WinCC parallèlement à C-Script, sans toutefois pouvoir mélanger les types de script :

- Au sein d'un projet et d'une vue, vous pouvez configurer des scripts VB et des scripts C.
- Il n'est pas possible d'appeler des scripts C dans des scripts VB et vice-versa.
- Dans VBS, vous disposez d'interfaces internes pour les variables et les objets de vue, tandis que vous pouvez également accéder à d'autres sous-systèmes de WinCC (le système de journaux p. ex.) lorsque vous travaillez dans l'environnement C.

VBS et VBA

Utilisez VBA dans WinCC Configuration lors de configuration, pour adapter Graphics Designer à vos besoins personnels ainsi que pour simplifier et automatiser la configuration. Les programmes VBA ne fonctionnent que dans l'environnement de la configuration de WinCC.

Contrairement à VBA, les scripts VB ne fonctionnent qu'en WinCC Runtime en vous permettant d'accéder, là, aux objets graphiques et aux variables. VBS ne vous permet pas de créer ni de modifier de façon durable des objets ou des vues, alors que VBA vous le permet.

Les différences essentielles dans le language qui existent entre VBA et VBS sont par exemple :

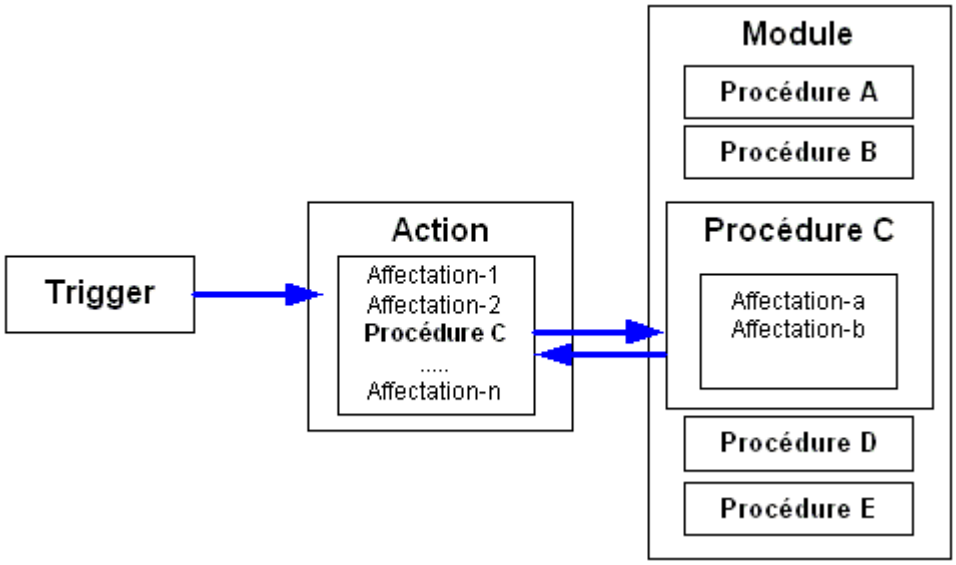
- VBS a été conçu pour l'utilisation dans Internet, VBA pour l'automatisation d'applications logicielles.
- Le type de données VBS-Variable est toujours VARIANT. VBA fait par contre une différence entre les types de données, à savoir INT, DOUBLE, STRING, etc.
- Certains éléments du language dans VBA ont été enlevés ou complétés dans VBS.
- La manipulation des erreurs dans VBS n'est pas résolue de la même manière que dans VBA.

Vous trouverez en annexe sous "Principes de base de VBScript" une liste des différences entre VBA et VBS.

Procédures, modules et actions

VBS dans WinCC vous permet d'utiliser des procédures, des modules et des actions permettant de dynamiser votre environnement Runtime :

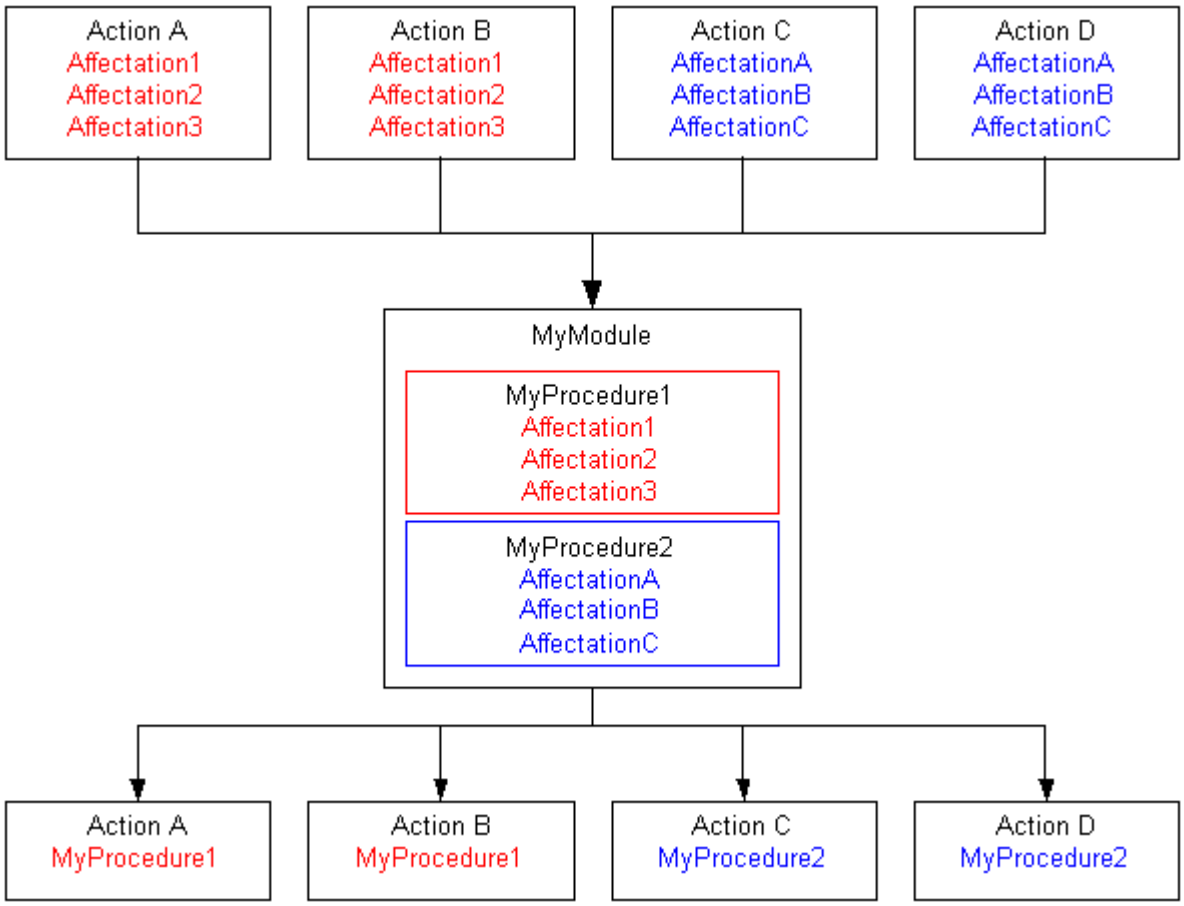
- Procédures : Une procédure correspond à une fonction en C. Inscrivez dans les procédures le code que vous voulez utiliser en plusieurs endroits de votre configuration. Vous appelez le code dans une action ou une autre procédure en affichant le nom de la procédure. Vous pouvez créer dans WinCC des procédures avec ou sans valeur de retour. Les procédures n'ont pas leur propre Trigger mais sont toujours appelées par une action.
- Modules : Les modules vous permettent de regrouper les procédures en unités significatives. Vous créez par exemple des modules pour des procédures qui sont utilisées dans une vue déterminée ou qui font partie d'une rubrique particulière, par exemple les fonctions auxiliaires mathématiques ou les fonctions d'accès à la banque de données.
- Actions : Les actions sont toujours déclenchées par un Trigger. Configurez les actions pour des propriétés d'objets graphiques, pour des événements qui interviennent pour un objet graphique ou bien globalement dans le projet. Dans les actions, vous pouvez appeler sous forme de procédures un code utilisé plusieurs fois.


Module
Procédure A
Procédure B
Procédure C
Affectation-a
Affectation-b
Procédure D
Procédure E
Action
Affectation-1
Affectation-2
Procédure C
.....
Affectation-n
Trigger

1.3 Modules et procédures

Introduction

Utilisez les procédures pour rendre un code disponible à plusieurs endroits de votre projet, mais en ne le créant qu'une seule fois. Appelez tout simplement la procédure en question au lieu de taper le code plusieurs fois. Votre code est plus simple et plus facile à gérer.



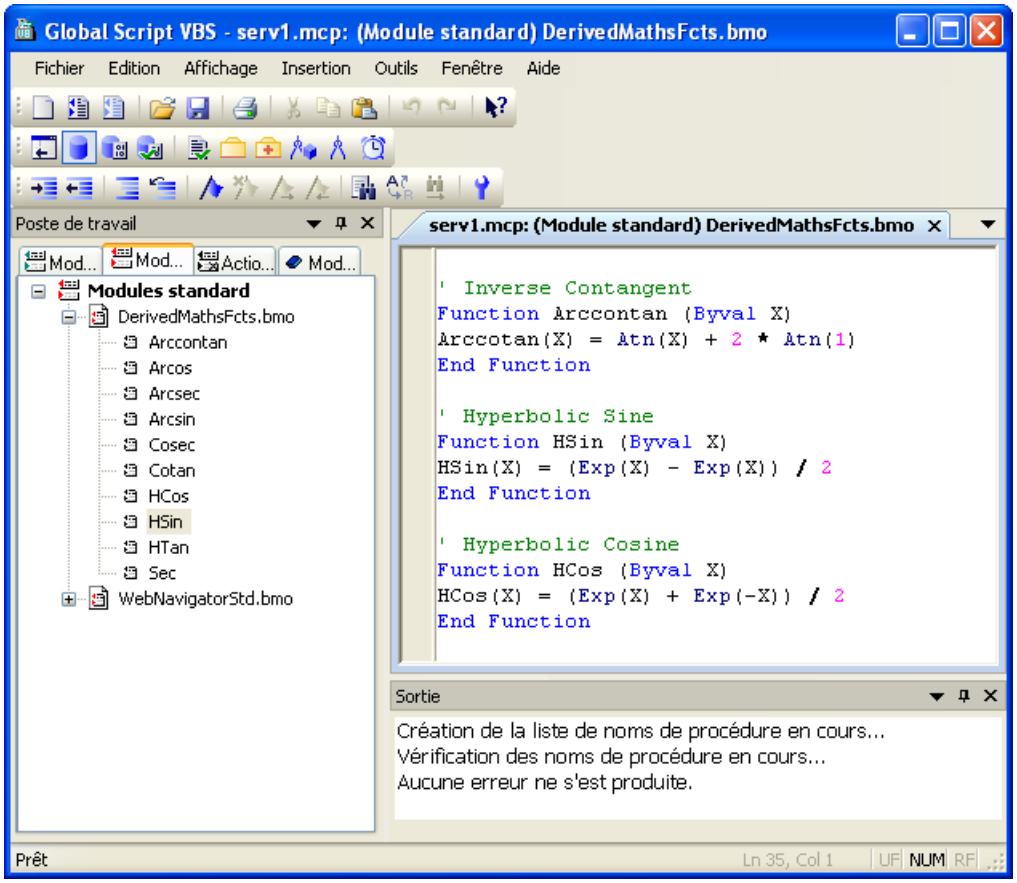
Vous enregistrez toujours dans des modules les procédures allant ensemble. Si vous appelez dans Runtime une procédure déterminée par le biais d'une action, le module chargé sera toujours celui dans lequel se trouve la procédure. Lors de la structuration de vos modules et procédures, il est donc nécessaire de prendre en considération les points suivants :

- Plus il y a de modules à charger lors de l'appel d'une vue, moins bonnes seront les performances en Runtime.
- Plus un module est grand, et donc plus il contient de procédures, plus le temps de chargement est long pour le module.

C'est pourquoi vous devez répartir vos modules de façon judicieuse, par exemple un module contenant les procédures pour une partie de l'installation ou une vue déterminée.

Une autre façon de structurer les procédures dans les modules est la structuration fonctionnelle, par exemple un module avec des fonctions mathématiques. Utilisez cette structuration par exemple pour les modules que vous voulez utiliser entre projets. L'exemple

suivant présente un module dans lequel sont inscrites des fonctions mathématiques dérivées des fonctions standard :



Caractéristiques des procédures

Les procédures sous WinCC ont les propriétés suivantes :

- Il vous revient de les créer et de les modifier.
- Vous pouvez les protéger contre toute consultation et modification grâce à un mot de passe.
- Elles n'ont pas de Trigger.
- Elles sont enregistrées dans un module.

WinCC ne met à disposition aucune procédure prédéfinie, mais vous propose par exemple des modèles de code et Intellisense pour simplifier la programmation. Selon le module auquel elles appartiennent, les procédures se classent en :

- procédures standard qui sont reconnues dans tous les projets sur l'ordinateur sur lequel elles ont été créés.
- Les procédures de projet ne peuvent être utilisées qu'au sein du projet dans lequel elles ont été créées.

Caractéristiques des modules

Un module est un fichier dans lequel une ou plusieurs procédures sont enregistrées. Les procédures sous WinCC ont les propriétés suivantes :

- Vous pouvez les protéger contre toute consultation et modification grâce à un mot de passe.
- le nom de fichier se termine par *.bmo.

Selon la validité de leurs procédures, les modules forment deux catégories :

- Les modules standard : ils contiennent des procédures disponibles dans tous les projets. Les modules standard sont inscrits dans le système de fichiers WinCC sous : <Répertoire d'installation WinCC>\ApLib\ScriptLibStd\<Nom du module>.bmo
- Les modules de projet : ils contiennent des procédures spécifiques au projet. Les modules de projet sont inscrits dans le système de fichiers WinCC sous : <Répertoire de projet> \ScriptLib\<Nom du module>.bmo. Etant donné que les modules de projet sont enregistrés dans le répertoire de projet, ils seront aussi copiés lors de la copie d'un projet WinCC.

Remarque

Si vous réinstallez WinCC et que vous voulez à nouveau utiliser les modules et procédures standard, sauvegardez les fichiers de module avant la réinstallation dans un autre répertoire et recopiez-les après l'installation dans le répertoire WinCC correspondant. Faute de quoi les modules standard se trouvant dans le répertoire d'installation WinCC seront effacés lors de l'installation.

Utilisation de procédures et de modules

Des procédures sont utilisées dans :

- les actions (dans Graphics Designer et Global Script)
- d'autres procédures (dans Global Script)
- Création de menus et de barres d'outils personnalisés

Vous structurez les procédures en modules.

Voir aussi

Création et édition de procédures (Page 40)

Les éditeurs VBScript (Page 28)

Principes de base de VBScript (Page 825)

Actions (Page 19)

Utilisation de Visual Basic Script dans WinCC (Page 12)

1.4 Actions

Introduction

Une action est toujours déclenchée par un Trigger. Dans le mode Runtime, une action est p. ex. exécutée lorsqu'un objet est manipulé par un clic sur la souris, lorsqu'une date/heure déterminée est arrivée ou qu'une valeur de variable a changé.

Caractéristique des actions

Les actions dans Global Script sont définies une seule fois pour ensuite être disponibles indépendamment des vues. Les actions Global Script ne sont valables que dans le projet dans lequel elles ont été définies. Les actions qui sont liées à des objets graphiques ne sont valables que dans la vue dans laquelle elles ont été définies.

Remarque

Dans VBS, il n'est actuellement pas possible de créer des actions spécifiques aux ordinateurs.

Valable pour les clients fonctionnant dans un système multiposte : Toutes les actions globales configurées sur un serveur sont également exécutées sur un client lors de l'ouverture d'un projet.
Valable pour les clients fonctionnant dans un système réparti : Si vous voulez utiliser des actions sur un ordinateur client, copiez sur le client les fichiers des actions dans le répertoire de projet correspondant.

Les actions ont les propriétés suivantes :

- Il vous revient de créer et de modifier les actions.
- Les actions dans Global Script peuvent être protégées par un mot de passe contre toute modification ou consultation.
- Les actions possèdent au moins un Trigger.
- Les actions dans Global Script ont un nom de fichier se terminant par *.bac.
- Les actions de Global Script sont inscrites dans le système de fichiers WinCC sous : <Répertoire de projet>\ScriptAct\Nom de l'action.bac

Trigger pour actions

Les Triggers sont nécessaires pour exécuter des actions en runtime. On lie un Trigger à une action pour créer l'événement déclencheur de l'appel de l'action. Les actions sans déclenchement ne sont pas exécutées.

Dans WinCC, les types de Trigger suivants sont disponibles :

- Temporisateur : Trigger acyclique ou cyclique, p. ex. lors de l'affichage d'une vue ou toutes les heures.
- Variables : Modification de la valeur
- Evénements : Modification des propriétés de l'objet (changement de couleur p. ex.) ou événements pour un objet (clic sur la souris p. ex.).

Traitement d'actions en Runtime

Dans le Graphics Designer

En Runtime, il n'est pas possible d'exécuter simultanément deux actions de même type. Pour que les actions cycliques ne soient par exemple pas entravées par une action exécutée par un clic sur la souris, les actions déclenchées par un événement et les actions cycliques/déclenchées par des variables sont exécutées indépendamment les unes des autres dans Graphics Designer.

Remarque

Notez que la synchronisation entre les deux types d'action dans WinCC ne peut se faire que par l'objet DataSet ou des variables internes WinCC. Le traitement s'effectuant séparément, il n'existe plus de zone de données commune entre les actions déclenchées par des événements et les actions cycliques/déclenchées par des variables.

Lorsque le traitement des actions cyclique dans des vues est entravé, par exemple par une surcharge du système ou d'autres actions, l'action sera à nouveau exécutée une fois dès que possible. Les cycles non réalisés ne sont pas placés dans une boucle d'attente, mais rejetés.

Après un changement d'image, les scripts en cours se terminent automatiquement au bout d'une minute.

Les scripts qui fonctionnent encore lorsque vous quittez Runtime se terminent au bout de cinq minutes.

Dans Global Script

Les actions de Global Script ne dépendant pas des vues sont exécutées les unes à la suite des autres dans Runtime lors du déclenchement. Lorsqu'une action est déclenchée, alors qu'une autre est justement en cours, la deuxième action est placée en boucle d'attente jusqu'à ce que son exécution soit possible.

Remarque

Notez qu'une synchronisation entre des actions dans Global Script et dans Graphics Designer ne peut avoir lieu que par l'objet DataSet ou des variables internes WinCC. Il n'existe aucune zone de données commune entre les actions dans Graphics Designer celles dans Global Script.

Utilisation des actions

Les actions peuvent être utilisées comme suit :

- Dans Global Script Les actions globales définies ici se déroulent en Runtime indépendamment des vues.
- Dans Graphics Designer : Les actions définies ici se déroulent uniquement dans la vue configurée. Il vous faut toujours configurer une action dans Graphics Designer pour les propriétés d'un objet ou un événement pour un objet graphique.

Voir aussi

Création et édition d'actions (Page 57)

Principes de base de VBScript (Page 825)

Modules et procédures (Page 16)

Utilisation de Visual Basic Script dans WinCC (Page 12)

1.5 Procédez comme suit pour utiliser plusieurs fois des procédures et des actions

Introduction

Une action configurée avec VBS dans WinCC est toujours valable pour le projet dans lequel elle a été définie.

Les domaines de validité des procédures sont les suivants:

- procédures standard qui sont reconnues dans tous les projets sur l'ordinateur sur lequel elles ont été créées
- procédures de projet qui ne peuvent être utilisées qu'au sein du projet dans lequel elles ont été créées. Lorsque vous copiez un projet, les procédures (modules) de projet sont copiées en même temps.

Utiliser plusieurs fois des procédures et des actions

Lorsque vous voulez utiliser des actions ou des procédures/modules dans d'autres projets ou sur d'autres ordinateurs, vous pouvez:

- soit utiliser la fonction "Enregistrer sous", pour enregistrer le module dans un autre répertoire de projet ou sur une disquette par exemple
- soit copier le fichier de l'action ou du module dans Windows-Explorer pour le copier sur l'ordinateur cible, dans le répertoire projet ou standard correspondant.

Les Trigger et propriétés configurés gardent leur validité lors du copiage. Les modules copiés sont directement disponibles en Runtime. Les actions copiées sont exécutées en Runtime après avoir été ouvertes et enregistrées.

Remarque

Les variables que vous utilisez dans une action ou une procédure doivent également exister sur l'ordinateur cible. L'action ou la procédure ne sont pas exécutées si la variable n'existe pas.

Les procédures que vous démarrez dans une action doivent exister sur l'ordinateur cible. Il y aura une erreur d'exécution dans Runtime si la procédure n'existe pas.

Inscription des procédures

Si vous voulez copier des procédures dans d'autres répertoires de projet pour les utiliser dans d'autres projets ou sur d'autres ordinateurs, prenez en considération le chemin des procédures au sein du système de fichiers WinCC:

- Procédures standard: <Répertoire d'installation WinCC>\ApLib\ScriptLibStd\Nom du module.bmo
- Procédures de projet: <Répertoire de projet>\ScriptLib\Nom du module.bmo

Remarque

Etant donné que les procédures sont toujours mémorisées dans des modules, copiez toujours le module (*.bmo) dans lequel se trouve la procédure.

Les procédures/modules copiés sont visibles après actualisation de la fenêtre de navigation de Global Script (commande "Mise à jour" dans menu contextuel) ou après redémarrage de l'éditeur.

Inscription d'actions

Si vous voulez copier des actions dans d'autres répertoires de projet pour les utiliser dans d'autres projets ou sur d'autres ordinateurs, prenez en considération le chemin des actions au sein du système de fichiers WinCC:

<Répertoire de projet>\ScriptAct\Nom de l'action.bac

Chaque action est enregistrée dans son propre fichier. Lors du copiage des actions, tous les Trigger en faisant partie sont également copiés.

Remarque

Seules les actions crées dans Global Script sont inscrites dans le système de fichiers WinCC. Les actions que vous voulez programmer dans Graphics Designer sont toujours enregistrées avec la vue actuelle et ne peuvent être transférées individuellement. Lorsque vous copiez une vue de Graphics Designer dans un autre répertoire de projet, les actions enregistrées avec la vue sont cependant également transférées.

Les actions copiées sont visibles après actualisation de la fenêtre de navigation de Global Script (commande "Mise à jour" dans menu contextuel) ou après redémarrage de l'éditeur.

Voir aussi

Modules et procédures (Page 16)

Pour renommer une procédure ou un module (Page 55)

Pour enregistrer une procédure (Page 53)

Pour protéger un module par un mot de passe (Page 51)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour utiliser des procédures de projet et standard (Page 48)

Pour écrire le code d'une procédure (Page 45)

Pour créer une nouvelle procédure (Page 44)

Création et édition de procédures (Page 40)

1.6 Interactions avec CrossReference

CrossReference et les Trigger de variable

CrossReference de WinCC vous permet de retrouver rapidement tous les occurrences de variables, même dans des actions VBS. Vous pouvez réaffecter avec CrossReference les Trigger de variable provenant des actions dans Graphics Designer, c'est-à dire les remplacer par d'autres variables à tous les emplacements ou seulement sur des emplacements choisis.

Remarque

Vous pouvez également réaffecter des variables directement dans Graphics Designer en sélectionnant l'objet graphique, puis la commande "Réaffecter..." dans le menu contextuel.

Vous trouverez de plus amples renseignements dans la documentation WinCC concernant CrossReference.

Actions et CrossReference

Vous pouvez afficher toutes les actions utilisées dans une vue via les propriétés de la vue. Sélectionnez à cet effet la vue dans l'explorateur WinCC, puis la commande "Propriétés" dans le menu contextuel. Vous obtiendrez des informations plus détaillées concernant le type de dynamisation en cliquant deux fois sur une inscription.

Vous pouvez en outre afficher toutes les variables et vues utilisées dans les actions via WinCC CrossReference. CrossReference vous permet également de réaffecter aisément des liaisons de variable des actions de Graphics Designer.

Variables et CrossReference

Toutes les variables qui sont adressées avec la formulation standard suivante sont saisies automatiquement par la CrossReference de WinCC et mentionnées dans les attributs des vues.

```
' VBS1
HMIRuntime.Tags("Tagname")
```

Si vous appelez des variables avec d'autres formules dans votre code, vous pouvez les communiquer à la CrossReference avec la section suivante :

```
' ' WINCC:TAGNAME_SECTION_START
Const TagNameInAction = "TagName"
' WINCC:TAGNAME_SECTION_END
```

Vous pouvez insérer cette section dans des actions VBS autant de fois que vous le voulez.

Remarque

Il n'y a aucune garantie que CrossReference saisisse des noms de variable composés.

Vues et CrossReference

Toutes les vues qui sont adressées avec la formulation standard suivante sont saisies automatiquement par la CrossReference de WinCC et mentionnées dans les attributs des vues.

```
'VBS2
HMIRuntime.BaseScreenName = "Screenname"
```

Si vous appelez des vues avec d'autres formules dans votre code, vous pouvez les communiquer à la CrossReference avec la section suivante :

```
' ' WINCC:SCREENNAME_SECTION_START
Const ScreenNameInAction = "ScreenName"
' WINCC:SCREENNAME_SECTION_END
```

Vous pouvez insérer cette section dans des actions VBS autant de fois que vous le voulez.

Remarque

Pour des raisons de compatibilité avec les versions ultérieures, il est toujours préférable d'écrire les noms de vue sans l'extension de fichier ".PDL" .

Voir aussi

Référence VBS (Page 110)

Les éditeurs VBScript (Page 28)

Principes de base de VBScript (Page 825)

Actions (Page 19)

Modules et procédures (Page 16)

Utilisation de Visual Basic Script dans WinCC (Page 12)

1.7 Utilisation de variables globales dans VBS

Introduction

Vous pouvez définir dans Global Script Editor les variables globales qui peuvent alors être utilisées dans toutes les actions et procédures.

Utilisation de variables globales dans Graphics Designer et Global Script

Lors de l'utilisation des variables dans Graphics Designer et Global Script, prenez en considération les conditions suivantes :

- Pour utiliser une variable globale dans une action dans Graphics Designer, appelez la procédure dans laquelle est définie la variable pour que le module correspondant soit chargé dans Runtime.
- Pour utiliser une variable globale dans une action dans Global Script, il vous faut appeler au moins une procédure dans une action globale au moins, et ce depuis le module dans lequel est définie la variable pour que ce dernier soit chargé dans Global Script Runtime. Il ne doit pas forcément s'agir de la procédure dans laquelle a été définie la variable.

Cette procédure est nécessaire étant donné que les actions de Global Script et de Graphics Designer en Runtime sont traitées indépendamment les unes des autres. Il n'existe pas de zone de données commune entre les deux systèmes Runtime.

Si vous devez synchroniser les actions de Global Script et de Graphics Designer, utilisez l'objet DataSet ou des variables internes WinCC.

Utilisation des variables globales au sein du Graphics Designer

Si vous utilisez des variables globales dans Graphics Designer, prenez en considération les conditions suivantes :

- Pour utiliser une variable globale dans une action cyclique ou déclenchée par une variable dans Graphics Designer, appelez la procédure dans laquelle est définie la variable. Cela est également valable lorsque la variable a déjà été appelée dans une action déclenchée par un événement.
- Pour utiliser une variable globale dans une action déclenchée par un événement dans Graphics Designer, appelez la procédure dans laquelle est définie la variable. Cela est également valable lorsque la variable a déjà été appelée dans une action cyclique ou déclenchée par une variable.

Cette procédure est nécessaire étant donné que les actions cycliques/déclenchées par une variable ainsi que les actions déclenchées par un événement dans Graphics Designer en Runtime sont traitées indépendamment les unes des autres. Il n'existe pas de zone de données commune entre les deux types d'action.

Utilisez l'objet DataSet ou des variables internes WinCC si vous devez synchroniser des actions cycliques/déclenchées par une variable et des actions déclenchées par un événement.

Vous avez en outre la possibilité de définir des variables globales dans une zone de déclaration séparée se trouvant dans Graphics Designer. Etant donné qu'en Runtime les actions déclenchées par un événement et les actions cycliques/déclenchées par une variable sont

traitées séparément, les variables globales ne sont ici accessibles en commun qu'au sein des actions déclenchées par un événement ou des actions cycliques/déclenchées par une variable.

Voir aussi

Principes de base de VBScript (Page 825)

Structure des fichiers VBScript (Page 97)

Création et édition d'actions (Page 57)

Création et édition de procédures (Page 40)

Les éditeurs VBScript (Page 28)

Interactions avec CrossReference (Page 24)

Actions (Page 19)

Modules et procédures (Page 16)

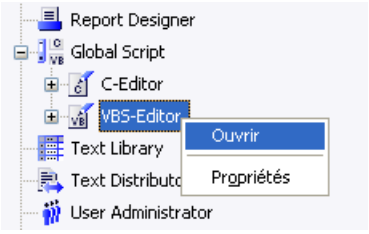
1.8 Les éditeurs VBScript

1.8.1 Les éditeurs VBScript

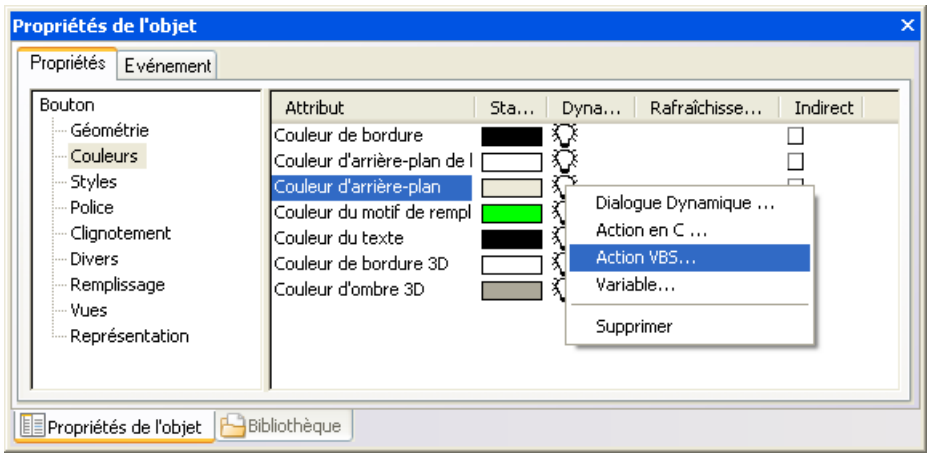
Introduction

Dans WinCC, il vous est possible de programmer des scripts VB en deux emplacements:

- Dans Global Script Global Script est l'éditeur central pour la programmation VBS. Vous pouvez l'appeler via l'explorateur WinCC.



- Dans Graphics Designer : Dans Graphics Designer, vous programmez des actions pour les propriétés de l'objet ou bien des événements pour des objets graphiques. Appelez l'éditeur d'actions dans Graphics Designer par le biais du menu contextuel dans le dialogue des propriétés d'un objet graphique.



Restrictions pour Global Script - Graphics Designer

Dans Graphics Designer, vous pouvez programmer des actions et des procédures spécifiques à la vue mais pas de procédures globales valables dans l'ensemble du projet. Vous pouvez cependant appeler des procédures globales qui ont été programmées dans Global Script.

Remarque

On décrira essentiellement Global Script dans cette documentation et les divergences par rapport aux fonctions de Graphics Designer seront au besoin mentionnées. Vous trouverez sous la rubrique d'aide de WinCC "Dynamisation" une description détaillée de l'éditeur d'actions dans Graphics Designer.

Informations complémentaires

Vous trouverez de plus amples renseignements dans la documentation "Dynamisation" de WinCC.

Voir aussi

L'éditeur Global Script (Page 29)

1.8.2 L'éditeur Global Script

Introduction

Vous pouvez créer et éditer des procédures globales et des actions dans Global Script Editor.

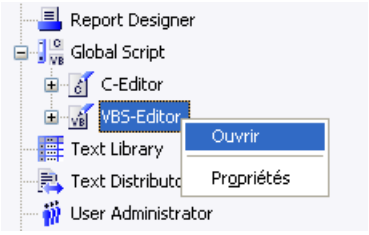
L'étendue des fonctions mises à votre disposition par Global Script est similaire à celle de C Script Editor dans WinCC.

Remarque

Vous trouverez dans la rubrique d'aide de WinCC "Dynamisation" une description détaillée de l'éditeur d'actions pour créer des procédures et des actions en fonction des vues dans Graphics Designer.

Appeler Global Script

Démarrez Global Script via le menu contextuel "Ouvrir" dans la fenêtre de projet de l'explorateur WinCC.

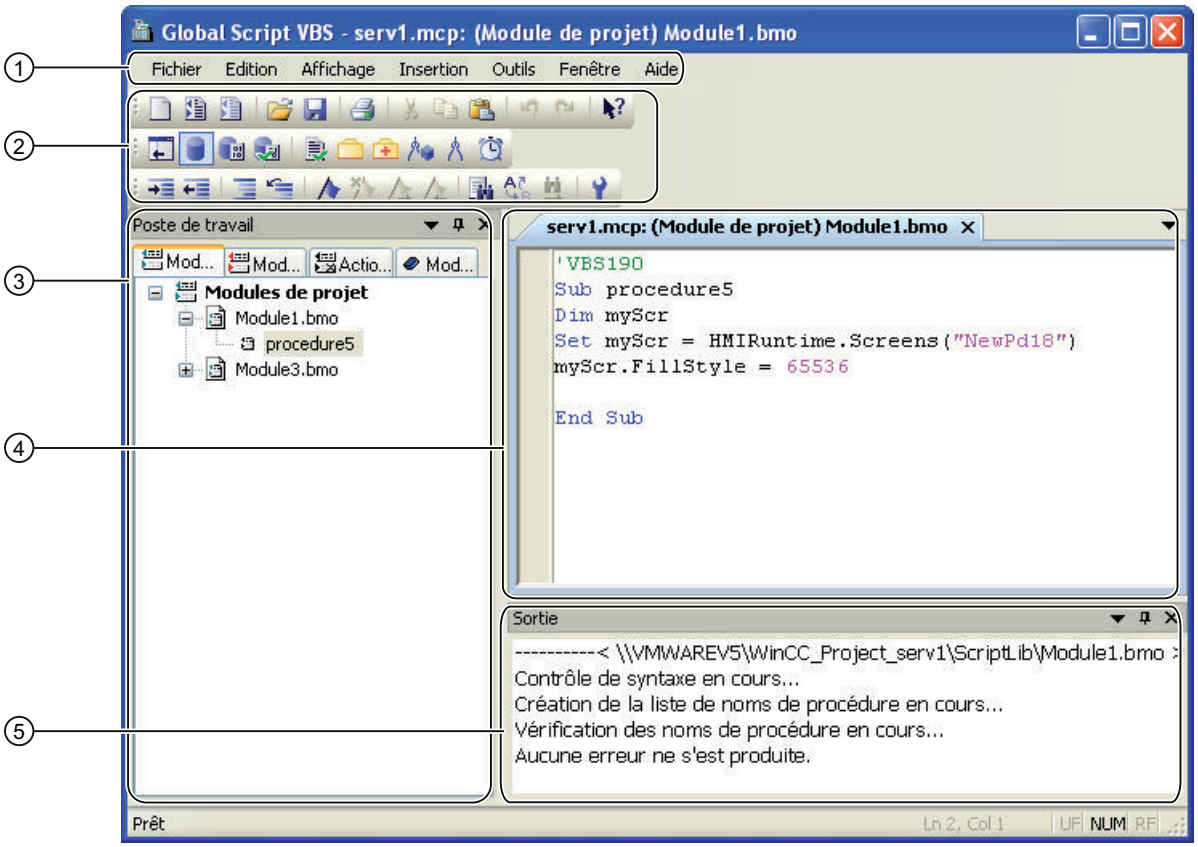


Global Script est également automatiquement appelé lorsque vous ouvrez par un double clic un module ou une action dans l'explorateur WinCC.

Structure de Global Script

Global Script est conçu selon les standards Windows.

L'éditeur d'actions dans Graphics Designer met à votre dispositions des fonctions similaires à celles de Global Script. Vous trouverez sous la rubrique d'aide de WinCC "Dynamisation" une description de l'éditeur d'actions dans le Graphics Designer.



Barre de menus (1) et barres d'outils (2)

Vous trouverez dans la barre de menus et dans les barres d'outils toutes les commandes dont vous avez besoin pour créer des procédures et des actions.

Les barres d'outils peuvent être activées et désactivées via la commande de menu "Affichage" > "Barres d'outils" et être déplacées dans l'éditeur à l'endroit voulu.

Fenêtre de navigation (3)

La fenêtre de navigation vous permet de gérer vos procédures, modules et actions. En outre, elle contient des modèles de code que vous pouvez ajouter par glisser&déplacer dans votre action ou procédure.

Vous pouvez appeler une procédure soit dans une autre procédure, soit dans une autre action, en la tirant par drag&drop de la fenêtre de navigation pour la placer à l'endroit respectif dans le code.

L'affichage de la fenêtre de navigation est mis à jour uniquement lors de l'enregistrement du document traité. Si vous modifiez un fichier, cela est indiqué par un * après le nom du fichier.

Dans la fenêtre de navigation, les procédures contenues dans un module s'affichent sous les fichiers. Les Triggers configurés pour les actions et les procédures au besoin directement définies dans un module d'action s'affichent en outre sur l'onglet "Actions".

La fenêtre de navigation vous permet en outre de :

- créer des sous-répertoires pour la structuration de vos scripts
- copier, coller, effacer, renommer directement des modules et des répertoires.

Vous pouvez configurer individuellement les affichages dans la fenêtre de navigation avec la commande de menu "Affichage" > "Poste de travail". Vous pouvez choisir ici si tous les types de fichiers, seulement les fichiers Script ou seulement les fichiers Script à syntaxe correcte doivent être affichés. La commande de menu "Affichage" > "Poste de travail" > "Afficher" vous permet d'afficher ou de masquer la fenêtre de navigation.

Fenêtre d'édition (4)

Dans la fenêtre d'édition, vous pouvez écrire et éditer vos procédures et vos actions. Chaque fonction ou action est ouverte dans une fenêtre d'édition qui lui est propre. Il est possible d'ouvrir simultanément plusieurs fenêtres d'édition.

Dans la fenêtre d'édition, l'utilisateur est assisté par la mise en relief de la syntaxe et Intellisense. Toutes les fonctions d'édition courantes (p. ex. Undo/Redo, Rechercher/Remplacer, Copier, Coller, Couper, Réglage de la police de caractères, Assistance imprimante) restent disponibles.

Fenêtre de sortie (5)

Les messages d'erreur s'affichent dans la fenêtre de sortie après un contrôle de syntaxe. Cliquez deux fois sur la ligne d'erreur correspondante pour parvenir à l'endroit respectif dans le code.

Ligne d'état (6)

La ligne d'état contient des informations concernant les fonctions actuellement choisies ou des conseils concernant la programmation.

Remarque

Si vous avez besoin d'informations concernant une commande ou un icône de l'éditeur, sélectionnez la commande de menu "?" > "Aide directe". Cliquez avec la souris sur l'icône/la commande correspondant(e). Vous disposez ainsi d'une aide directe rapide pour tous les organes de commande de l'éditeur. Appuyez "ESC" lorsque vous voulez quitter le mode d'aide directe.

Ancrage des fenêtres

L'ancrage des fenêtres permet de disposer les différentes fenêtres de manière flexible. Vous avez la possibilité de déplacer les fenêtres, de les positionner librement en tant que fenêtres autonomes ou de les regrouper dans des groupes d'onglets. Vous pouvez, par exemple, positionner vos actions les unes à côté des autres, les unes sous les autres ou sous forme d'onglets. Vous pouvez masquer automatiquement des fenêtres et les réafficher en cas de besoin. Pour plus d'informations, reportez-vous au chapitre "Créer des vues de process".

Voir aussi

Procédez comme suit pour supprimer des actions ou des procédures (Page 38)

Utiliser les barres d'outils (Page 35)

Travailler dans la fenêtre d'édition (Page 32)

1.8.3 Travailler dans la fenêtre d'édition

Introduction

La fenêtre d'édition vous permet d'éditer des procédures et des actions.

Zone de déclaration dans des actions (Graphics Designer uniquement)

Quand vous créez des actions dans le Graphics Designer, vous pouvez afficher dans la fenêtre d'édition la zone de déclaration de l'action avec le bouton .

La zone de déclaration vous permet en outre de choisir les définitions générales que vous ne voulez utiliser que pour la vue actuelle, par exemple :

- définitions de variables
- procédures que vous ne désirez utiliser que dans cette vue

Remarque

Dans la zone "Déclaration" vous ne devez pas créer de code directement exécutable!

Notez que lors de la création d'une variable, cette dernière ne contient aucune valeur (Value = VT_EMPTY). Initialisez les variables après la déclaration avec une valeur correspondante.

Lors des procédures de définition dans la zone "Déclaration", prenez en considération la structure des fichiers Script comme décrit sous "Structure des fichiers VBScript".

"Option explicit" dans des actions

Lors de la création d'une nouvelle action, la commande "Option explicit" non supprimable s'inscrit automatiquement dans la zone de déclaration (Graphics Designer) ou sur la première ligne d'une action (Global Script). La commande est nécessaire pour éviter des erreurs par faute d'orthographe dans les variables sans déclaration. L'instruction vous invite à toujours définir les variables avec l'instruction "Dim" dans votre code.

Remarque

N'utilisez pas la commande "Option explicit" dans votre code, sinon il pourrait y avoir des erreurs d'exécution.

Assistance de l'utilisateur dans la fenêtre d'édition

Lorsque vous travaillez dans la fenêtre d'édition, vous êtes assisté par les fonctions suivantes.

Codage des couleurs et enfilades dans la fenêtre d'édition

Certaines parties du code sont en standard identifiées par des couleurs :

Couleur	Signification	Exemple
bleu	Mots-clés Fonctions	Sub, End Sub, Next
vert	Commentaires	' c'est un commentaire
rouge	Chaînes de caractères (chaîne de caractères et nombres)	"Object1"
bleu foncé	Instructions préprocesseur	--
gras, noir	Constantes	**vbTrue, vbFalse**
noir	Autre élément de code	--

Vous pouvez déterminer individuellement le codage des couleurs dans la fenêtre d'édition par le biais des réglages de l'éditeur. Appelez à cet effet via la commande de menu "Outils" > "Options" le dialogue "Script Editor Options" et déterminez y les paramètres.

Pour plus de clarté, vous pouvez structurer votre code par des retraits. Dans le dialogue "Script Editor Options", il vous est également possible de régler à cet effet la distance entre tabulateurs et l'inscription automatique en retrait.

Intellisense et mise en relief de la syntaxe

Pendant l'entrée apparaissent des listes contextuelles, dans lesquelles vous sont proposés les propriétés, les méthodes et les objets possibles à l'emplacements de code actuel. Quand vous insérez un élément de la liste, la syntaxe nécessaire vous est spécifiée automatiquement.

Remarque

L'Intellisense complète pour tous les objets n'est utilisable dans le Graphics Designer que quand l'accès est effectué par le nom d'objet sur la liste et que le résultat est affecté à une variable. Sinon, le système n'affiche que les propriétés standard.

Exemple d'Intellisense complète :

```
Variable Dim
Positionner Variable = ScreenItems ("Cercle1")
Variable.
```

Quand, lors de l'adressage, des limites de la fenêtre de vue sont dépassées, seules les propriétés standard sont également proposées, étant donné que la vue de la fenêtre de vue n'est pas chargée.

Vous pouvez activer et désactiver la mise en relief de la syntaxe dans le dialogue "Script Editor Options". Appelez le dialogue par la commande de menu "Outils" > "Options".

Fonctions générales VBS

Vous pouvez afficher une liste de fonctions générales VBS via la commande contextuelle "Liste de fonctions" dans la fenêtre d'édition.

Listes des objets, des propriétés et des méthodes

Par le menu contextuel dans la fenêtre d'édition, vous pouvez faire afficher une liste des objets possibles avec la commande "Liste d'objets" dans le Graphics Designer. Dans Global Script, vous n'obtenez dans cette liste que l'objet "HMIRuntime", étant donné que vous ne pouvez pas directement accéder aux objets de Graphics Designer.

A l'aide de la commande contextuelle "Propriétés/Méthodes", vous obtenez une liste des propriétés et méthodes possibles.

Vous obtenez les mêmes listes, selon le contexte, dans le Script avec la combinaison de touches <Ctrl+Espace>.

Modèles de code

Dans la fenêtre de navigation de l'éditeur sur l'onglet Modèles de code, vous trouverez une sélection d'instructions souvent utilisées, par ex. pour des boucles et des instructions conditionnées. Vous pouvez insérer ces modèles par "drag&drop" dans votre code de procédure.

Lorsque vous insérez un modèle de code dans votre code, vous devez remplacer les caractères génériques "_XYZ_" des modèles par les données correspondantes.

Dialogues de sélection

Si vous utilisez des variables ou des objets WinCC dans le code, les dialogues de sélection suivants sont à votre disposition :

- Ouvre un dialogue de sélection pour variables et donne le nom de variable sélectionné comme valeur de retour.
- Ouvre un dialogue de sélection pour variables et donne en retour le nom de variable avec la référence correspondante.
- Ouvre un explorateur de vues/d'objets où il est possible de sélectionner une vue/un objet dont le nom est restitué comme valeur de retour.
- Ouvre un dialogue de sélection pour les vues et restitue le nom de vue, au besoin accompagné d'un préfixe Serveur.

Signet

Vous pouvez placer des signets dans votre code pour retrouver plus facilement telle ou telle position dans le code.

- Positionne un signet sur la ligne où se trouve actuellement le curseur.
- Supprime tous les signets dans la fenêtre d'édition active.
- Saute dans le code d'un signet vers l'avant.
- Saute dans le code d'un signet vers l'arrière.

Voir aussi

Structure des fichiers VBScript (Page 97)

L'éditeur Global Script (Page 29)

1.8.4 Utiliser les barres d'outils

Objectif

La position standard des barres d'outils est sous la barre de menu, en haut de la fenêtre de l'éditeur VBS. Les touches figurant sur les barres d'outils permettent d'accéder rapidement et aisément aux fonctions proposées par Global Script ou l'éditeur d'actions dans Graphics Designer.

Dans Global Script/Graphics, vous disposez des barres d'outils suivantes :

Barre d'outils "Standard"

Bouton	Fonction	Combinaison de touches
	Crée un nouveau module de projet (Global Script uniquement)	<Alt+F1>
	Crée un nouveau module standard (Global Script uniquement)	<Alt+F2>
	Crée une nouvelle action globale (Global Script uniquement)	<Alt+F3>
	Ouvre une action ou un module existant déjà (Global Script uniquement)	<CTRL+O>
	Enregistre le contenu de la fenêtre d'édition active. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte. L'affichage est mis à jour dans la fenêtre de navigation après l'enregistrement. (Global Script uniquement)	<CTRL+S>
	Coupe le texte marqué et le copie dans le presse-papiers. La fonction n'est disponible que lorsqu'un texte est marqué.	<CTRL+X>
	Copie dans le presse-papiers le texte marqué. La fonction n'est disponible que lorsqu'un texte est marqué.	<CTRL+C>
	Insère le contenu du presse-papiers à la position du curseur. Cette fonction n'est disponible que si le presse-papiers n'est pas vide.	<CTRL+V>
	Imprimer le contenu de la fenêtre d'édition active comme documentation de projet. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte.	<CTRL+P>

Contenu de la barre d'outils "Editeur"

Bouton	Fonction	Combinaison de touches
	Déplace d'une position vers la droite la ligne sur laquelle se trouve le curseur.	--
	Déplace d'une position vers la gauche la ligne sur laquelle se trouve le curseur.	--
	Identifie en tant que commentaire les lignes sélectionnées avec la souris. Si aucune ligne n'est sélectionnée avec la souris, c'est la ligne sur laquelle se trouve le curseur qui est identifiée en tant que commentaire.	--
	Supprime l'identification en tant que commentaire sur la ligne sélectionnée avec la souris. Si aucune ligne n'est sélectionnée avec la souris, c'est l'identification se trouvant sur la ligne où se trouve le curseur qui est supprimée.	--
	Place un signet sur la ligne actuelle. Le signet est effacé de la ligne actuelle si l'on appuie une seconde fois.	<Ctrl+F9>
	Supprime tous les signets dans le code actuel dans la fenêtre d'édition.	<Ctrl+Maj+F9>
	Place le curseur au signet suivant.	<F9>

Bouton	Fonction	Combinaison de touches
	Place le curseur au signet précédent.	<Ctrl+F9>
	Ouvre le dialogue "Rechercher" pour chercher des textes dans le code.	<Ctrl+F>
	Ouvre le dialogue "Remplacer" pour rechercher et remplacer des textes dans le code.	<Ctrl+H>
	Recommence la procédure de recherche.	<F3>
	Ouvre le dialogue "Script Editor Options".	--
	Annuler la dernière de 30 actions d'édition (maxi.). La fonction n'est disponible que si une action d'édition a été exécutée.	<CTRL+Z>
	Restaurer la dernière action d'édition annulée. La fonction n'est disponible que lorsqu'une action d'édition a été annulée.	<Ctrl+Y>

Contenu de la barre d'outils "Editer"

Bouton	Fonction	Combinaison de touches
Poste de travail		
	Sélectionne le fichier dans la fenêtre de navigation auquel appartient la fenêtre d'édition actuelle (Global Script uniquement).	--
	Affiche tous les fichiers dans la fenêtre de navigation (Global Script uniquement).	--
	Affiche uniquement les fichiers Script dans la fenêtre de navigation (Global Script uniquement).	--
	Affiche uniquement les fichiers Script à syntaxe correcte dans la fenêtre de navigation (Global Script uniquement).	--
Script		
	Exécute le contrôle de syntaxe dans le code de la fenêtre d'édition actuelle.	<F7>
Objets WinCC		
	Ouvre le dialogue "Info/Trigger".	<Ctrl+T>
Déclenchement standar	Affiche le nom du déclenchement.	
	Ouvre un dialogue de sélection pour variables et donne le nom de variable sélectionné comme valeur de retour.	<Ctrl+U>
	Ouvre un dialogue de sélection pour variables et donne en retour le nom de variable avec la référence correspondante.	<Ctrl+W>

Bouton	Fonction	Combinaison de touches
	Ouvre une vue/Browser d'objet où il est possible de sélectionner une vue/un objet dont le nom est donné comme valeur de retour.	<Ctrl+Q>
	Ouvre un dialogue de sélection pour les vues et donne en retour le nom de vue, au besoin accompagné d'un préfixe Serveur.	<Ctrl+B>

Boutons supplémentaires dans Graphics Designer

Outre les boutons existant dans Global Script, l'éditeur d'actions dispose dans Graphics Designer des boutons suivants :

- Affichage de la zone de déclaration (<Ctrl+E>)
- Masquage de la zone de déclaration (<Ctrl+A>)

Voir aussi

L'éditeur Global Script (Page 29)

1.8.5 Procédez comme suit pour supprimer des actions ou des procédures

Introduction

Si vous supprimez une action, une procédure ou un module dans un éditeur de Script, le code et le fichier correspondant sont également effacés dans le répertoire de projet.

Faites attention de n'effacer que les procédures n'étant plus utilisées dans d'autres procédures ou d'autres actions. Si une action appelle une procédure n'existant plus, l'action est interrompue en Runtime à l'endroit défectueux. Une référence dans le code ne peut pas être reconnue par le contrôle de syntaxe si elle n'existe pas.

Remarque

Vous ne pouvez supprimer les procédures qu'au sein d'un module, en effaçant le code, et non dans la fenêtre de navigation de l'éditeur.

Marche à suivre

1. Ouvrez Global Script.
2. Sélectionnez dans la fenêtre de navigation l'action ou le module que vous voulez supprimer.
3. Choisissez la commande "Supprimer" dans le menu contextuel.
4. Si vous voulez supprimer une procédure: Ouvrez le module correspondant et effacez le code correspondant dans la fenêtre d'édition.

Voir aussi

Actions (Page 19)
Modules et procédures (Page 16)
L'éditeur Global Script (Page 29)

1.9 Création et édition de procédures

1.9.1 Création et édition de procédures

Introduction

Dans WinCC, vous pouvez programmer avec VBS des procédures de projet et standard :

- Les procédures de projet ne peuvent être appelées que dans le projet actuel. Etant donné que les procédures sont inscrites dans le répertoire de projet, elles seront automatiquement copiées lors de la procédure de copiage d'un projet.
- Les procédures standard peuvent être appelées dans tous les projets sur tout l'ordinateur. Si vous copiez un projet sur un autre ordinateur, il vous faut copier manuellement les procédures standard sur l'ordinateur cible, dans le répertoire correspondant.

Les procédures copiées sont directement disponibles en Runtime. Elle sont visibles dans l'éditeur une fois que l'affichage a été mis à jour.

Outre les procédures que vous avez programmées, il vous est possible d'utiliser des fonctions VBS générales (p. ex. Abs, Array,... Year). Vous pouvez appeler ces fonctions générales VBS dans votre code via la commande "Liste de fonctions" dans le menu contextuel.

WinCC met en outre à votre disposition les instructions les plus courantes en tant que modèles de code (p. ex. If...Then, When...While). Vous pouvez faire glisser par drag&drop de la fenêtre de navigation sur l'onglet "Modèles de code" les modèles de code pour les placer directement dans votre code.

Lorsque vous insérez un modèle de code dans votre code, notez que les conditions sont par exemple identifiées par "_XYZ_" dans les modèles. Vous devez compléter cet affichage par défaut par les données correspondantes.

Utilisation de procédures

Utilisez les procédures pour créer et gérer de façon centralisée des codes que vous voulez utiliser en plusieurs endroits de votre configuration. Vous écrivez et enregistrez le code dans une procédure, puis appelez cette procédure avec les paramètres actuels dans des actions ou d'autres procédures au lieu de taper à nouveau le même code.

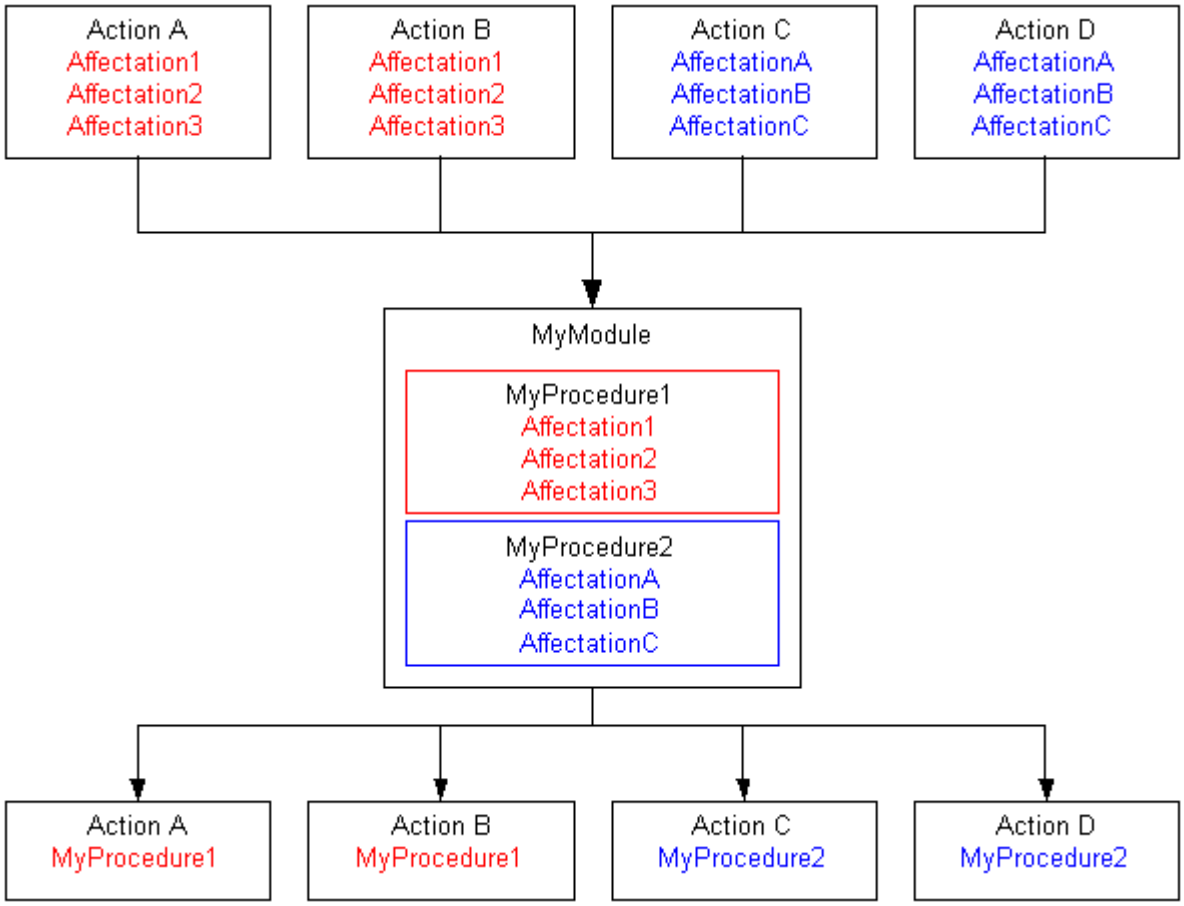
Utilisez les procédures pour des fonctions répétitives, par exemple

- Les calculs avec des valeurs initiales différentes (procédure avec valeur de retour)
- Contrôle des valeurs de variable (procédure avec valeur de retour)
- Opérations exécutant (procédure sans valeur de retour)

Ceci présente les avantages suivants:

- Le code n'est programmé qu'une seule fois.
- Les modifications sont faites en un seul endroit, dans la procédure et pas dans chaque action
- Le code des actions est plus court et plus clair.

Dans WinCC, vous enregistrez toujours dans des modules les procédures allant ensemble.



Les procédures sont chargées en Runtime lors de l'exécution de l'action appelant.

Si une procédure (module) utilisée dans une vue est modifiée, la modification est validée lorsque la vue est à nouveau chargée. Une vue actuellement représentée fonctionne donc avec la procédure modifiée seulement après que la vue ait été rechargée.

Après avoir modifié un module de projet vous devez, en plus de l'enregistrement dans l'éditeur VBS, ouvrir la vue de process correspondante dans Graphics Designer et l'enregistrer. C'est

alors seulement que la modification est appliquée dans Runtime. Une fois la vue enregistrée, les informations sur les modules de projet requis sont adoptées dans le fichier de vue.

Remarque

Les procédures peuvent être utilisées dans des actions dans Global Script et dans Graphics Designer.

Si vous voulez utiliser une variable globale définie dans Global Script au cours d'une action de Graphics Designer, prenez en considération le point suivant :

Pour que vous puissiez accéder à la variable, il vous faut toujours appeler la procédure dans laquelle est définie la variable.

Dans Global Script, si vous voulez utiliser une variable globale dans des actions ne dépendant pas de la vue, prenez en considération le point suivant :

Pour que vous puissiez accéder à la variable, il vous faut appeler dans une action globale au moins, au moins une procédure du module dans lequel a été définie la variable.

Restriction Procédure - Action

Vous ne pouvez créer que dans Global Script les procédures globales qui sont valables dans l'ensemble du projet. Dans Graphics Designer, il vous est seulement possible de créer des procédures spécifiques à la vue et d'appeler des procédures globales dans des actions. Définissez dans la zone de déclaration d'une action les procédures spécifiques à la vue dans Graphics Designer.

Une procédure n'est pas exécutée sans action.

Nom de fichier et nom de procédure

Spécifiez le nom de la procédure sur la première ligne du code de procédure. La procédure s'affiche sous ce nom dans la fenêtre de navigation et est appelée dans des actions. Les procédures n'ont pas leur propre nom de fichier, mais sont enregistrées dans un module.

Attribuez le nom de module dans la fenêtre de navigation de l'éditeur. La commande "Enregistrer sous" vous permet d'inscrire un module sous un autre nom dans le répertoire de projet.

Etant donné que les procédures dans Global Script sont valables à l'échelle du projet, il est impératif que les noms de procédure soient toujours univoques. Les noms de module peuvent être utilisés deux fois dans un même projet s'ils sont par exemple inscrits dans des sous-répertoires différents ou séparément dans le répertoire standard et dans le répertoire de projet.

Représentation des procédures et des modules

Si vous enregistrez un module contenant au moins une procédure dont la syntaxe est erronée, il sera affiché dans la fenêtre de navigation accompagné d'un symbole.

Remarque

Si un module contient une procédure à syntaxe incorrecte, il ne peut plus être chargé. Il n'est plus possible d'appeler une procédure depuis le module.

Si vous enregistrez un module ne contenant que des procédures à syntaxe correcte, il sera affiché dans la fenêtre de navigation accompagné d'un symbole.

Procédures et modules

Les procédures sont classées en fonction de leur affectation aux modules standard ou de projet, soit procédure standard, soit procédure de projet. Vous trouverez les modules standard et de projet dans la fenêtre de navigation de Global Script sur les onglets correspondants.

Utilisez les modules pour regrouper les procédures en groupes fonctionnel appropriés. Lors de la structuration de vos modules et procédures, prendre en considération les points suivants :

- Plus il y a de modules à charger lors de l'appel d'une vue, moins bonnes seront les performances en Runtime.
- Plus un module est grand, et donc plus il contient de procédures, plus le temps de chargement est long pour le module.

C'est pourquoi vous devez répartir vos modules de façon judicieuse, par exemple un module contenant les procédures pour une partie de l'installation ou une vue déterminée.

Voir aussi

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)

Pour renommer une procédure ou un module (Page 55)

Pour enregistrer une procédure (Page 53)

Pour protéger un module par un mot de passe (Page 51)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour utiliser des procédures de projet et standard (Page 48)

Pour écrire le code d'une procédure (Page 45)

Pour créer une nouvelle procédure (Page 44)

Exemples pour VBScript (Page 792)

Modules et procédures (Page 16)

1.9.2 Pour créer une nouvelle procédure

Introduction

Dans WinCC, vous pouvez programmer avec Global Script des procédures de projet et standard.

L'affectation soit à un module de projet, soit à un module standard vous permet de décider le type de procédure. La procédure pour créer des procédures de projet ou standard est identique.

Si vous créez une nouvelle procédure, WinCC lui attribue automatiquement un nom standard "procedure#" où # représente un numéro consécutif. Lorsque vous avez édité la procédure dans la fenêtre d'édition, donnez lui un nom adéquat via lequel vous l'appellerez ultérieurement dans des actions. Le nouveau nom est validé lors de l'enregistrement de la procédure dans la fenêtre de navigation.

Remarque

Le nom des procédures doit être univoque dans un projet. Si une procédure de même nom existe déjà, le module est identifié comme ayant une syntaxe incorrecte. Le nom des modules peuvent être utilisés deux fois si les modules sont inscrits dans des répertoires différents.

Vous ne pouvez programmer les procédures globales (valables à l'échelle du projet) que dans Global Script. Dans Graphics Designer, vous pouvez appeler des procédures via des actions et créer des procédures relatives aux vues dans le domaine de déclaration d'une action. Dans une action globale dans Global Script, il vous est également possible de créer directement dans le code des procédures qui ne sont alors valables que pour cette action.

Pour appeler des procédures, il faut toujours que vous programmiez une action.

Marche à suivre

La lobal Script:

1.

2. re de projet,
u Module de projet.

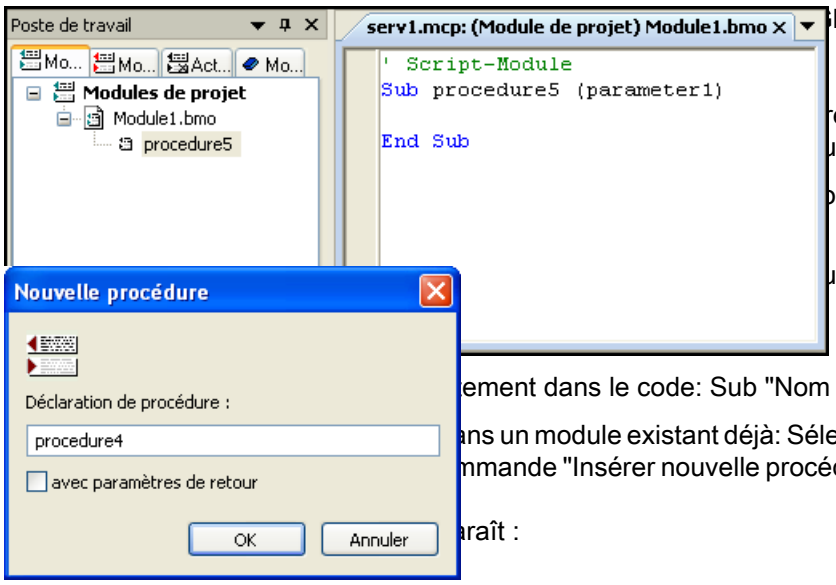
3. ommande de menu
"Module standard".

4. ure sans valeur de

5. ement dans le code: Sub "Nom de procédure".

6. ns un module existant déjà: Sélectionnez le module
mmande "Insérer nouvelle procédure" dans le menu

raît :



7. Spécifiez un nom de procédure et décidez si la procédure doit avoir un paramètre de retour. La définition d'une variable pour la valeur de retour est alors déjà inscrite dans le code (Dim RetVal).
8. Validez vos entrées avec OK.

Remarque

Vous pouvez évidemment également inscrire directement une nouvelle procédure dans le module. Une procédure sans valeur de retour doit toujours commencer par l'instruction "Sub <Nom de la procédure>" et se terminer par "End Sub". Une fonction avec valeur de retour doit toujours commencer par l'instruction "Fonction <Nom de la procédure>" et se terminer par "End Function". La nouvelle procédure s'affiche dans la fenêtre de navigation lors de l'enregistrement du module.

Voir aussi

Création et édition de procédures (Page 40)

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)

Pour renommer une procédure ou un module (Page 55)

Pour enregistrer une procédure (Page 53)

Pour protéger un module par un mot de passe (Page 51)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour utiliser des procédures de projet et standard (Page 48)

Pour écrire le code d'une procédure (Page 45)

Modules et procédures (Page 16)

1.9.3 Pour écrire le code d'une procédure

Introduction

Ecrivez le code de la procédure dans la fenêtre d'édition de Global Script. Toute procédure peut appeler dans son code d'autres procédures via son nom de procédure.

Vous pouvez créer des procédures avec ou sans valeur de retour. Faites usage de la valeur de retour pour par exemple savoir si la procédure a été correctement exécutée.

Si vous modifiez une procédure dans une vue, la modification n'est acceptée qu'au changement de la vue suivante.

Fonctions dans Global Script

Global Script vous assiste lors de la création des codes de procédure, avec les fonctions suivantes :

Intellisense et mise en relief de la syntaxe

Pendant la saisie, des listes contextuelles proposent les propriétés, les méthodes et les objets possibles à l'emplacements de code actuel. Quand vous insérez un élément de la liste, la syntaxe nécessaire est spécifiée automatiquement.

Remarque

L'Intellisense complète pour tous les objets n'est utilisable dans le Graphics Designer que quand l'accès est effectué par le nom d'objet sur la liste et que le résultat est affecté à une variable. Sinon, seules les propriétés standard vous sont proposées.

Exemple d'Intellisence complète :

```
Dim Variable
Set Variable = ScreenItems ("Cercle1")
```

```
Variable.<Sélection Intellisense>
```

Quand, lors de l'adressage, des limites de la fenêtre de vue sont dépassées, seules les propriétés standard sont également proposées, étant donné que la vue de la fenêtre de vue n'est pas chargée.

Fonctions générales VBS

Vous pouvez afficher une liste de fonctions générales VBS via la commande contextuelle "Liste de fonctions" dans la fenêtre d'édition.

Listes des objets, des propriétés et des méthodes

Par le menu contextuel dans la fenêtre d'édition, vous pouvez faire afficher une liste des objets possibles avec la commande "Liste d'objets" dans le Graphics Designer. Dans Global Script, vous n'obtenez dans cette liste que l'objet "HMIRuntime", étant donné que vous ne pouvez pas directement accéder aux objets de Graphics Designer.

A l'aide de la commande contextuelle "Propriétés/Méthodes", vous obtenez une liste des propriétés et méthodes possibles.

Vous obtenez les mêmes listes, selon le contexte, dans le Script avec la combinaison de touches <Ctrl+Espace>.

Modèles de code

Dans la fenêtre de navigation de l'éditeur sur l'onglet Modèles de code, vous trouverez une sélection d'instructions souvent utilisées, par ex. pour des boucles et des instructions conditionnées. Vous pouvez insérer ces modèles par "drag&drop" dans votre code de procédure.

Lorsque vous insérez un modèle de code dans votre code, vous devez remplacer les caractères génériques "_XYZ_" des modèles par les données correspondantes.

Dialogues de sélection

Si vous utilisez des variables ou des objets WinCC dans le code, les dialogues de sélection suivants sont à votre disposition :

- Ouvre un dialogue de sélection pour variables et donne le nom de variable sélectionné comme valeur de retour.
- Ouvre un dialogue de sélection pour variables et donne en retour le nom de variable avec la référence correspondante.
- Ouvre un explorateur de vues/d'objets où il est possible de sélectionner une vue/un objet dont le nom est restitué comme valeur de retour.
- Ouvre un dialogue de sélection pour les vues et restitue le nom de vue, au besoin accompagné d'un préfixe Serveur.

Contrôle de syntaxe

Global Script vous assiste par un contrôle de syntaxe que vous pouvez effectuer après avoir créé le code. Les erreurs syntaxiques dans le code sont affichées dans la fenêtre de sortie de l'éditeur. Vous accédez à l'emplacement de l'erreur dans le code par un double-clic sur l'erreur dans la fenêtre de sortie.

Remarque

Le contrôle de syntaxe ne peut détecter que les erreurs syntaxiques dans le code. Des erreurs de programmation, comme par ex. des références manquantes, ne sont reconnues qu'en Runtime. Pour cette raison, ne manquez pas de toujours contrôler vos scripts dans l'environnement Runtime.

Modification d'une procédure

Si vous modifiez une procédure en Runtime, la modification devient active aux instants suivants :

- les procédures qui sont appelées par des actions ou d'autres procédures dans des vues deviennent actives après un changement de vue
- les procédures dans Global Script sont actives directement, dès l'appel suivant

Marche à suivre

1. Ouvrez Global Script.
2. Ouvrez le module dans lequel vous voulez éditer une procédure.
3. En cliquant deux fois sur la procédure dans la fenêtre de navigation, le curseur retourne au début de la procédure voulue.
4. Editez la procédure. Si vous créez une procédure avec paramètres de renvoi, par ex. pour programmer des calculs ou des contrôles répétitifs, spécifiez en fin de procédure la valeur en retour avec "procedurename = RetVal" an.

Voir aussi

Pour écrire le code d'une procédure (Page 45)

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)

Pour renommer une procédure ou un module (Page 55)

Pour enregistrer une procédure (Page 53)

Pour protéger un module par un mot de passe (Page 51)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour créer une nouvelle procédure (Page 44)

Modules et procédures (Page 16)

Création et édition de procédures (Page 40)

1.9.4 Pour utiliser des procédures de projet et standard

Introduction

Insérez une procédure dans le code actuel tout simplement par drag&drop depuis la fenêtre de navigation ou via le menu contextuel.

Vous ne pouvez utiliser les procédures de projet qu'au sein du projet actuel, tandis que les procédures standard sont, elles, disponibles dans tous les projets se trouvant sur le PC.

Vous pouvez également utiliser dans d'autres projets ou sur d'autres ordinateurs des procédures que vous avez déjà créées une fois. Copiez à cet effet dans les répertoires de projet ou standard les modules qui contiennent les procédures.

Utilisation de procédures dans Graphics Designer et Global Script

Une fois que vous avez défini des procédures dans Global Script, vous pouvez les appeler dans des actions dans Global Script et Graphics Designer. Lorsque l'action s'effectue en Runtime, c'est toujours la totalité du module, dans lequel se trouve la procédure, qui est chargé.

Si vous voulez utiliser une variable globale qui est définie dans une procédure dans Global Script, prenez en considération les points suivants :

Dans Graphics Designer, il vous faut toujours appeler la procédure dans laquelle est définie la variable pour que vous puissiez utiliser cette dernière. Sans l'appel de procédure, le module correspondant n'est pas chargé et il est impossible d'accéder à la variable.

Dans les actions ne tenant pas compte de la vue dans Global Script, il vous faut toujours appeler dans une action globale au moins, une procédure au moins depuis le module dans lequel est définie la variable.

Remarque

En général, la partie déclaration des vues n'est pas vérifiée si un nom de procédure ou de fonction a déjà été attribué. Il est donc théoriquement possible qu'un nom soit utilisé plusieurs fois et que la fonction devant être exécutée ne soit pas définie. Ceci correspond au comportement standard de MS Scripting Engine.

Marche à suivre

1. Ouvrez la procédure ou l'action dans laquelle vous voulez insérer la procédure.
2. Dans la fenêtre de navigation, tirez la procédure à insérer par drag&drop à l'endroit adéquat dans le code.
   ou
3. Positionnez le curseur à l'endroit dans le code où vous voulez insérer la procédure.
4. Marquez la procédure dans la fenêtre de navigation avec la souris.
5. Sélectionnez la commande "Transférer l'appel de procédure" dans le menu contextuel.

Voir aussi

Création et édition de procédures (Page 40)

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)

Pour renommer une procédure ou un module (Page 55)

Pour enregistrer une procédure (Page 53)

Pour protéger un module par un mot de passe (Page 51)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour écrire le code d'une procédure (Page 45)

Pour créer une nouvelle procédure (Page 44)

Modules et procédures (Page 16)

1.9.5 Pour ajouter des informations de module supplémentaires

Introduction

Vous pouvez ajouter à chaque module des informations supplémentaires pour pouvoir ultérieurement rapidement identifier les fonctions du module ou des procédures qu'il contient. Si vous êtes plusieurs éditeurs à configurer un projet, il est recommandé de laisser aux collègues des informations de module supplémentaires.

Si vous créez un nouveau module, la date de création non modifiable s'inscrit automatiquement dans les informations supplémentaires du module. Le module reçoit en outre le numéro de version 1.0. Vous pouvez attribuer individuellement les numéros de version lorsque vous éditez un module. Lorsque vous modifiez et enregistrez un module, la date de modification non modifiable s'inscrit automatiquement.

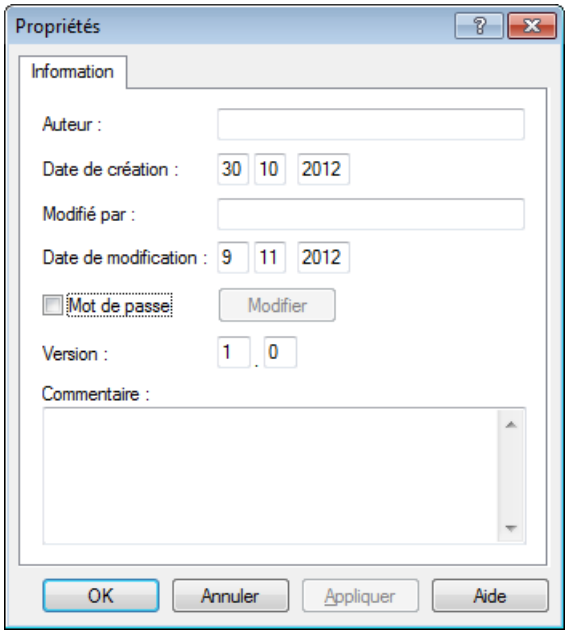
Vous pouvez laisser les informations supplémentaires suivantes :

- "Editeur"
- "Modifié par"
- "Commentaire" p. ex. fonctions du module/des procédures qu'il contient

Vous pouvez en outre taper ici un mot de passe pour le module. Vous trouverez sous "Protection d'un module par mot de passe" de plus amples informations concernant l'attribution des mots de passe.

Marche à suivre

1. Ouvrez Global Script.
2. Sélectionnez le module pour lequel vous voulez laisser des informations supplémentaires, dans la fenêtre de navigation.
3. Cliquez sur le bouton "Info/Trigger" dans la barre d'outils ou sélectionnez la commande "Info" dans le menu contextuel. Le dialogue "Propriétés " apparaît.



Tapez les informations voulues.

Remarque

Vous pouvez également appeler le dialogue "Info/Trigger" lorsque vous sélectionnez une procédure ouverte dans la fenêtre de navigation. Les informations que vous inscrivez dans ce dialogue sont cependant toujours valables pour l'ensemble du module ainsi que toutes les procédures qu'il contient.

Voir aussi

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)

Pour renommer une procédure ou un module (Page 55)

Pour enregistrer une procédure (Page 53)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour utiliser des procédures de projet et standard (Page 48)

Pour écrire le code d'une procédure (Page 45)

Pour créer une nouvelle procédure (Page 44)

Modules et procédures (Page 16)

Création et édition de procédures (Page 40)

1.9.6 Pour protéger un module par un mot de passe

Introduction

Vous pouvez protéger contre tout accès non autorisé un module par un mot de passe. Le mot de passe est un élément des informations supplémentaires du module.

Remarque

Lorsque vous protégez un module par un mot de passe, vous protégez également par ce même mot de passe toutes les procédures qu'il contient.

Marche à suivre

1. Ouvrez Global Script.
2. Sélectionnez dans la fenêtre de navigation le module que vous voulez protéger par un mot de passe.
3. Appuyez sur le bouton "Info/Trigger" dans la barre d'outils ou sélectionnez la commande "Info" dans le menu contextuel. Le dialogue "Propriétés " apparaît.
4. Activez la case à cocher "Mot de passe".
   Le dialogue "Saisie du mot de passe" apparaît.

5. Tapez le mot de passe puis validez-le.
6. Validez vos entrées avec OK.

Résultat

Le mot de passe vous sera demandé si vous ouvrez le module ou l'une des procédures qu'il contient.

Désactiver la protection par mot de passe

Pour annuler la protection par mot de passe, désactivez la case "Mot de passe".

Changer mot de passe

Si vous voulez modifier le mot de passe, appelez le dialogue Propriétés et appuyez sur le bouton "Modifier". Tapez maintenant votre nouveau mot de passe.

Remarque

Vous ne pouvez plus éditer un module si vous en avez oublié le mot de passe.

Remarque

Vous pouvez également appeler le dialogue "Info/Trigger" lorsque vous avez sélectionné une procédure dans la fenêtre de navigation. Les informations que vous inscrivez dans ce dialogue sont cependant toujours valables pour l'ensemble du module ainsi que toutes les procédures qu'il contient.

Voir aussi

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)

Pour renommer une procédure ou un module (Page 55)

Pour enregistrer une procédure (Page 53)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour utiliser des procédures de projet et standard (Page 48)

Pour écrire le code d'une procédure (Page 45)

Pour créer une nouvelle procédure (Page 44)

Modules et procédures (Page 16)

Création et édition de procédures (Page 40)

1.9.7 Pour enregistrer une procédure

Introduction

Ce n'est jamais une procédure individuelle que vous enregistrez, mais plutôt le module dans lequel vous avez programmé la procédure.

Avant d'enregistrer un module, vérifiez si la syntaxe du code est correcte. Lors de l'enregistrement d'un module, les procédures qui y sont contenues sont automatiquement contrôlées et en cas d'erreur de syntaxe il vous est demandé si vous voulez ou non enregistrer le module erroné. Cela vous permet par exemple d'écrire des modules et leurs procédures alors qu'ils ne sont pas encore entièrement programmés. Les procédures ayant des erreurs de syntaxe ne fonctionnent pas en Runtime.

Remarque

Si un module contient une procédure à syntaxe incorrecte, il ne peut plus être chargé. Il n'est plus possible d'appeler une procédure depuis le module.

Remarque

Le contrôle de syntaxe ne peut détecter que les erreurs syntaxiques dans le code. Des erreurs de programmation, comme par ex. des références manquantes, ne sont reconnues qu'en Runtime. C'est pourquoi vous devez toujours vérifier vos scripts, même en environnement Runtime, et au besoin utiliser un débogueur pour identifier et éliminer les erreurs.

Dans Runtime, seuls les modules de syntaxe correcte sont appelés.

Vous trouverez sous "Principes de base de VBScript" une liste des erreurs de syntaxe possibles.

Lorsque vous vérifiez avant l'enregistrement si une procédure ne contient pas d'erreur de syntaxe, les erreurs éventuelles s'affichent en bas de la fenêtre d'édition En cliquant deux fois sur une ligne erronée, vous parvenez directement à l'endroit erroné dans le code.

Utilisez la commande "Enregistrer sous", si vous voulez enregistrer le module sous un autre nom. Notez que le nouveau module ne s'affichera qu'après la mise à jour de la vue dans la fenêtre de navigation.

Condition

La procédure/le module à enregistrer doivent toujours être ouverts dans la fenêtre d'édition

Marche à suivre

1. Cliquez sur le bouton "Contrôle de syntaxe" dans la barre d'outils.
2. Si des erreurs de syntaxe s'affichent dans la fenêtre de sortie, cliquez deux fois sur la ligne erronée et corrigez l'erreur dans le code. Répétez les étapes 1 et 2 jusqu'à ce que le code soit correct.
3. Enregistrez le module avec le bouton "Enregistrer" dans la barre d'outils.

Remarque

Vous devez ouvrir la vue contenant une procédure modifiée dans Graphics Designer et l'enregistrer à nouveau

En cas de modification du module de projet, vous devez rouvrir et enregistrer la vue correspondante dans Graphics Designer, en plus de l'enregistrement dans l'éditeur VBS. La modification est ensuite active en Runtime. Une fois seulement que la vue est enregistrée, les informations sur les modules de projet requis sont adoptées dans le fichier de vue.

Voir aussi

Diagnostic (Page 83)

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)

Pour enregistrer une procédure (Page 53)

Pour protéger un module par un mot de passe (Page 51)

Pour ajouter des informations de module supplémentaires (Page 49)

Pour utiliser des procédures de projet et standard (Page 48)

Pour écrire le code d'une procédure (Page 45)

Pour créer une nouvelle procédure (Page 44)

Modules et procédures (Page 16)

Création et édition de procédures (Page 40)

1.9.8 Pour renommer une procédure ou un module

Introduction

Renommez une procédure ou un module dans les cas suivants:

- si vous changez le nom standard (procedure# ou Modul#) qui est automatiquement attribué lors de la création d'un nouveau module/d'une nouvelle procédure, pour employer un nom ayant un sens
- si vous copiez un module ou une procédure pour par exemple créer un nouveau module à partir d'un module existant déjà pour que son contenu soit similaire.
  Notez alors que les noms de procédure au sein d'un projet doivent toujours être univoques. Les noms de procédure existant deux fois sont affichés en tant qu'erreur lors du contrôle de syntaxe.
  IContrairement aux noms de procédure, vous pouvez utiliser des noms identiques pour les module si les modules en question se trouvent dans différents répertoires.

Remarque

Le nom de module est toujours identique au nom de fichier du module dans le système de fichiers WinCC. Lorsque vous modifiez un nom de module dans Windows-Explorer par exemple, le nouveau nom de module est accepté dans la fenêtre de navigation de Global Script.

Marche à suivre

Renommer la procédure

1. Ouvrez la procédure que vous voulez renommer.
2. Tapez le nouveau nom sur l'en-tête de la procédure.
3. Enregistrez la procédure pour que le nom soit validé dans la fenêtre de navigation. Les noms de procédure sont toujours univoques et ne doivent pas être utilisés plusieurs fois.

Renommer un module

1. Fermez le module que vous voulez renommer.
2. Sélectionnez le module dans la fenêtre de navigation puis la commande "Renommer" dans le menu contextuel.
3. Tapez le nouveau nom dans la fenêtre de navigation. Les noms de module au niveau répertoire sont toujours univoques et ne doivent pas être utilisés plusieurs fois.

Voir aussi

Procédez comme suit pour utiliser plusieurs fois des procédures et des actions (Page 22)
Pour enregistrer une procédure (Page 53)
Pour protéger un module par un mot de passe (Page 51)
Pour ajouter des informations de module supplémentaires (Page 49)
Pour utiliser des procédures de projet et standard (Page 48)
Pour écrire le code d'une procédure (Page 45)
Pour créer une nouvelle procédure (Page 44)
Modules et procédures (Page 16)
Création et édition de procédures (Page 40)

1.10 Création et édition d'actions

1.10.1 Création et édition d'actions

Introduction

Dans VBS, dans WinCC, il n'y a contrairement au système C aucune différence entre les actions locales (valables à l'échelle du projet) et globales (valables dans tout l'ordinateur). Une action configurée est toujours valable de façon globale.

Les actions copiées sont disponibles en Runtime après un redémarrage ou l'ouverture puis l ´enregistrement de l'action. Elle sont visibles dans l'éditeur une fois que l'affichage a été mis à jour.

Les actions VBS vous permettent de dynamiser en Runtime des objets graphiques et les propriétés de l'objet ou d'effectuer des actions ne dépendant pas de la vue.

Remarque

Notez que la longueur du nom des objets que vous dynamisez dans Graphics Designer est limitée à 200 caractères environ, et que dans les fichiers de script chaque caractère spécial que vous utilisez dans un nom d'objet représente cinq caractères. Après un X en première position, le caractère spécial est représenté par un code hexadécimal à quatre positions. Si le nom d'un objet dynamisé est trop long, un message d'erreur correspondant apparaît. Vous obtiendrez de plus amples informations dans cette aide sous "Structure des fichiers VBScript".

Remarque

Si vous dynamisez une propriété d'objet avec une action VBS par le biais de la valeur retour d'un script, la valeur de la propriété d'objet ne sera inscrite que si elle a changé par rapport au dernier passage du script. Peu importe alors si la valeur a été modifiée depuis un autre endroit.

C'est pourquoi les propriétés qui ont été dynamisées avec une action VBS par le biais de la valeur retour, ne sont pas modifiées d'un autre endroit (autres scripts C ou VBS par exemple).

Si vous ne prenez pas ce fait en considération, il se peut que les valeurs affichées soient erronées.

Utilisation des actions

Vous pouvez utiliser des actions en procédant comme suit :

Sur des objets graphiques dans Graphics Designer

En tant que dynamisation d'une propriété (action avec valeur de retour), p. ex. :

```
Function BackColor_Trigger(ByVal Item)
'VBS143
```

```
    BackColor_Trigger = RGB(125,0,0)
End Function
```

Déclenchées par un événement pour un objet (action sans valeur de retour), p. ex. :

```
Sub OnClick(ByVal Item)
'VBS144
    Item.BackColor = RGB(255,0,0)
End Sub
```

Indépendamment de la vue dans Global Script

En tant qu'action cyclique, p. ex. le comptage d'une variable :

```
Option Explicit
Function action
'VBS145
    Dim objTag1
    Dim lngValue
    Set objTag1 = HMIRuntime.Tags("Tag1")
    lngValue = objTag1.Read
    objTag1.Write lngValue + 1
    action = CLng(objTag1.value)
End Function
```

Exécution des actions

Vous pouvez attribuer plusieurs Trigger à une action. L'action sera toujours exécutée lorsque l'un des événements de déclenchement se produit. Prenez en considération les particularités suivantes :

- Il est impossible d'exécuter simultanément des actions dans Global Script. La dernière action déclenchée est maintenue en boucle d'attente jusqu'à ce que l'action en cours soit terminée.
- Dans Graphics Designer, il est impossible d'exécuter simultanément des actions cycliques et des actions déclenchées par une variable. Si l'exécution d'une action déclenchée par une variable gêne une action cyclique, cette dernière sera à nouveau exécutée dès la fin de l'action déclenchée par une variable. L'action cyclique n'est pas mise en boucle d'attente en cas de non-exécution. Une fois l'action en cours terminée, l'action cyclique est à nouveau exécutée en cycle normal.
- Dans Graphics Designer, il est impossible d'exécuter simultanément des actions déclenchées par événement.

Les types d'action mentionnés ne se gênent pas mutuellement lors de l'exécution : L'exécution d'actions dans Global Script n'a aucune influence sur les actions dans Graphics Designer. De

même, l'exécution d'actions cycliques ou déclenchées par une variable dans Graphics Designer n'a aucune influence sur l'exécution d'actions déclenchées par événement.

Remarque

Le système termine les actions si elles fonctionnent encore dans les vues alors que ces dernières ont été désactivées depuis plus d'une minute. Cela s'affiche dans l'inscription Logfile.

Recherche des actions

Vous pouvez afficher toutes les actions utilisées dans une vue via les propriétés de la vue. Sélectionnez à cet effet la vue dans l'explorateur WinCC, puis la commande "Propriétés" dans le menu contextuel. Vous obtiendrez des informations plus détaillées concernant le type de dynamisation en cliquant deux fois sur une inscription.

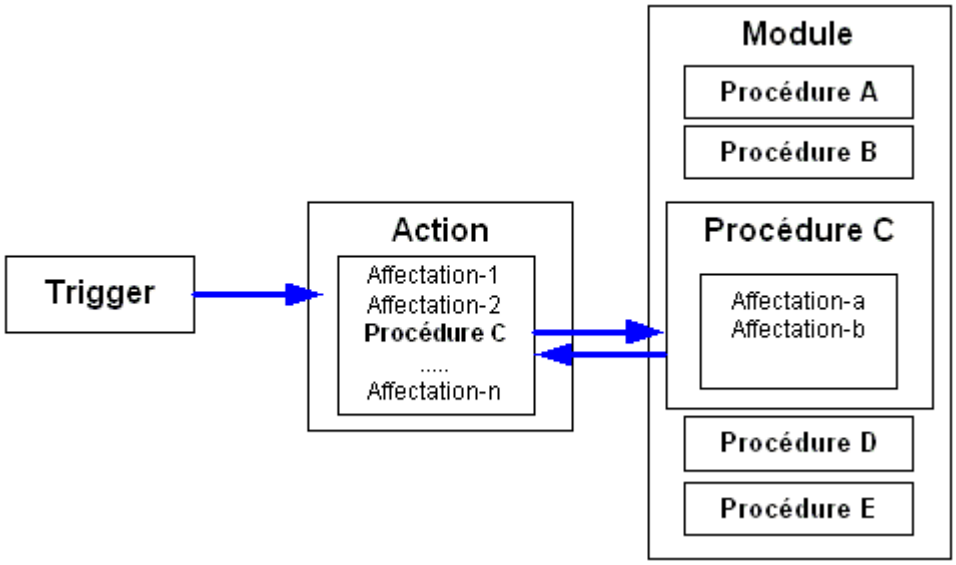
Vous pouvez en outre afficher toutes les variables et vues utilisées dans les actions via WinCC CrossReference. CrossReference vous permet également de réaffecter aisément des liaisons de variable des actions de Graphics Designer.

Remarque

Utilisez les formules standard suivantes pour l'adressage des vues et des variables dans votre code
HMIRuntime.BaseScreenName = "Screenname" et HMIRuntime.Tags("Nom de variable"), pour garantir que les vues et les variables seront correctement saisies par CrossReference.

Restriction Action - Procédure

Vous pouvez programmer des instructions et appeler des procédures dans les actions. Dans les procédures, programmez le code qui est utilisé en plusieurs endroits de votre configuration. Contrairement aux procédures, les actions possèdent toujours un Trigger.



Création et édition d'actions

Vous pouvez configurer les actions dans Global Script et dans Graphics Designer. Dans Global Script, vous configurez des actions globales qui doivent être exécutées indépendamment de la vue ouverte. Dans Graphics Designer, vous configurez des actions pour des objets graphiques et qui sont exécutées lorsque la vue est ouverte en Runtime ou lorsque le déclenchement configuré intervient.

Les éditeurs Script dans WinCC vous offre la possibilité de contrôler si la syntaxe de vos scripts est correcte, cela sans les exécuter. Les erreurs dans le script s'affichent dans la fenêtre de sortie sous la fenêtre d'édition. Cliquez deux fois sur la ligne d'erreur pour parvenir directement à l'endroit respectif dans le code.

Remarque

Le contrôle de syntaxe ne peut contrôler que les objets que le système connaît au moment de leur exécution. Les variables et les objets auxquels vous accédez dans vos scripts doivent donc être créés dans WinCC.

Dans Runtime, seules les actions de syntaxe correcte sont exécutées.

Les objets d'automatisation "PDLRuntime" et "WinCC-Runtime-Projet" ne peuvent pas être utilisés dans des actions VBS.

Représentation des actions

Si vous enregistrez une action à syntaxe erronée, elle s'affiche dans la fenêtre de navigation de Global Script accompagnée du symbole ci-contre.

Si vous enregistrez une action à syntaxe correcte sans déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script accompagnée du symbole ci-contre.

Si vous enregistrez une action à syntaxe correcte avec déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script accompagnée du symbole ci-contre.

Remarque

Dans Graphics Designer, il vous est uniquement possible de valider des actions de syntaxe correcte. Si vous gardez malgré tout une action comportant des erreurs et que vous voulez la quitter, il vous faut auparavant tout commenter.

Comportement du système lorsque des actions sont modifiées, supprimées ou enregistrées dans Runtime.

Si une action locale est enregistrée pendant Runtime, toutes les actions locales et globales de l'ordinateur sur lequel cette action s'exécute sont réinitialisées sur cet ordinateur.

Si une action globale est enregistrée pendant Runtime, toutes les actions locales et globales du projet entier et donc de tous les ordinateurs sont réinitialisées.

La réinitialisation réinitialise également par exemple les variables et les horaires utilisés dans les actions comme déclenchement, et l'action se déclenche alors.

Les variables statistiques utilisées dans les actions réinitialisées sont elles aussi réinitialisées.

Voir aussi

Structure des fichiers VBScript (Page 97)

Nom de l'action et de la procédure dans Debugger (Page 99)

Pour renommer une action (Page 80)

Pour enregistrer une action (Page 69)

Pour protéger une action par un mot de passe (Page 68)

Pour ajouter des informations supplémentaires à l'action (Page 66)

Pour éditer une action (Page 63)

Pour créer une nouvelle action (Page 62)

Déclenchement (Page 71)

Actions (Page 19)

1.10.2 Pour créer une nouvelle action

Introduction

Lors de la création d'une nouvelle action, l'éditeur vous propose automatiquement un nom de fichier (Action#.bac) que vous pouvez par la suite modifier.

Vous pouvez configurer les actions dans Global Script et dans Graphics Designer.

- Vous configurez dans Global Script les actions qui doivent être exécutées en Runtime indépendamment de la vue. Vous ouvrez Global Script via l'explorateur WinCC.
- Dans Graphics Designer, vous configurez une nouvelle action pour les propriétés d'un objet graphique en appuyant sur la touche droite de la souris, onglet "Propriétés", colonne "Dynamique" et sélectionnez l'inscription "Action VBS". Vous configurez de la même manière une action pour un événement sur l'onglet Evénement.
  Dans les deux cas, l'éditeur d'action du Graphics Designer s'ouvre.

Remarque

Vous trouverez sous WinCC - Rubrique Aide "Dynamisation", la procédure détaillée pour lier des actions aux objets graphiques.

Marche à suivre

1. Ouvrez Global Script.
2. Activez l'onglet "Actions" dans la fenêtre de navigation.
3. Appuyez sur le bouton ci-contre dans la barre d'outils ou sélectionnez la commande "Fichier" > "Nouveau" > "Action".
   Une nouvelle action s'ouvre dans la fenêtre d'édition. L'action apparaît dans la fenêtre de navigation, si vous l'avez enregistrée.

Remarque

Lors de la création d'une nouvelle action, la commande "Option explicit" non supprimable s'inscrit automatiquement dans la zone de déclaration. La commande est nécessaire pour éviter des erreurs par faute d'orthographe dans les variables sans déclaration.

L'instruction vous invite à toujours définir les variables avec l'instruction "Dim" dans votre code.

N'utilisez pas la commande "Option explicit" dans votre code, sinon il pourrait y avoir des erreurs d'exécution.

Voir aussi

Pour renommer une action (Page 80)

Pour enregistrer une action (Page 69)

Pour protéger une action par un mot de passe (Page 68)

Pour ajouter des informations supplémentaires à l'action (Page 66)

Pour éditer une action (Page 63)

Déclenchement (Page 71)

Création et édition d'actions (Page 57)

Actions (Page 19)

1.10.3 Pour éditer une action

Introduction

Vous éditez une action comme une procédure dans la fenêtre d'édition de l'éditeur ou dans l'éditeur d'actions de Graphics Designer.

Vous avez besoin d'un Trigger pour qu'une action puisse être exécutée en Runtime. Les actions qui sont déclenchées par un événement dans Graphics Designer n'ont pas besoin de Trigger.

Si vous modifiez une action en Runtime, la modification sera validée lorsque la vue sera à nouveau chargée (pour les actions dans Graphics Designer) ou lorsque l'action sera à nouveau appelée (pour les actions dans Global Script).

Remarque

Une modification dans le code ne peut pas être validée en Runtime si une autre action est en train d'être exécutée.

Vous pouvez insérer dans l'action un appel de procédure en faisant glisser cette dernière de la fenêtre de navigation de l'éditeur par "drag&drop" pour la placer à l'endroit adéquat du code dans la fenêtre d'édition. Vous ne pouvez pas appeler les Scripts C dans les actions VBS.

Zone de déclaration dans les actions

Quand vous créez des actions dans le Graphics Designer, vous pouvez afficher la zone de déclaration de l'action avec le bouton . Lors de la création d'une nouvelle action, la commande "Option explicit" non supprimable s'inscrit automatiquement dans la zone de déclaration. La commande est nécessaire pour éviter des erreurs par faute d'orthographe dans les variables sans déclaration.

L'instruction vous invite à toujours définir les variables avec l'instruction "Dim" dans votre code.

N'utilisez pas l'instruction "Option explicit" dans votre code, car sinon, il peut se produire des erreurs d'exécution.

Dans la zone de déclaration, vous pouvez en plus fixer des définitions générales que vous voulez utiliser globalement pour la vue courante, par ex. :

- définitions de variables
- procédures que vous ne désirez utiliser que dans cette vue

Dans les zones de déclaration des actions, vous pouvez définir des variables globales indépendantes les unes des autres dans les zones "Evénement" et "Propriétés" d'un objet. Dans ces deux zones, il n'existe aucune relation entre des variables globales de même nom.

Remarque

Définissez toujours les procédures dans la zone de déclaration avec la syntaxe correcte, avec "Sub" - "End Sub". Ne créez pas de code directement exécutable dans la zone de déclaration étant donné que cela entraîne des erreurs d'exécution.

Si vous utilisez des variables globales dans la zone de déclaration des actions dans Graphics Designer, notez que les actions déclenchées par événement et les actions cycliques/ déclenchées par une variable sont traitées séparément dans le mode Runtime. Dans Runtime, aucune synchronisation des variables globales n'est prévue entre ces deux systèmes Runtime. Si une synchronisation des variables s'avère nécessaire, configurez-la par l'objet DataSet ou par des variables internes WinCC.

Lors des procédures de définition dans la zone "Déclaration", prenez en considération la structure des fichiers Script comme décrit sous "Structure des fichiers VBScript".

Fonctions permettant d'éditer des actions

Les éditeurs de script vous assistent dans la création des codes d'action avec les fonctions suivantes :

Intellisense et mise en relief de la syntaxe

Pendant l'entrée apparaissent des listes contextuelles, dans lesquelles vous sont proposés les propriétés, les méthodes et les objets possibles à l'emplacements de code actuel. Quand vous insérez un élément de la liste, la syntaxe nécessaire vous est spécifiée automatiquement.

Remarque

L'Intellisense complète pour tous les objets n'est utilisable dans le Graphics Designer que quand l'accès est effectué par le nom d'objet sur la liste et que le résultat est affecté à une variable. Sinon, le système n'affiche que les propriétés standard.

Exemple d'Intellisense complète :

```
Dim Variable
Set Variable = ScreenItems ("Cercle1")

Variable.<Intellisense>
```

Quand, lors de l'adressage, des limites de la fenêtre de vue sont dépassées, seules les propriétés standard sont également proposées, étant donné que la vue de la fenêtre de vue n'est pas chargée.

Fonctions générales VBS

Vous pouvez afficher une liste de fonctions générales VBS via la commande contextuelle "Liste de fonctions" dans la fenêtre d'édition.

Listes des objets, des propriétés et des méthodes

Par le menu contextuel dans la fenêtre d'édition, vous pouvez faire afficher une liste des objets possibles avec la commande "Liste d'objets" dans le Graphics Designer. Dans Global Script, vous n'obtenez dans cette liste que l'objet "HMIRuntime", étant donné que vous ne pouvez pas directement accéder aux objets de Graphics Designer.

A l'aide de la commande contextuelle "Propriétés/Méthodes", vous obtenez une liste des propriétés et méthodes possibles.

Vous obtenez les mêmes listes, selon le contexte, dans le Script avec la combinaison de touches <Ctrl+Espace>.

Modèles de code

Dans la fenêtre de navigation de l'éditeur sur l'onglet Modèles de code, vous trouverez une sélection d'instructions souvent utilisées, par ex. pour des boucles et des instructions conditionnées. Vous pouvez insérer ces modèles par "drag&drop" dans votre code de procédure.

Lorsque vous insérez un modèle de code dans votre code, vous devez remplacer les caractères génériques "_XYZ_" des modèles par les données correspondantes.

Dialogues de sélection

Si vous utilisez des variables ou des objets WinCC dans le code, les dialogues de sélection suivants sont à votre disposition :

- Ouvre un dialogue de sélection pour variables et donne le nom de variable sélectionné comme valeur de retour.
- Ouvre un dialogue de sélection pour variables et donne en retour le nom de variable avec la référence correspondante.
- Ouvre un explorateur de vues/d'objets où il est possible de sélectionner une vue/un objet dont le nom est restitué comme valeur de retour.
- Ouvre un dialogue de sélection pour les vues et restitue le nom de vue, au besoin accompagné d'un préfixe Serveur.

Contrôle de syntaxe

Global Script vous assiste par un contrôle de syntaxe que vous pouvez effectuer après avoir créé le code. Des erreurs syntaxiques dans le code vous sont affichées dans la fenêtre de

sortie de l'éditeur. Vous accédez directement à l'emplacement de l'erreur dans le code par un double-clic sur l'erreur dans la fenêtre de sortie.

Remarque

Le contrôle de syntaxe ne peut détecter que les erreurs syntaxiques dans le code. Des erreurs de programmation, comme par ex. des références manquantes, ne sont reconnues qu'en Runtime. C'est pourquoi vous devez toujours vérifier vos scripts, même en environnement Runtime, et au besoin utiliser un débogueur pour identifier et éliminer les erreurs. Vous trouverez dans cette documentation, sous la commande "Diagnostic" > "Tester avec Debugger" comment tester les scripts avec un débogueur.

Marche à suivre

1. Ouvrez Global Script.
2. Cliquez deux fois l'action sur l'onglet "Actions" dans la fenêtre de navigation.
3. Editez l'action.

Voir aussi

Utilisation de variables globales dans VBS (Page 26)
Tester avec Debugger (Page 90)
Structure des fichiers VBScript (Page 97)
Pour renommer une action (Page 80)
Pour enregistrer une action (Page 69)
Pour protéger une action par un mot de passe (Page 68)
Pour éditer une action (Page 63)
Pour créer une nouvelle action (Page 62)
Déclenchement (Page 71)
Création et édition d'actions (Page 57)
Actions (Page 19)

1.10.4 Pour ajouter des informations supplémentaires à l'action

Introduction

Dans Global Script, vous pouvez ajouter des informations supplémentaires à chaque action pour pouvoir ultérieurement identifier rapidement la fonction de l'action. Si vous êtes plusieurs éditeurs à configurer un projet, il est recommandé de laisser aux collègues des informations supplémentaires de l'action.

Lorsque vous créez une nouvelle action, la date de création s'inscrit automatiquement et de façon non modifiable dans les informations supplémentaires de l'action. Le numéro de version 1.0 est en outre attribué à l'action. Vous pouvez attribuer individuellement les numéros de version lors de l'édition d'une action. Lorsque vous modifiez et enregistrez une action, la date de modification actuelle est automatiquement inscrite de façon non modifiable.

Vous pouvez laisser les informations supplémentaires suivantes:

- "Editeur"
- "Modifié par"
- "Commentaire" par exemple les fonctions de l'action

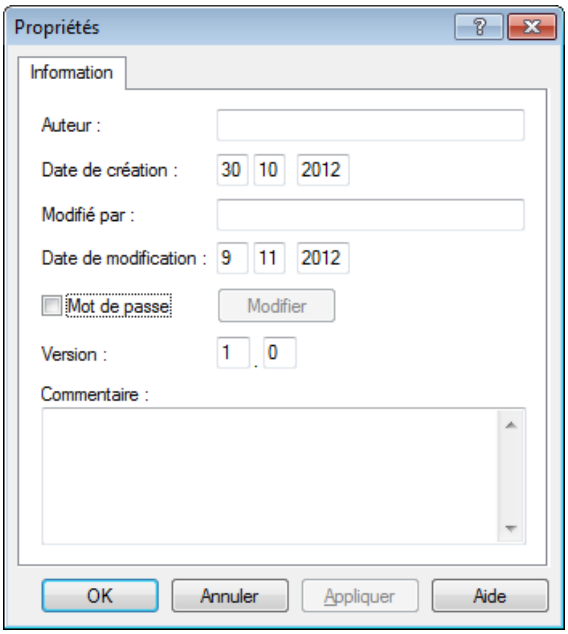
Vous pouvez en outre taper ici un mot de passe pour l'action. Vous trouverez sous "Protection d'une action par mot de passe" de plus amples informations concernant l'attribution des mots de passe.

Remarque

Vous pouvez uniquement apporter des informations supplémentaires aux actions dans Global Script, pas aux actions dans Graphics Designer.

Marche à suivre

1. Ouvrez Global Script.
2. Ouvrez l'action pour laquelle vous voulez apporter des informations supplémentaires
3. Appuyez sur le bouton "Info/Trigger" dans la barre d'outils ou sélectionnez la commande "Info/Trigger" dans le menu contextuel. Le dialogue "Propriétés " apparaît.



4. Tapez les informations voulues.

Voir aussi

Pour renommer une action (Page 80)
Pour enregistrer une action (Page 69)
Pour protéger une action par un mot de passe (Page 68)
Pour éditer une action (Page 63)
Pour créer une nouvelle action (Page 62)
Déclenchement (Page 71)
Création et édition d'actions (Page 57)
Actions (Page 19)

1.10.5 Pour protéger une action par un mot de passe

Introduction

Pour éviter tout accès non autorisé, vous pouvez protéger une action dans Global Script à l'aide d'un mot de passe. Le mot de passe est un élément des informations supplémentaires de l'action.

Remarque

Vous pouvez uniquement protéger par mot de passe les actions dans Global Script, pas les actions dans Graphics Designer.

Marche à suivre

1. Ouvrez Global Script.
2. Ouvrez l'action que vous voulez protéger par mot de passe.
3. Appuyez sur le bouton "Info/Trigger" dans la barre d'outils ou sélectionnez la commande "Info/Trigger" dans le menu contextuel. Le dialogue "Propriétés " apparaît.
4. Activez la case "Mot de passe"
5. Appuyez sur le bouton "Modifier". La fenêtre "Saisie du mot de passe" apparaît.
6. Tapez le mot de passe puis validez-le.
7. Validez vos entrées avec OK.

Résultat

Si l'action est ouverte, le mot de passe est tout d'abord exigé.

Désactiver la protection par mot de passe

Pour annuler la protection par mot de passe, désactivez la case "Mot de passe".

Changer mot de passe

Si vous voulez modifier le mot de passe, appelez le dialogue Propriétés et appuyez sur le bouton "Modifier". Tapez maintenant votre nouveau mot de passe.

Remarque

Vous ne pouvez plus éditer une action si vous en avez oublié le mot de passe.

Voir aussi

Pour renommer une action (Page 80)

Pour enregistrer une action (Page 69)

Pour ajouter des informations supplémentaires à l'action (Page 66)

Pour éditer une action (Page 63)

Pour créer une nouvelle action (Page 62)

Déclenchement (Page 71)

Création et édition d'actions (Page 57)

Actions (Page 19)

1.10.6 Pour enregistrer une action

Introduction

Une action doit être enregistrée avant que vous ne puissiez l'exécuter en Runtime. Enregistrez l'action tout comme les autres fichiers Windows avec la commande "Fichier" > "Enregistrer" ou avec l'icône correspondant.

Remarque

Les actions dans Graphics Designer sont automatiquement validées avec la vue lorsque vous quittez l'éditeur d'actions. Dans Graphics Designer, il vous est uniquement possible de valider des fonctions de syntaxe correcte. Si vous gardez malgré tout une action comportant des erreurs et que vous voulez la quitter, il vous faut auparavant tout commenter.

Vous trouverez sous "Principes de base de VBScript" une liste des erreurs de syntaxe possibles.

Si vous voulez enregistrer une action sous un autre nom, par exemple pour utiliser cette action comme base pour une autre action, utilisez la commande "Enregistrer sous".

Notez que lorsque vous utilisez "Enregistrer sous", seul le nom de fichier est modifié, pas le nom de l'action.

Avant l'enregistrement

Avant d'enregistrer une action, vérifiez si la syntaxe du code est correcte. Les erreurs de syntaxe dans le code sont indiquées dans la fenêtre de sortie de Global Script. En cliquant deux fois sur une ligne erronée, vous parvenez à l'endroit respectif dans le code.

Remarque

Le contrôle de syntaxe ne peut détecter que les erreurs syntaxiques dans le code. Des erreurs de programmation, comme par ex. des références manquantes, ne sont reconnues qu'en Runtime. C'est pourquoi vous devez toujours vérifier vos scripts, même en environnement Runtime, et au besoin utiliser un débogueur pour identifier et éliminer les erreurs.

Si vous voulez enregistrer des actions sans contrôle préalable de la syntaxe, l'éditeur vous prévient que vous enregistrez une action de syntaxe incorrecte et qu'elle ne pourra pas être exécutée en Runtime.

Les actions dont la syntaxe est erronée s'affichent dans la fenêtre de navigation accompagnées du symbole ci-contre.

Marche à suivre

1. Cliquez sur le bouton "Contrôle de syntaxe" dans la barre d'outils.
2. Si des erreurs s'affichent en bas de la fenêtre d'édition, cliquez deux fois sur la ligne erronée et corrigez l'erreur dans le code. Répétez les étapes 1 et 2 jusqu'à ce que le code soit correct.
3. Enregistrez l'action avec le bouton "Enregistrer" dans la barre d'outils.

Voir aussi

Actions (Page 19)

Pour renommer une action (Page 80)

Pour protéger une action par un mot de passe (Page 68)

Pour ajouter des informations supplémentaires à l'action (Page 66)

Pour éditer une action (Page 63)

Pour créer une nouvelle action (Page 62)

Déclenchement (Page 71)

Création et édition d'actions (Page 57)

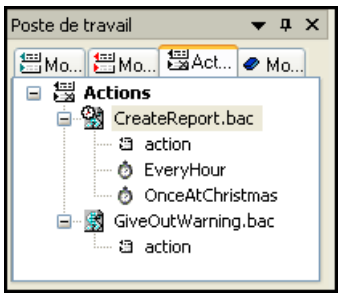
1.10.7 Déclenchement

1.10.7.1 Déclenchement

Définition et utilisation

Des déclenchements sont utilisés pour exécuter les actions en Runtime. A cet effet, un déclenchement est mis en corrélation avec une action et constitue ainsi l'événement qui déclenchera l'appel de cette action. Les actions sans déclenchement ne sont pas exécutées.

Les Triggers définis pour une action s'affichent dans la fenêtre de navigation de Global Script.



Types de déclenchement

Les déclenchements suivants sont à disposition :

Déclenchements acycliques

Ils sont constitués d'une date et d'une heure. L'action liée à un trigger de ce type sera exécutée, une fois, à la date et à l'heure paramétrées.

Déclenchement cyclique

Ils sont constitués d'un intervalle de temps et de l'instant du début d'exécution. Les triggers cycliques suivants sont disponibles:

- Cycle standard. Le début du premier intervalle de temps coïncide avec le démarrage du Runtime. La longueur de l'intervalle de temps est déterminée par le cycle.
- Heure. Le début de l'intervalle de temps est indiqué en minutes et secondes. La longueur de l'intervalle est d'une heure.
- Jour. Le début de l'intervalle de temps est paramétré par indication de l'heure (heure, minute et seconde). La longueur de l'intervalle est d'un jour.
- Semaine. Le début de l'intervalle de temps est paramétré par indication du jour de la semaine (lundi, mardi...) et de l'heure. La longueur de l'intervalle est d'une semaine.
- Mois. Le début de l'intervalle de temps est indiqué par le jour et l'heure. La longueur de l'intervalle est d'un mois.
- An. Le début de l'intervalle de temps est indiqué par le jour, le mois et l'heure. La longueur de l'intervalle est d'un an.

Utilisez les Triggers synchronisés pour les actions dans Global Script et pour les actions servant à la dynamisation des objets graphiques.

Déclenchement par variable

Ils sont constitués d'une ou de plusieurs variables. L'action, liée à un tel déclenchement est exécutée à chaque fois qu'il se produit une modification de la valeur de l'une de ces variables.

La manière dont les valeurs de variable sont scrutées est paramétrable individuellement pour chaque variable. Vous pouvez choisir entre les modes suivants:

- interrogation cyclique de la valeur de variable: vous spécifiez le cycle standard. La valeur de la variable est interrogée dans l'intervalle choisi (toutes les 2 secondes par exemple). L'action est déclenchée quand le système constate que la valeur de la variable a changé. Selon la fréquence du cycle, il se peut que la variable change mais que le système ne le remarque pas.
  Si vous réglez par exemple un cycle de 5 minutes et que la valeur de la variable change plusieurs fois au cours de ces 5 minutes, seule la valeur actuelle lors de l'interrogation suivante sera prise en compte. Les changements de valeur entre les deux interrogations ne seront pas pris en compte.
- En cas de changement de valeur de variable: Le système prend en compte chaque changement de valeur de variable. L'action est exécutée à chaque fois que sa valeur change.

Vous pouvez utiliser les à variable(s) pour les actions dans Global Script et pour les actions servant à la dynamisation des objets graphiques.

Trigger d'événement

Si vous configurez une action pour un événement pour un objet graphique, l'action se déclenche par événement, par exemple lors du clic de souris ou lors de changement de la couleur d'arrière-plan par une autre action.

Cycle d'animation

A partir de WinCC V7.0, vous disposez du mode de déclenchement "Cycle d'animation" pour la dynamisation d'objets avec VBS. Le cycle d'animation vous permet d'activer et de désactiver des actions ainsi que de modifier le temps de déclenchement dans Runtime.

Vous trouverez de plus amples informations à la rubrique "Déclenchement d'animation".

Répercussion des Triggers sur les actions

Si l'action n'est liée qu'à un seul déclenchement, elle est exécutée dès que l'événement se produit.

Une action peut cependant être conditionnée par plusieurs événements comme un déclenchement cyclique et la modification d'une variable. L'action est alors exécutée dès que l'un des événements se produit. Si deux événements se produisent simultanément, l'action est exécutée deux fois de suite. Si deux variables sont modifiées simultanément, l'action n'est exécutée qu'une fois.

Traitement des actions dans Graphics Designer

Dans Graphics Designer, les règles suivantes sont valables pour le traitement des actions:

- Aucune action déclenchée par événement ne peut être exécutée tant qu'une autre action déclenchée par événement est en cours.
- Aucune action cyclique/déclenchée par une variable ne peut être exécutée tant qu'une autre action cyclique/déclenchée par une variable est en cours.
- Les deux types d'action ne s'influencent pas mutuellement: Une action commandée par événement peut être exécutée même si une action cyclique est en cours.
- Lorsque l'exécution d'actions est bloquée par d'autres actions (une action cyclique par exemple bloquée par une action déclenchée par une variable), toutes les actions bloquées seront exécutées une fois dès que possible. Les actions cycliques continuent de fonctionner dans leur cycle normal après qu'elles aient été exécutées une fois.

Traitement d'actions dans Global Script

Les actions de Global Script ne dépendant pas des vues sont exécutées les unes à la suite des autres dans Runtime lors du déclenchement. Lorsqu'une action est déclenchée, alors qu'une autre est justement en cours, la deuxième action est placée en boucle d'attente jusqu'à ce que son exécution soit possible.

Les actions dans Global Script et Graphics Designer ne s'influencent pas mutuellement.

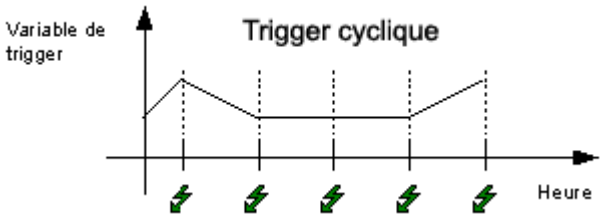
Remarque

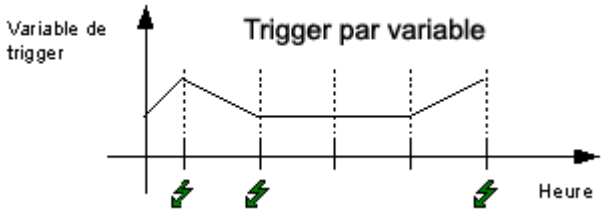
Si l'action ne doit pas être exécutée à chaque événement, vous pouvez formuler dans l'action une condition commandant la suite de l'exécution de l'action en fonction du résultat du test.

Remarque concernant la configuration de Triggers

Pour des raisons internes au système, il n'est pas possible de garantir qu'une action à déclenchement cyclique soit exécutée exactement aux instants spécifiés. Si cette garantie est nécessaire, cette tâche (tests de vérifications, par exemple) doit être effectuée en usine.

Il est préférable d'opter pour les Triggers à variable(s) plutôt que pour les Triggers cycliques: Lors des actions cycliques, l'action est toujours exécutée, toutes les 20 secondes par exemple. Le Trigger à variable(s) n'exécute l'action que lorsqu'un changement de valeur de variable a été constaté lors de l'interrogation cyclique. Le système est ainsi moins sollicité et les performances augmentent.





Lorsque vous utilisez un Trigger à variable(s), configurez aussi rarement que possible le cycle "Sur modification". Ce cycle d'interrogation fait en sorte que la variable déclenche une action à chaque modification. Le système serait alors trop sollicité.

Réaffectation des Triggers à variable(s)

CrossReference de WinCC vous permet de retrouver rapidement tous les occurrences de variables, même dans des actions VBS. Vous pouvez réaffecter avec CrossReference les Trigger de variable provenant des actions dans Graphics Designer, c'est-à dire les remplacer par d'autres variables à tous les emplacements ou seulement sur des emplacements choisis.

Remarque

Vous pouvez également réaffecter des variables directement dans Graphics Designer en sélectionnant l'objet graphique, puis la commande "Réaffecter..." dans le menu contextuel.

Utilisez les formules standard suivantes pour l'adressage des vues et des variables dans votre code
`HMIRuntime.BaseScreenName = "Screenname"` et
`HMIRuntime.Tags("Tagname")`, pour garantir que les vues et les variables seront correctement saisies par CrossReference.

Vous trouverez de plus amples renseignements dans la documentation WinCC concernant CrossReference.

Voir aussi

Actions (Page 19)

Pour supprimer un déclenchement (Page 79)

Pour modifier un déclenchement (Page 78)

Pour ajouter un Trigger du type "Variable" (Page 77)

Pour ajouter un Trigger du type "Temporisateur" (Page 76)

Création et édition d'actions (Page 57)

1.10.7.2 Déclenchement d'animation

Introduction

A partir de WinCC V7.0, vous disposez du mode de déclenchement "Cycle d'animation" pour la dynamisation d'objets avec VBS. Le cycle d'animation permet d'activer et de désactiver des actions ainsi que de modifier le temps de déclenchement dans Runtime.

Cycles d'animation

Nom	Cycle	Nom	Cycle
CycleTime125ms	125 ms	CycleUser1	Cycle utilisateur 1
CycleTime250ms	250 ms	CycleUser2	Cycle utilisateur 2
CycleTime500ms	500 ms	CycleUser3	Cycle utilisateur 3
CycleTime1s	1 s	CycleUser4	Cycle utilisateur 4
CycleTime2s	2 s	CycleUser5	Cycle utilisateur 5
CycleTime5s	5 s	CyclePicture	Cycle vue
CycleTime10s	10 s	CycleWindow	Cycle de fenêtre
CycleTime1min	1 min		
CycleTime5min	5 min		
CycleTime10min	10 min		
CycleTime1h	1 h		

Vous utilisez le déclenchement en écrivant une action et en en utilisant le mode de déclenchement "Cycle d'animation". Avec les méthodes "ActivateDynamic" et "DeactivateDynamic" cette action peut être activée et désactivée dans Runtime. Les méthodes sont décrites dans la référence VBS de WinCC Information System. La syntaxe correcte des méthodes diverge de la description dans la référence VBS est et représentée dans les deux exemples suivants.

Exemple

Avec une action sur la propriété fixe "Position X" (Left), le carré est décalé de 5 pixels vers la droite. Comme déclenchement, sélectionnez l'événement "Cycle d'animation" dans l'action.

Dans la propriété "Left", entrez comme action :

```
item.Left = item.Left + 5
```

Avec les méthodes suivantes, vous pouvez activer ou désactiver l'action de la propriété "Position X".

Avec la méthode "ActivateDynamic", le déclenchement est activé dans Runtime :

```
Dim obj
Set obj = ScreenItems.Item("Rectangle1")
obj.ActivateDynamic "Left", "CycleTime1s"
```

Avec la méthode "DeactivateDynamic", le déclenchement est désactivé dans Runtime :

```
Dim obj
Set obj = ScreenItems.Item("Rectangle1")
obj.DeactivateDynamic "Left"
```

Remarque

Les variables WinCC restent sollicités, même lorsque le déclenchement est désactivé.

Voir aussi

Méthode ActivateDynamic (Page 683)

1.10.7.3 Pour ajouter un Trigger du type "Temporisateur"

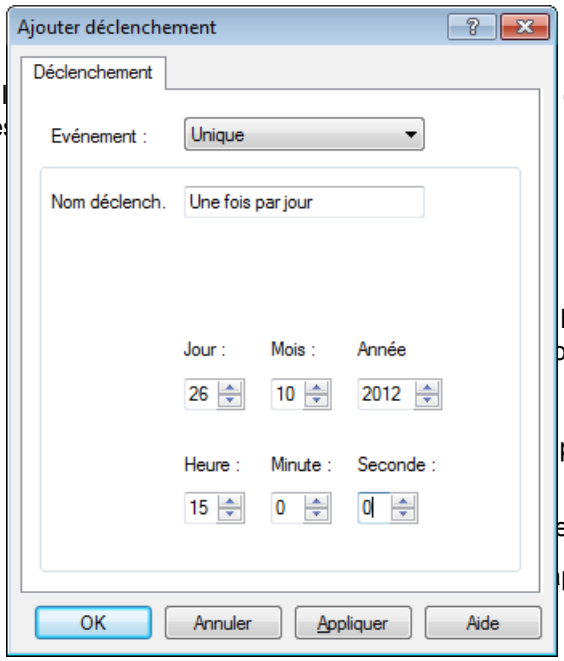
Introduction

A la date/heure déterminée, les Triggers de type "Temporisateur" exécutent une action. Les Triggers de type "Temporisateur" peuvent être des Triggers cycliques ou acycliques.

- Triggers acycliques Ils déclenchent une action une seule fois lorsque la date/heure configurée arrive.
- Triggers cycliques Ils déclenchent une action à intervalles réguliers. Vous devez configurer l'intervalle de temps et le moment de déclenchement du Trigger. Si vous sélectionnez comme Trigger cyclique un Trigger standard, le moment du départ est toujours le démarrage du Runtime. Vous pouvez également utiliser les cycles spécifiques à l'utilisateur en tant que cycles standard.

Remarque

Les Triggers cycliques garantissent une vitesse d'actualisation élevée dans votre système mais sollicitent assez fortement ce dernier. Sélectionnez les déclenchements cycliques uniquement pour les actions où le rafraîchissement est très important.
En cas de sollicitation extrême du système, il se peut que certaines actions ne puissent plus être exécutées.



Util dynamiser des propriétés dans Graphics De

Marche à suivre

1.
2. barre d'outils ou sélectionnez la commande ogue "Propriétés " apparaît.
3.
4. puis le type de déclenchement que vous
5. er Trigger" apparaît.
6. pez un nom de déclenchement ayant un doit être exécutée.
7. Si vous avez sélectionné le type de déclenchement "cyclique": tapez un nom de déclenchement ayant un sens et réglez la date/l'heure de départ à laquelle l'action doit être exécutée la première fois. Tapez un cycle de répétition de l'action.
   Validez votre entrée avec "OK".

Remarque

Vous pouvez attribuer plusieurs Trigger à une action. L'action sera toujours exécutée lorsque l'un des événements de déclenchement se produit.

Voir aussi

Pour supprimer un déclenchement (Page 79)

Pour ajouter un Trigger du type "Variable" (Page 77)

Déclenchement (Page 71)

Création et édition d'actions (Page 57)

Actions (Page 19)

1.10.7.4 Pour ajouter un Trigger du type "Variable"

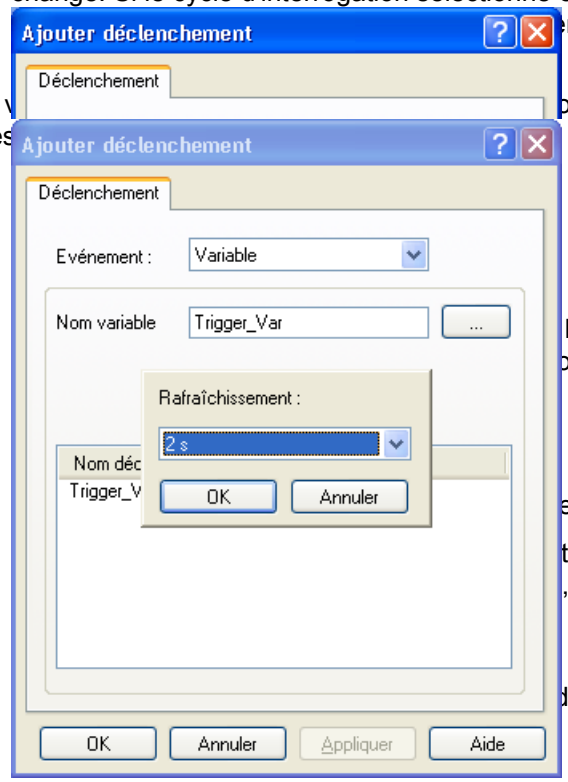
Introduction

Les Trigger de type "Variable" exécutent une action lorsqu'une variable change de valeur. Vous pouvez utiliser comme variable de déclenchement toute variable interne ou externe créée dans WinCC.

Vous pouvez faire exécuter les actions avec déclenchement de variable aux occasions suivantes:

- En cas de modification de la variable: l'action est exécutée dès que la valeur de la variable change. Etant donné que ce réglage sollicite extrêmement le système, il est recommandé d'utiliser aussi rarement que possible cette fréquence de mise à jour.
- Interrogation de l'état des variables selon les cycles standard (cycles utilisateur compris): Vous définissez un cycle dans les intervalles duquel la valeur des variables est interrogée. L'action n'est exécutée que si, au moment de l'interrogation, la valeur de la variable a changé. Si le cycle d'interrogation sélectionné est trop long, il se peut que la valeur de la ende compte. Dans ce cas l'action n'est pas

Si v on est exécutée lorsque la valeur de l'une des



Marche à suivre

1.
2. barre d'outils ou sélectionnez la commande ogue "Propriétés " apparaît.
3.
4.
5. er Trigger" apparaît.
6. tiliser comme Trigger ou appuyez sur le , pour sélectionner une variable du dialogue
7. d" pour ouvrir le dialogue de sélection pour

Sélectionnez un cycle et validez en tapant OK.

Voir aussi

Pour supprimer un déclenchement (Page 79)

Pour ajouter un Trigger du type "Variable" (Page 77)

Pour ajouter un Trigger du type "Temporisateur" (Page 76)

Déclenchement (Page 71)

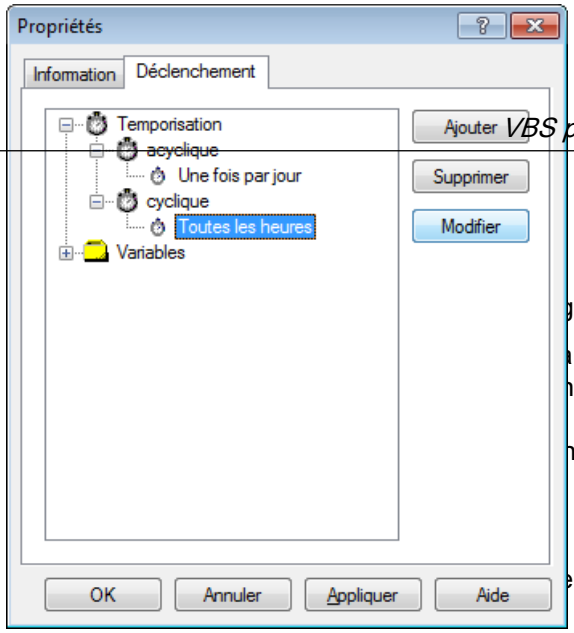
Création et édition d'actions (Page 57)

Actions (Page 19)

1.10.7.5 Pour modifier un déclenchement

Introduction

Vous pouvez à tout moment modifier des Trigger déjà définis, même pendant Runtime.



Marche à suivre

1. gger.
2. barre d'outils
   s le menu contextuel. Le dialogue
   s ouvrir l'action, en cliquant deux fois le
3.
4. r et appuyez sur le bouton "Modifier".
5. Modifiez le Trigger et validez votre saisie avec OK.

Voir aussi

Déclenchement (Page 71)

Pour supprimer un déclenchement (Page 79)

Pour ajouter un Trigger du type "Variable" (Page 77)

Pour ajouter un Trigger du type "Temporisateur" (Page 76)

Création et édition d'actions (Page 57)

Actions (Page 19)

1.10.7.6 Pour supprimer un déclenchement

Introduction

Les triggers paramétrés peuvent être supprimés à tout moment. Vous pouvez également supprimer des Trigger pendant Runtime.

Lorsqu'un trigger est supprimé au runtime, la suppression n'est effective qu'après l'enregistrement de l'action.

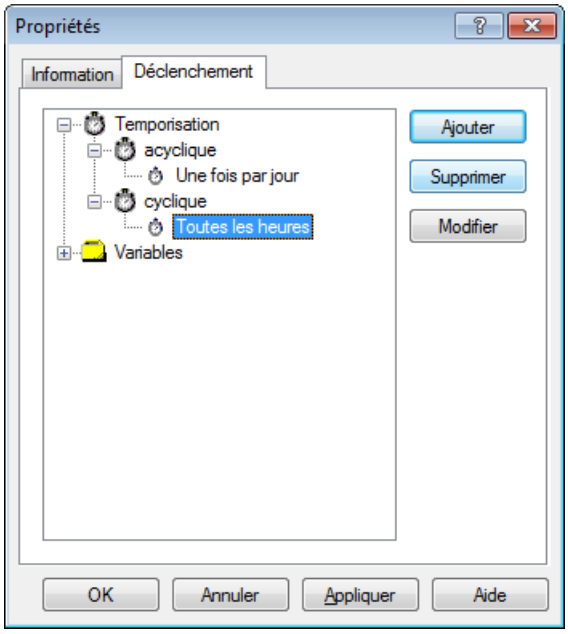
Remarque

Une action sans Trigger ne peut être exécutée dans Runtime. Toutes les actions qui ont utilisé le Trigger supprimé ne sont plus exécutées.

Marche à suivre

1. Ouvrez l'éditeur Global Script ou l'éditeur d'actions dans Graphics Designer.
2. Ouvrez l'action.
3. Appuyez sur le bouton "Info/Trigger" dans la barre d'outils ou sélectionnez la commande "Info/Trigger" dans le menu contextuel. Le dialogue "Propriétés " apparaît.
4. Sélectionnez l'onglet "Trigger"

5. Sélectionnez le Trigger que vous voulez supprimer et appuyez sur le bouton "Supprimer".



6. Le Trigger est immédiatement supprimé.

Remarque

Vous pouvez également directement supprimer des Triggers dans la fenêtre de navigation de Global Script via la commande "Supprimer" du menu contextuel.

Voir aussi

Actions (Page 19)

Pour modifier un déclenchement (Page 78)

Pour ajouter un Trigger du type "Variable" (Page 77)

Pour ajouter un Trigger du type "Temporisateur" (Page 76)

Déclenchement (Page 71)

Création et édition d'actions (Page 57)

1.10.8 Pour renommer une action

Introduction

Vous pouvez renommer une action dans Global Script. Lorsque vous renommez une action, ce sont le nom de l'action et le nom de fichier qui sont modifiés.

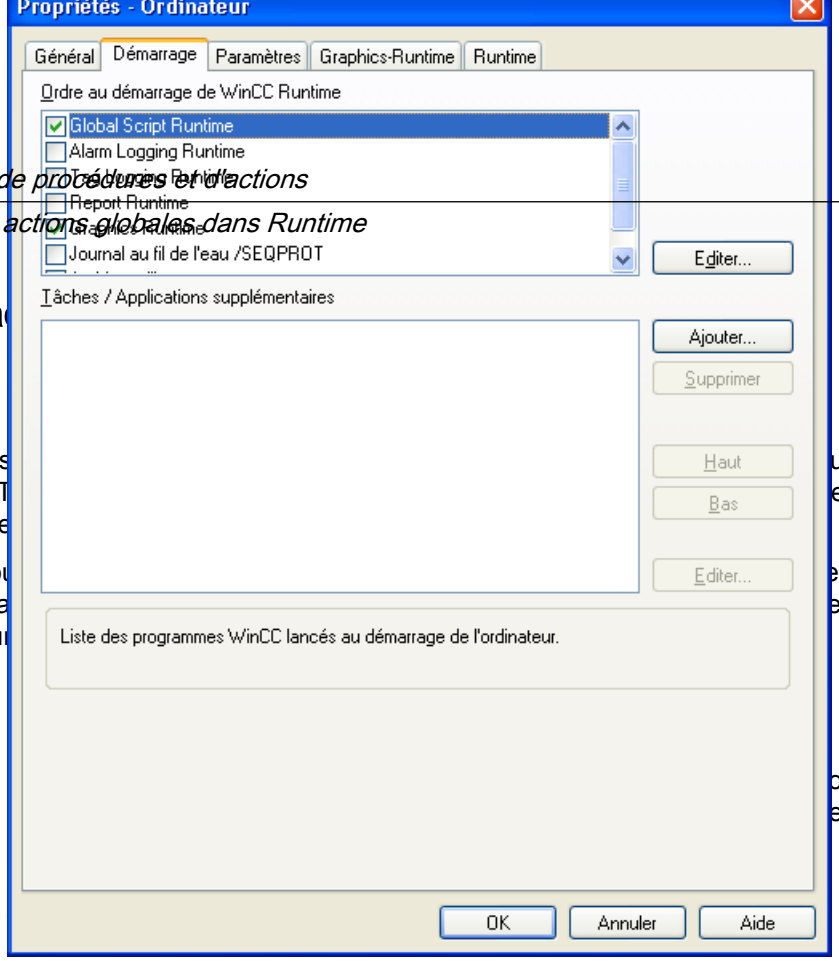
L'action que vous voulez renommer ne doit pas être ouverte dans la fenêtre d'édition.

Marche à suivre

1. Ouvrez Global Script.
2. Sélectionnez l'action que vous voulez renommer dans la fenêtre de navigation de l'éditeur.
3. Choisissez la commande "Renommer" dans le menu contextuel.
4. Tapez un nouveau nom pour l'action se terminant par *.bac.

Voir aussi

Pour protéger une action par un mot de passe (Page 68)
Pour enregistrer une action (Page 69)
Pour ajouter des informations supplémentaires à l'action (Page 66)
Pour éditer une action (Page 63)
Pour créer une nouvelle action (Page 62)
Déclenchement (Page 71)
Création et édition d'actions (Page 57)
Actions (Page 19)



1.11 Pour a

Introduction

Les ue fois que survient
le T exécutés lorsque la
vue

Pou e, soient exécutées,
il fa e de l'ordinateur
Ru

Marche à suivre

1. ordinateur, la
e.
2.
3.
4. Validez votre entrée avec "OK".

Voir aussi

Création et édition d'actions (Page 57)

Création et édition de procédures (Page 40)

Les éditeurs VBScript (Page 28)

Utilisation de Visual Basic Script dans WinCC (Page 12)

1.12 Diagnostic

1.12.1 Diagnostic

Introduction

Si vous exécutez et testez vos scripts en Runtime, il vous est possible d'afficher rapidement une analyse à l'aide de la fenêtre de diagnostic.

Outils de diagnostic

WinCC propose des outils pour l'analyse du comportement des actions au runtime.

- Les fenêtres d'application GSC-Runtime et Diagnostic GSC
- Utilisez un débogueur

GSC-Runtime et GCS-Diagnostic

Pour utiliser les fenêtres d'application GSC-Runtime et Diagnostic GSC, vous devez les placer dans une vue de process. Il peut alors s'agir d'une vue de process spécialement conçue pour le diagnostic et qui sera appelée dans Runtime.

Ces fenêtres d'application permettent des stratégies différentes:

GSC-Runtime renseigne sur le comportement dynamique de toutes les actions (Global Script), permet de lancer individuellement et de déclarer le début et la fin de chaque action; il permet aussi de basculer dans l'éditeur Global Script le Runtime étant actif.

La fonction Diagnostic GSC affiche, dans l'ordre chronologique de leur appel, les méthodes Trace contenues dans les actions. Il affiche également les instructions Trace codées dans les procédures appelées dans des actions. L'utilisation ciblée d'instructions Trace, p. ex. pour afficher des valeurs de variable, permet de suivre l'exécution des actions et des procédures appelées dans celles-ci. Spécifiez les instructions Trace sous la forme "HMIRuntime.Trace(<Sortie>)".

Les affichages Trace de C et VBS sont édités dans la fenêtre de diagnostic GSC.

Remarque

Les erreurs de temps d'exécution ne sont pas affichées

Certaines erreurs de script ne sont pas éditées via Trace ni affichées dans la boîte de dialogue d'erreur. Utilisez Microsoft Script Debugger.

Debugger

Pour tester vos scripts en Runtime, vous pouvez utiliser un débogueur outre les fenêtres de diagnostic. L'utilisation du Microsoft Script Debugger est décrite au chapitre "Tester avec Debugger".

Vous trouverez le Microsoft Script Debugger dans le Download Center de Microsoft, sous l'adresse URL suivante :

- http://www.microsoft.com/downloads/Search.aspx?displaylang=en

Cherchez "Script Debugger" dans le champ "Search" et sélectionnez le téléchargement dont vous avez besoin.

Voir aussi

Tester avec Debugger (Page 90)

GSC Runtime (Page 87)

Diagnostic GSC (Page 84)

Microsoft Download Center (http://www.microsoft.com/downloads/Search.aspx?displaylang=en)

1.12.2 Diagnostic GSC

1.12.2.1 Diagnostic GSC

Introduction

Diagnostic GSC affiche dans la fenêtre de diagnostic, dans l'ordre chronologique de leur appel, les instructions Trace contenues dans les actions. Il affiche également les instructions Trace codées dans les procédures appelées dans des actions. L'utilisation ciblée d'instructions Trace, p. ex. pour afficher des valeurs de variable, permet de suivre l'exécution des actions et des procédures appelées dans celles-ci.

Application

Pour utiliser Diagnostic GSC vous devez placer une fenêtre d'application du type Diagnostic GSC dans une vue de process. Les attributs de Diagnostic GSC permettent de paramétrer l'apparence de la fenêtre de Diagnostic GSC.

La fenêtre de Diagnostic GSC est supprimée aux changements de vues.

Remarque

Des messages sont également émis dans la fenêtre "Diagnostic GSC" lorsque le débogueur est activé.

Voir aussi

La barre d'outils de Diagnostic GSC (Page 86)

Les attributs de Diagnostic GSC (Page 85)

Pour insérer dans une vue la fenêtre de diagnostic GSC (Page 85)

1.12.2.2 Pour insérer dans une vue la fenêtre de diagnostic GSC

Introduction

Pour utiliser Diagnostic GSC, vous devez placer Diagnostic GSC dans une vue de process. Cette vue de process peut être une vue existante ou une vue servant uniquement au diagnostic. Diagnostic GSC ne peut pas être inséré directement comme application dans la vue de process, mais doit être placé comme application dans une fenêtre d'application. La fenêtre d'application elle-même est alors un élément de la vue de process.

Condition

Graphics Designer est lancé et la vue de process est ouverte.

Marche à suivre

1. Insérez dans votre vue la "Fenêtre d'application", depuis l'arborescence des objets "Objets complexes".
2. Sélectionnez dans le dialogue "Contenu de la fenêtre" l'inscription "Global Script" et validez avec "OK".
3. Sélectionnez dans le dialogue "Modèle" l'inscription "Diagnostic GSC".
4. Validez avec OK pour insérer la fenêtre de diagnostic.

Voir aussi

La barre d'outils de Diagnostic GSC (Page 86)

Les attributs de Diagnostic GSC (Page 85)

Diagnostic GSC (Page 84)

1.12.2.3 Les attributs de Diagnostic GSC

Liste récapitulative

Diagnostic GSC possède des attributs qui permettent de configurer l'apparence de la fenêtre de diagnostic GSC au runtime. Ce sont les attributs de géométrie et en particulier les suivants:

- Affichage: Cet attribut sert à paramétrer si la fenêtre est affichée ou masquée. Cet attribut est dynamisable par le nom "Visible"
- Taille modifiable: indique si la fenêtre peut être redimensionnée au runtime
- Déplaçable: Indique si la fenêtre doit pouvoir être déplacée au runtime
- Bordure: Indique si la fenêtre doit être entourée d'une bordure. Si la fenêtre a une bordure, sa hauteur et sa largeur sont redimensionnables au runtime
- Titre: Indique si la fenêtre doit posséder un titre
- Maximisable: Indique si la barre de titre contient le bouton Maximiser de la fenêtre

- Peut être fermé: indique si la barre de titre contient le bouton servant à fermer la fenêtre
- Premier plan: indique si la fenêtre doit toujours se trouver au premier plan

Voir aussi

La barre d'outils de Diagnostic GSC (Page 86)

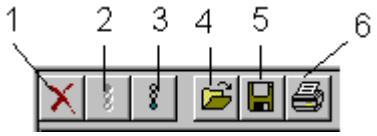
Pour insérer dans une vue la fenêtre de diagnostic GSC (Page 85)

Diagnostic GSC (Page 84)

1.12.2.4 La barre d'outils de Diagnostic GSC

Liste récapitulative

La barre d'outils de Diagnostic GSC permet de commander l'affichage dans la fenêtre de diagnostic ainsi que l'enregistrement, l'impression et l'ouverture du contenu de la fenêtre.



1: supprime le contenu de la fenêtre de diagnostic

2: stoppe la mise à jour de la fenêtre

3: active la mise à jour de la fenêtre

4: ouvre un fichier texte dans la fenêtre

5: enregistre le contenu de la fenêtre dans un fichier texte

6: imprime le contenu de la fenêtre

Voir aussi

Les attributs de Diagnostic GSC (Page 85)

Pour insérer dans une vue la fenêtre de diagnostic GSC (Page 85)

Diagnostic GSC (Page 84)

1.12.3 GSC Runtime

1.12.3.1 GSC Runtime

Introduction

GSC-Runtime est une fenêtre qui affiche le comportement dynamique de toutes les actions Global Script au runtime. En outre GSC-Runtime permet de piloter l'exécution de chaque action et de basculer dans l'éditeur Global Script au runtime.

Actions

Dans la fenêtre GSC-Runtime, on distingue les actions C et VBS:

- Configure une action en C
- Configure une action en VBS

Les informations suivantes sont affichées:

- Nom de l'action: Le nom de l'action
- ID: Les ID d'action. Elle est utilisée de façon interne au système. GSC-Runtime fournit le nom de l'action correspondant à l'ID de l'action. L'association de l'ID au nom de l'action n'est valide que jusqu'à l'arrêt du runtime ou, lorsque le runtime est actif, jusqu'à l'enregistrement d'une action.
- Etat: Renseigne sur l'état courant de l'action. Les états possibles sont indiqués dans le tableau ci-dessous
- Intervalle d'activation: C'est le temps au format Heure:Minute:Seconde qui s'est écoulé entre deux appels de l'action
- Retour: la valeur retournée par l'action
- Démarré le: Date et heure du déclenchement de l'action
- Prochain dém: Date et heure du prochain déclenchement de l'action
- Message d'erreur: contient le texte d'erreur en cas d'erreur

Etat des actions

Les différents états possibles d'une action:

- Le début d'action a été déclaré
- La fin d'action a été déclarée
- L'action a été arrêtée
- Action en cours
- Erreur lors de la déclaration de début d'action !
- Erreur lors de l'exécution de l'action !

Menu contextuel

Le menu contextuel de chaque action contient les fonctions suivantes:

- Déclarer fin d'action: L'action concernée ne sera plus exécutée à l'issue de l'exécution en cours
- Déclarer début d'action: L'action concernée sera à nouveau exécutée au prochain trigger
- Démarrer: L'action concernée est exécutée une fois
- Editer: L'action concernée est ouverte dans l'éditeur Global Script pour édition. Le runtime reste actif. Lorsque l'action exécutée est compilée (si nécessaire) et enregistrée, les modifications sont prises en compte immédiatement par le système runtime
  Vous pouvez, pour chaque action, gérer la possibilité d'ouvrir le menu contextuel en attribuant une autorisation.
  Pour utiliser GSC-Runtime vous devez insérer une fenêtre d'application du type GSC-Runtime dans une vue de process. Les attributs de GSC-Runtime permettent de configurer l'apparence de la fenêtre de GSC-Runtime.

Remarque

Le rafraîchissement de la fenêtre GSC-Runtime augmente la charge de calcul du système. La charge du système dépend du nombre d'actions affichées dans la fenêtre. Vous pouvez réduire la charge du système en diminuant la hauteur de la fenêtre pour qu'elle affiche moins de lignes.

Voir aussi

Pour insérer dans une image la fenêtre GSC Runtime (Page 88)

Les attributs de GSC Runtime (Page 89)

1.12.3.2 Pour insérer dans une image la fenêtre GSC Runtime

Introduction

Pour utiliser GSC-Runtime, vous devez placer GSC-Runtime dans une image processus. Cette vue de process peut être une vue existante ou une vue servant uniquement au diagnostic. Il n'est pas possible d'insérer GSC-Runtime directement dans la vue de process; il doit être placé comme application dans une fenêtre d'application. La fenêtre d'application elle-même est alors un élément de la vue de process.

Condition

Graphics Designer est lancé et la vue de process est ouverte.

Marche à suivre

1. Insérez dans votre vue la "Fenêtre d'application", depuis l'arborescence des objets "Objets complexes".
2. Sélectionnez dans le dialogue "Contenu de la fenêtre" l'inscription "Global Script" et validez avec "OK".
3. Sélectionnez dans le dialogue "Modèle" la commande "GSC-Runtime"
4. Validez avec OK pour insérer la fenêtre de diagnostic.

Voir aussi

GSC Runtime (Page 87)

Les attributs de GSC Runtime (Page 89)

1.12.3.3 Les attributs de GSC Runtime

Liste récapitulative

GSC-Runtime possède des attributs qui permettent de configurer l'apparence de la fenêtre GSC-Runtime au runtime. Ce sont les attributs de géométrie et en particulier les suivants:

- Affichage: Cet attribut sert à paramétrer si la fenêtre est affichée ou masquée. Cet attribut est dynamisable par le nom "Visible"
- Taille modifiable: indique si la fenêtre peut être redimensionnée au runtime
- Déplaçable: Indique si la fenêtre doit pouvoir être déplacée au runtime
- Bordure: Indique si la fenêtre doit être entourée d'une bordure. Si la fenêtre a une bordure, sa hauteur et sa largeur sont redimensionnables au runtime
- Titre: Indique si la fenêtre doit posséder un titre
- Maximisable: Indique si la barre de titre contient le bouton Maximiser de la fenêtre
- Peut être fermé: indique si la barre de titre contient le bouton servant à fermer la fenêtre
- Premier plan: indique si la fenêtre doit toujours se trouver au premier plan

Voir aussi

GSC Runtime (Page 87)

Pour insérer dans une image la fenêtre GSC Runtime (Page 88)

1.12.4 Tester avec Debugger

1.12.4.1 Tester avec Debugger

Liste récapitulative

Vous pouvez utiliser un Debugger pour tester vos VB-Scripts dans Runtime, par exemple :

- Microsoft Script Debugger
- Debugger "InterDev" (livré avec Developer Studio)
- Microsoft Script Editor (MSE) Debugger (livré avec Microsoft Office)

On décrira par la suite uniquement comment utiliser le Microsoft Script Debugger.

Téléchargement du Microsoft Script Debugger

Vous trouverez Microsoft Script Debugger dans le Download Center de Microsoft, sous l'adresse URL suivante :

- http://www.microsoft.com/downloads/Search.aspx?displaylang=en833a6a92-961e-4ce1-9069-528d22605127

Cherchez dans le champ "Search" "Script Debugger" et sélectionnez le téléchargement dont vous avez besoin.

Information concernant MSE Debugger

Lorsque vous utilisez MSE Debugger, il vous faut modifier le réglage suivant pour pouvoir afficher les processus en cours :

1. Dans la fenêtre "Processus", cliquez sur le bouton "Propriétés".
2. Activez dans le dialogue "Propriétés débogueur" l'option "Déboguer Just-In-Time".
3. Redémarrez l'ordinateur.
4. Désactivez dans MS Internet Explorer l'option "Disable script debugging", pour que Internet Explorer n'empêche pas le déroulement de la procédure de débogage de WinCC.

Voir aussi

Principes de base du débogage (Page 93)
Pour exécuter des instructions Script (Page 107)
Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)
Pour activer des signets dans le Script (Page 105)
Pour supprimer des points d'arrêt (Page 105)
Pour définir des points d'arrêt (Page 103)
Pour traiter les Scripts étape par étape (Page 102)
Pour sélectionner un Script en vue de l'éditer (Page 101)
Nom de l'action et de la procédure dans Debugger (Page 99)
Structure des fichiers VBScript (Page 97)
Composants du Microsoft Script Debugger (Page 95)
Pour activer le débogueur (Page 91)
Diagnostic (Page 83)
Microsoft Download Center (http://www.microsoft.com/downloads/Search.aspx?displaylang=en)

1.12.4.2 Pour activer le débogueur

Principe

Plusieurs possibilités s'offrent à vous pour activer le débogueur :

- Appel automatique du débogueur en cas d'erreur en Runtime.
- Affichage d'une boîte d'erreurs en Runtime via laquelle le débogueur peut être appelé.
- Démarrage du débogueur à partir du menu de départ et ouvrir l'un des Scripts Runtime.

Condition

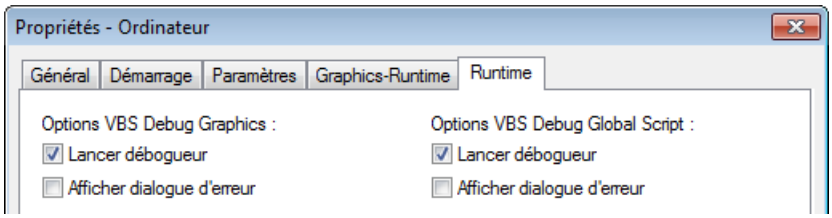
Microsoft Script Debugger doit être installé sur l'ordinateur de configuration.

Marche à suivre

La procédure suivante décrit les deux premiers points, l'activation du programme de test dans WinCC.

1. Dans l'explorateur WinCC, sélectionnez dans le menu contextuel de l'ordinateur, la commande "Propriétés". Le dialogue "Propriétés - Ordinateur" s'ouvre.
2. Sélectionnez l'onglet "Runtime".

3. Activez les options Debug que vous voulez. Vous pouvez régler indépendamment l'un de l'autre le comportement Debug pour les actions dans Global Script et celui dans Graphics Designer :



4. Sélectionnez "Lancer débogueur" si le débogueur doit être démarré directement dès qu'une erreur survient en Runtime.
5. Sélectionnez "Afficher dialogue d'erreur" si le débogueur ne doit pas être démarré directement, mais qu'un boîtier d'erreur doit apparaître pour donner des informations sur l'erreur. Vous pouvez démarrer le débogueur via un bouton, à partir du boîtier d'erreurs.
6. Validez votre entrée avec "OK".

Démarrer Debugger et ouvrir un Script en cours

Vous pouvez également démarrer ultérieurement le débogueur et le lier au système fonctionnant actuellement. Etablissez à cet effet dans le débogueur une liaison avec les processeurs respectifs, "pdlrt.exe" pour le système Runtime graphique et "gscrt.exe" pour le système Runtime global. Vous trouverez dans la rubrique "Sélectionner Script" comment ouvrir dans le débogueur un Script en cours.

Quitter Debugger

Vous pouvez quitter le débogueur sans quitter le mode Runtime de WinCC.

Voir aussi

Pour activer des signets dans le Script (Page 105)
Pour exécuter des instructions Script (Page 107)
Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)
Pour supprimer des points d'arrêt (Page 105)
Pour définir des points d'arrêt (Page 103)
Pour traiter les Scripts étape par étape (Page 102)
Pour sélectionner un Script en vue de l'éditer (Page 101)
Nom de l'action et de la procédure dans Debugger (Page 99)
Structure des fichiers VBScript (Page 97)
Composants du Microsoft Script Debugger (Page 95)
Principes de base du débogage (Page 93)
Tester avec Debugger (Page 90)
Diagnostic (Page 83)

1.12.4.3 Principes de base du débogage

Introduction

Pour déboguer vos VBScripts, vous pouvez utiliser Microsoft Script Debugger. Microsoft Script Debugger vous permet:

- de consulter le code source du Script que vous déboguez
- de contrôler le traitement par étapes des Scripts
- de consulter et de modifier des variables et des valeurs caractéristiques
- de suivre et de contrôler le déroulement des Scripts

Remarque

Notez que, dans le débogueur, votre code est protégé contre l'écriture. Vous ne pouvez pas modifier directement le code dans le debogueur, mais seulement tester les modifications nécessaires.

Types d'erreur

On distingue les types d'erreur suivants lors du débogage:

Erreur de syntaxe

Les erreurs de syntaxe surviennent lorsque par exemple vous écrivez mal un mot-clé ou ne fermez pas une parenthèse. Lorsque vous utilisez le contrôle de syntaxe de WinCC, vous pouvez déjà exclure les erreurs de syntaxe en Runtime, avant le test des Scripts. Dans

Graphics Designer, seuls les Scripts de syntaxe correcte seront en principe enregistrés. Le contrôle de syntaxe dans WinCC contrôle en outre:

- si les noms de procédure dans Global Script sont univoques
- si les modules d'action dans Global Script ne contiennent qu'une seule procédure
- si la partie Action dans Graphics Designer ne contient qu'une seule procédure

Le contrôle de syntaxe effectué dans WinCC permet d'analyser le Script sans l'exécuter. Le Script est encore analysé juste avant l'exécution en Runtime. Tous les Scripts sont analysés, même ceux qui sont exécutés ultérieurement, par exemple seulement après qu'une action déterminée se soit produite.

Si votre Script contient des erreurs de syntaxe, le Script ne sera pas exécuté en Runtime.

Erreur d'exécution

Une erreur d'exécution survient lorsqu'une action non valable/défectueuse doit être exécutée, par exemple parce qu'une variable n'est pas définie. Pour éviter les erreurs d'exécution, vous pouvez utiliser dans VBScript l'instruction "On Error Resume Next". Cette instruction fait en sorte qu'une instruction consécutive est exécutée à la suite d'une erreur d'exécution. Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. Utilisez l'instruction "On Error Goto 0" pour désactiver à nouveau le traitement des erreurs d'exécution dans le Script.

Erreurs logiques

Le débogueur est particulièrement utile pour éliminer les erreurs logiques. Une erreur logique se produit lorsque le résultat escompté n'arrive pas, parce que par exemple une condition est mal vérifiée. Pour éviter les erreurs logiques, passez votre Script en revue étape par étape pour identifier la partie qui ne fonctionne pas.

Procédure de base

Lorsqu'une erreur s'est produite et que le débogueur est ouvert, le Script protégé contre l'écriture apparaît dans une fenêtre. Vous pouvez naviguer dans le document Script, définir des points d'arrêt, exécuter à nouveau le Script en Runtime et traiter les Scripts étape par étape.

Vous trouverez au paragraphe "Traiter un Script étape par étape" les étapes essentielles du débogage de vos Scripts.

Vous ne pouvez pas éditer directement dans le débogueur le code source de vos Scripts. Si vous avez trouvé une erreur, vous pouvez la corriger dans le Script d'origine dans WinCC.

Remarque

Vous trouverez dans l'aide en ligne du Microsoft Script Debugger des conseils et astuces concernant le débogage, mais également les sources d'erreur les plus fréquentes ainsi que d'autres informations.

Changement de vue lors du débogage

Lorsque vous effectuez un changement d'image pendant le débogage, le document Script de l'"ancienne" image reste ouvert mais n'est plus valable. Il se peut que des erreurs périmées

s'affichent étant donné que les objets appelés après le changement d'image ne sont plus disponibles.

Voir aussi

Tester avec Debugger (Page 90)

Pour exécuter des instructions Script (Page 107)

Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)

Pour activer des signets dans le Script (Page 105)

Pour supprimer des points d'arrêt (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour traiter les Scripts étape par étape (Page 102)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Nom de l'action et de la procédure dans Debugger (Page 99)

Structure des fichiers VBScript (Page 97)

Composants du Microsoft Script Debugger (Page 95)

Pour activer le débogueur (Page 91)

Diagnostic (Page 83)

1.12.4.4 Composants du Microsoft Script Debugger

Introduction

Microsoft Script Debugger vous présente plusieurs composants qui vous sont utiles pour le débogage :

Fenêtre "Command Window"

Appelez "Command Window" via la commande de menu "View" > "Command Window".

Lorsqu'un Script fonctionne en Runtime, vous pouvez utiliser la fonction "Command Window" du débogueur pour par exemple déterminer et modifier les valeurs des variables et des propriétés du Script en cours de fonctionnement. Les modifications que vous exécutez dans "Command Window" font immédiatement effet sur le Script en cours, ce qui vous permet de tester directement les modifications voulues.

"Command Window" vous permet d'effectuer les actions suivantes :

- Saisir des instructions : Vous pouvez saisir et exécuter des instructions directement dans VBScript
- Modifier les valeurs variable : Dans "Command Window", vous pouvez déterminer et modifier directement des valeurs variable. Cela concerne aussi bien les variables dans le Script actuel que les variables globales.
- Modifier les propriétés : Vous pouvez non seulement lire mais également écrire les propriétés de tous les objets du contexte du Script actuel.

Vous pouvez toujours utiliser "Command Window" lorsqu'un Script est parvenu à un point d'arrêt ou lorsque vous êtes passé d'un point d'arrêt à d'autres instructions.

Remarque

Notez que les modifications que vous effectuez dans "Command Window" n'ont aucune influence sur le code source de vos Scripts, mais qu'elles servent uniquement à des tests dans le débogueur.

Fenêtre "Running Documents"

Appelez la fenêtre "Running Documents" via la commande de menu "View" > "Running Documents".

Vous apercevez dans cette fenêtre tous les Scripts actuellement en cours dans WinCC Runtime, répartis en Scripts de Global Script ("Global Script Runtime") et Scripts du système graphique Runtime ("PDLRT"). Vous voyez respectivement toutes les actions en cours et tous les modules du Global Script Runtime. Dans le système graphique Runtime, les Scripts sont répartis en Actions déclenchées par Trigger (Nom de vue_trigger) et Actions déclenchées par des événements (Nom de vue_events).

Fenêtre "Call Stack"

Appelez la fenêtre "Call Stack" via la commande de menu "View" > "Call Stack".

Vous voyez dans cette fenêtre une liste de toutes les actions en cours et de toutes les procédures appelées. Lorsqu'une procédure est par exemple appelée, son nom est ajouté à la liste "Call Stack. Une fois la procédure terminée, le nom est supprimé de la liste. Vous pouvez sélectionner une procédure de la liste pour sauter à l'endroit correspondant dans le document Script, endroit d'où a été appelée la procédure.

Voir aussi

Pour exécuter des instructions Script (Page 107)

Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)

Pour activer des signets dans le Script (Page 105)

Pour supprimer des points d'arrêt (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour traiter les Scripts étape par étape (Page 102)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Nom de l'action et de la procédure dans Debugger (Page 99)

Structure des fichiers VBScript (Page 97)

Principes de base du débogage (Page 93)

Pour activer le débogueur (Page 91)

Tester avec Debugger (Page 90)

Diagnostic (Page 83)

1.12.4.5 Structure des fichiers VBScript

Principe

Les actions déclenchées par événement et les actions cycliques/déclenchées par une variable sont strictement séparées lors du traitement pour ne pas gêner le traitement simultané des Scripts cycliques et de ceux déclenchés par événement dans le système graphique Runtime. Une action cyclique ne peut donc pas gêner l'exécution d'une action qui doit être déclenchée lorsqu'un bouton est appuyé.

Pour garantir cela, les actions déclenchées par événement et les actions cycliques ainsi que celles déclenchées par une variable sont écrites dans des fichiers Script séparés lors de l'enregistrement d'une vue. Si vous avez défini une partie globale pour toutes les vues dans les actions du Graphics Designer, cette partie sera copiée dans les deux fichiers Script. On copiera de même dans les deux fichiers Script les modules qui sont utilisés dans les actions.

Si une variable doit être utilisée depuis un module, il vous faut appeler le module en question. Faute de quoi le module ne sera pas copié dans le fichier Script et une erreur sera générée.

Remarque

Etant donné que les deux fichiers Script sont traités séparément, il n'ont aucune zone de données commune. Il n'y aura donc aucune synchronisation des variables globales entre les deux fichiers Script. Si une synchronisation s'avère nécessaire, implémentez-la par l'objet DataSet et ou par des variables internes WinCC.

Structure des fichiers Script

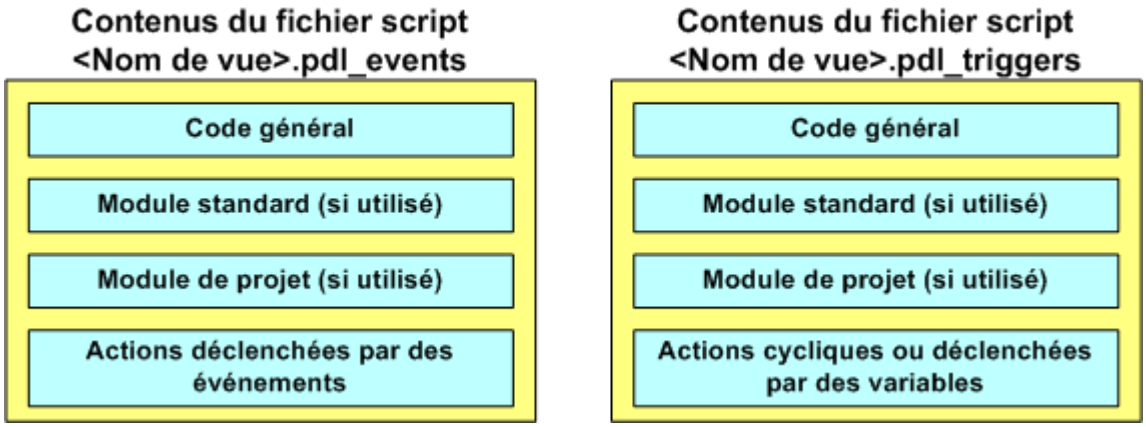
Si vous déboguez des Scripts avec le débogueur, les fichiers Script des différents systèmes Runtime s'ouvrent toujours.

Pour le système graphique Runtime, cela signifie que vous recevez deux fichiers Script par vue :

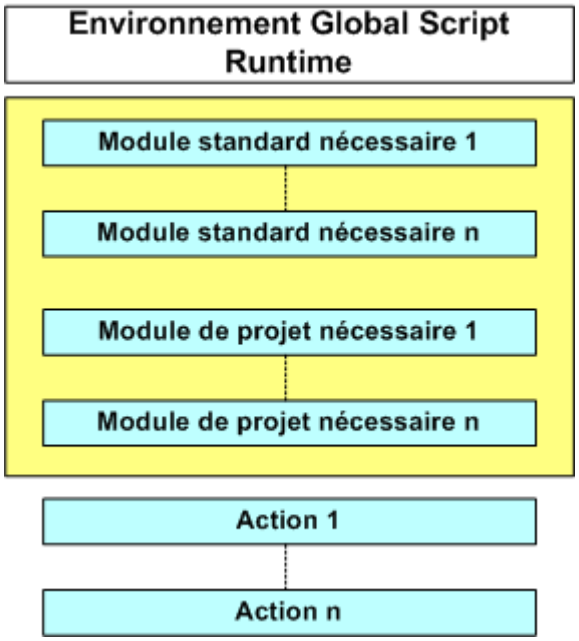
- <Nom de la vue>.pdl_events: Contient les actions déclenchées par événement
- <Nom de la vue>.pdl_triggers: Contient les actions cycliques et les actions déclenchées par une variable.

La structure des fichiers Script vous est présentée ci-après :

Système graphique Runtime



Global Script Runtime



Remarque

Notez que les actions et les procédures du système graphique Runtime ne s'affichent pas dans le fichier Script avec les noms d'action sous lesquels vous les avez enregistrées dans WinCC. Vous trouverez sous "Noms d'action et de procédure dans Debugger" les conventions de nom pour les actions et procédures dans les fichiers Script.

Voir aussi

Pour activer le débogueur (Page 91)
Pour exécuter des instructions Script (Page 107)
Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)
Pour activer des signets dans le Script (Page 105)
Pour supprimer des points d'arrêt (Page 105)
Pour définir des points d'arrêt (Page 103)
Pour traiter les Scripts étape par étape (Page 102)
Pour sélectionner un Script en vue de l'éditer (Page 101)
Nom de l'action et de la procédure dans Debugger (Page 99)
Composants du Microsoft Script Debugger (Page 95)
Principes de base du débogage (Page 93)
Tester avec Debugger (Page 90)
Diagnostic (Page 83)

1.12.4.6 Nom de l'action et de la procédure dans Debugger

Nom de l'action et de la procédure dans Debugger

Les noms de procédures et d'actions dans les fichiers Script dans Debugger se distinguent des noms sous lesquels vous avez enregistré vos Scripts dans WinCC.

Les noms d'actions et de procédures dans les fichiers Script se composent selon les règles suivantes :

Type d'action	Nom dans le fichier Script
Action cyclique ou déclenchée par une variable pour une propriété	NomObjet_NomPropriété_Trigger
Evénements au niveau de la souris	NomObjet_OnClick NomObjet_OnLButtonDown NomObjet_OnLButtonUp NomObjet_OnRButtonDown NomObjet_OnRButtonUp
Evénements au niveau du clavier	NomObjet_OnKeyDown NomObjet_OnKeyUp
Evénements au niveau de l'objet	NomObjet_OnObjectChanged NomObjet_OnSetFocus

Type d'action	Nom dans le fichier Script
Evénements au niveau de propriétés	NomObjet_PropertyName_OnPropertyChanged NomObjet_PropertyName_OnPropertyStateChanged
Evénements au niveau des vues	Document_OnOpen Document_OnClosed

Longueur admise pour les noms d'action

Les noms d'action dans les fichiers Script ne doivent pas dépasser 255 caractères. Chaque caractère spécial utilisé dans un nom d'objet est remplacé par cinq caractères. Après un X en première position, le caractère spécial est représenté par un code hexadécimal à quatre positions. Si vous configurez par exemple sur clic de souris une action pour un bouton portant le nom "AppuieMoi", le script apparaîtra dans le fichier Script sous la forme "DrXFFFCckMich_OnClick".

Si ce nom d'objet composé est trop long, un message d'erreur est émis lors du contrôle de syntaxe dans WinCC. Du fait de cette restriction, vous devez faire attention à la longueur des objets graphiques pendant la configuration.

Remarque

Si vous voulez déterminer le nom d'un objet en Runtime, appuyez <Ctrl+Alt+Maj> et amenez la souris sur l'objet en question. Le nom de la vue et le nom de l'objet s'affichent alors dans une info-bulle.

Voir aussi

Pour exécuter des instructions Script (Page 107)

Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)

Pour activer des signets dans le Script (Page 105)

Pour supprimer des points d'arrêt (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour traiter les Scripts étape par étape (Page 102)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Structure des fichiers VBScript (Page 97)

Composants du Microsoft Script Debugger (Page 95)

Principes de base du débogage (Page 93)

Pour activer le débogueur (Page 91)

Tester avec Debugger (Page 90)

Diagnostic (Page 83)

1.12.4.7 Pour sélectionner un Script en vue de l'éditer

Introduction

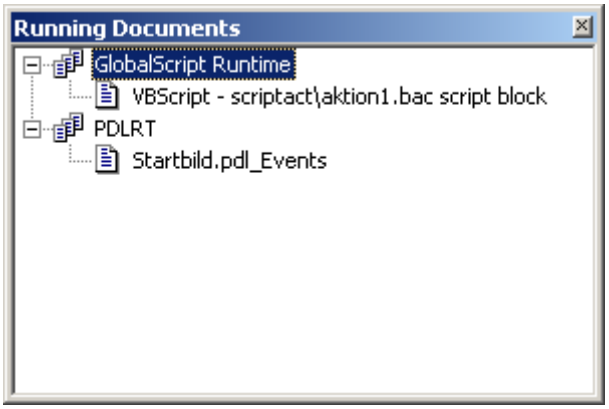
Si vous appelez Microsoft Script Debugger à partir de la fenêtre d'accueil de Windows pendant le Runtime, et non par activation automatique dans WinCC, vous pouvez sélectionner un Script en cours en vue de le traiter.

Condition préalable

Runtime est activé, la vue que vous voulez déboguer est active.

Marche à suivre

1. Démarrez le débogueur à partir de la fenêtre d'accueil de Windows ("Start" > "Programmes" > "Accessoires" > "Microsoft Script Debugger").
2. Dans la barre de menus, activez la commande "View" > "Running Documents".
   La fenêtre "Running Documents" s'ouvre. Vous apercevez dans cette fenêtre tous les Scripts actuellement en cours dans WinCC-Runtime, répartis en Scripts de Global Script ("Global Script Runtime") et Scripts du système graphique Runtime ("PDLRT").



   Dans l'exemple ci-dessus, la "Vue d'accueil.pdl" est active et il n'existe dans la vue d'accueil que des Scripts déclenchés par événement.
3. Double-cliquez dans la fenêtre "Running Documents" le document Script que vous voulez déboguer.
   Le document Script s'ouvre protégé contre l'écriture dans la fenêtre du débogueur.

Voir aussi

Pour exécuter des instructions Script (Page 107)
Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)
Pour activer des signets dans le Script (Page 105)
Pour supprimer des points d'arrêt (Page 105)
Pour définir des points d'arrêt (Page 103)
Pour traiter les Scripts étape par étape (Page 102)
Nom de l'action et de la procédure dans Debugger (Page 99)
Structure des fichiers VBScript (Page 97)
Composants du Microsoft Script Debugger (Page 95)
Principes de base du débogage (Page 93)
Pour activer le débogueur (Page 91)
Tester avec Debugger (Page 90)
Diagnostic (Page 83)

1.12.4.8 Pour traiter les Scripts étape par étape

Introduction

Vous pouvez utiliser Microsoft Script Debugger, pour traiter vos Scripts étape par étape et localiser ainsi systématiquement les erreurs logiques par exemple. Vous pouvez ainsi tester en Runtime les effets de chacune des lignes du Script.

Principe de la procédure

1. Activez le document à déboguer en Runtime.
2. Démarrez le débogueur manuellement à partir du menu d'accueil, puis ouvrez le fichier Script voulu ou activez le débogueur dans WinCC. Si le débogueur est activé dans WinCC, il s'ouvre automatiquement lorsqu'un Script incorrect est exécuté.
3. Placez un point d'arrêt dans le fichier Script. Il est typique de définir les points d'arrêt devant la ligne de code où vous soupçonnez des erreurs.
4. Permutez sur WinCC Runtime et déclenchez une action provoquant la mise en marche du Script.
   Le débogueur s'arrête au premier point d'arrêt et marque la ligne actuelle.

5. Pour parcourir étape par étape le document Script, sélectionnez l'une des commandes de menu suivante:
   "Debug" > "Step Into" : Passe à la ligne de code suivante. Si le Script de cette ligne affiche une procédure, sautez à cette dernière avec la commande "Step Into". Vous pouvez alors traiter étape par étape la procédure que vous avez appelée.

   "Debug" > "Step Over" : Saute la procédure appelée. La procédure est exécutée mais le débogueur ne vous fait pas parcourir chaque ligne. Au lieu de cela , une fois la procédure exécutée, il vous mène à la ligne suivante du Script actuel.
6. Pour interrompre le traitement étape par étape d'une procédure, sélectionnez la commande de menu "Debug" > "Step Out". Le débogueur saute alors à l'action suivante.
7. Procédez étape par étape jusqu'à la fin du document, ou sélectionnez la commande de menu "Debug" > "Run", pour redémarrer le Script en Runtime.

Voir aussi

Principes de base du débogage (Page 93)

Pour exécuter des instructions Script (Page 107)

Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)

Pour activer des signets dans le Script (Page 105)

Pour supprimer des points d'arrêt (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Nom de l'action et de la procédure dans Debugger (Page 99)

Structure des fichiers VBScript (Page 97)

Composants du Microsoft Script Debugger (Page 95)

Pour activer le débogueur (Page 91)

Tester avec Debugger (Page 90)

Diagnostic (Page 83)

1.12.4.9 Pour définir des points d'arrêt

Introduction

Vous pouvez définir dans un Script des points d'arrêt permettant d'arrêter en certaines endroits le traitement de ce Script et de démarrer le débogueur. Définissez par exemple un point d'arrêt devant une ligne sur laquelle vous soupçonnez une erreur dans le Script.

Vous pouvez:

- définir des points d'arrêt sur certaines lignes pour localiser étape par étape les erreurs logiques dans votre Script
- définir un point d'arrêt et appeler le débogueur avant que la ligne suivante du Script ne soit traitée. Utilisez cette procédure pour des événements tels que "Changement de vue".

Tous les points d'arrêt sont perdus lorsqu'un fichier Script est mis à jour dans le débogueur.

Si vous définissez un point d'arrêt dans l'un des fichiers Script "<Nom de la vue>.pdl_trigger" ou "<Nom de la vue>.pdl_évent", toutes les procédures déclenchées par Trigger ou déclenchées par événement sont arrêtées en Runtime.

Condition

Runtime est activé, la vue à déboguer est active.

Marche à suivre

Définir un point d'arrêt

1. Démarrez le débogueur et sélectionnez le Script. Si vous avez sélectionné l'activation automatique du débogueur dans WinCC, il est appelé dès qu'un Script incorrect est exécuté.
2. Placez le curseur dans l'action où vous voulez définir un point d'arrêt.
3. Dans le menu "Debug", sélectionnez la commande "Toggle Breakpoint" ou l'icône dans la barre d'outils.
   La prochaine ligne exécutable sera marquée d'un point rouge.
4. Permutez sur WinCC Runtime et exécutez l'action que vous voulez déboguer.
   Le débogueur s'arrête sur le premier point d'arrêt trouvé dans le Script. Le fond de la ligne actuelle est représenté en jaune. Vous pouvez à présent parcourir le Script étape par étape.

Voir aussi

Pour supprimer des points d'arrêt (Page 105)

Pour exécuter des instructions Script (Page 107)

Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)

Pour activer des signets dans le Script (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Nom de l'action et de la procédure dans Debugger (Page 99)

Structure des fichiers VBScript (Page 97)

Composants du Microsoft Script Debugger (Page 95)

Principes de base du débogage (Page 93)

Pour activer le débogueur (Page 91)

Tester avec Debugger (Page 90)

Diagnostic (Page 83)

1.12.4.10 Pour supprimer des points d'arrêt

Introduction

Lorsque vous avez éliminé une erreur, vous pouvez à nouveau supprimer quelques ou tous les points d'arrêt dans un Script.

Marche à suivre

1. Placez le curseur sur la ligne où se trouve le point d'arrêt à supprimer.
2. Dans le menu "Debug", sélectionnez la commande "Toggle Breakpoint" ou l'icône dans la barre d'outils.
   La ligne sera à nouveau affichée sans point rouge.
3. Pour effacer tous les points d'arrêt dans un Script, sélectionnez dans le menu "Debug" la commande "Clear all Breakpoints" ou l'icône dans la barre d'outils.

Voir aussi

Pour exécuter des instructions Script (Page 107)

Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)

Pour activer des signets dans le Script (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Nom de l'action et de la procédure dans Debugger (Page 99)

Structure des fichiers VBScript (Page 97)

Composants du Microsoft Script Debugger (Page 95)

Principes de base du débogage (Page 93)

Pour activer le débogueur (Page 91)

Tester avec Debugger (Page 90)

Diagnostic (Page 83)

1.12.4.11 Pour activer des signets dans le Script

Introduction

Pendant le débogage, vous pouvez activer des signets sur les lignes de code pour retrouver plus facilement ultérieurement une ligne déterminée.

Activer ou supprimer des signets

Placez à cet effet le pointeur de la souris sur la ligne où vous voulez activer un signet et appuyez <Ctrl+F2> soit pour activer un signet, soit pour le supprimer.

Sauter au signet suivant

<F2> vous permet de passer au signet suivant dans le Script.

Sauter au signet précédent

<Maj+F2> vous permet de retourner au signet précédent dans le Script.

Voir aussi

Pour exécuter des instructions Script (Page 107)
Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)
Pour supprimer des points d'arrêt (Page 105)
Pour définir des points d'arrêt (Page 103)
Pour sélectionner un Script en vue de l'éditer (Page 101)
Nom de l'action et de la procédure dans Debugger (Page 99)
Structure des fichiers VBScript (Page 97)
Composants du Microsoft Script Debugger (Page 95)
Principes de base du débogage (Page 93)
Pour activer le débogueur (Page 91)
Tester avec Debugger (Page 90)
Diagnostic (Page 83)

1.12.4.12 Pour déterminer et modifier des valeurs de variable ou de propriété

Introduction

Lorsqu'un Script fonctionne en Runtime, vous pouvez utiliser la fonction "Command Window" du débogueur pour déterminer et modifier les valeurs des variables ou des propriétés du Script en cours de fonctionnement. Vous pouvez ainsi par exemple remettre à zéro une valeur de process pour le Script, sans pour autant devoir arrêter le process.

Remarque

Si vous voulez déterminer le nom d'un objet WinCC en Runtime, appuyez <Ctrl+Alt+Maj> et amenez la souris sur l'objet en question. Le nom de la vue et le nom de l'objet s'affichent alors dans une info-bulle.

Condition

Le Script fonctionne en Runtime et le débogueur est ouvert.

Marche à suivre

1. Définissez au moins un point d'arrêt dans le Script actuel.
2. Permutez sur WinCC Runtime et déclenchez une action qui fait se dérouler le Script.
   Le débogueur s'arrête au premier point d'arrêt.
3. Activez dans le menu "View" l'inscription "Command Window".
   La "Command Window" apparaît.
4. Pour déterminer la valeur d'une variable ou d'une propriété, tapez "?", suivi d'un espace puis du nom de la variable ou de la propriété que vous voulez déterminer, par exemple "? myTag".
   Appuyez sur <OK> pour exécuter la commande.
5. Pour modifier la valeur d'une variable/d'une propriété, attribuez-lui une valeur dans la syntaxe VBS.

Voir aussi

Principes de base du débogage (Page 93)

Pour exécuter des instructions Script (Page 107)

Pour activer des signets dans le Script (Page 105)

Pour supprimer des points d'arrêt (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Nom de l'action et de la procédure dans Debugger (Page 99)

Structure des fichiers VBScript (Page 97)

Composants du Microsoft Script Debugger (Page 95)

Pour activer le débogueur (Page 91)

Tester avec Debugger (Page 90)

Diagnostic (Page 83)

1.12.4.13 Pour exécuter des instructions Script

Introduction

Pendant le fonctionnement d'un Script en Runtime, vous pouvez utiliser "Command Window" du débogueur pour exécuter directement des instructions Script et manipuler ainsi le déroulement du Script en cours. Vous pouvez exécuter directement les instructions Script pour les tester, sans générer l'instruction dans un Script, ni exécuter ce dernier. Vous pouvez p. ex.:

- Appeler des méthodes
- appeler des procédures
- manipuler les propriétés de l'objet

Dans "Command Window", vous pouvez en principe exécuter toutes les instructions que vous pouvez également exécuter dans VB-Script.

Condition

Le Script fonctionne en Runtime et le débogueur est ouvert.

Marche à suivre

1. Définissez au moins un point d'arrêt dans le Script actuel.
2. Permutez sur WinCC Runtime et déclenchez une action qui fait se dérouler le Script.
   Le débogueur s'arrête au premier point d'arrêt.
3. Activez dans le menu "View" l'inscription "Command Window".
   La "Command Window" apparaît.
4. Tapez l'instruction voulue et appuyez "ENTER".

Remarque

Si vous entrez une commande erronée dans Command Window, aucun message d'erreur ne sera généré dans Runtime. Par contre, le message "<Script Error>" apparaît dans la fenêtre Command.

Voir aussi

Pour déterminer et modifier des valeurs de variable ou de propriété (Page 106)

Pour activer des signets dans le Script (Page 105)

Pour supprimer des points d'arrêt (Page 105)

Pour définir des points d'arrêt (Page 103)

Pour sélectionner un Script en vue de l'éditer (Page 101)

Nom de l'action et de la procédure dans Debugger (Page 99)

Structure des fichiers VBScript (Page 97)

Composants du Microsoft Script Debugger (Page 95)

Principes de base du débogage (Page 93)

Pour activer le débogueur (Page 91)

Tester avec Debugger (Page 90)

Diagnostic (Page 83)

1.13 Pour imprimer des VB-Scripts

Principe

Vous pouvez documenter dans WinCC non seulement les actions et procédures configurées dans Global Script mais également celles configurées dans Graphics Designer.

On distingue les possibilités de documentation suivantes:

- Imprimer l'aperçu avant impression: Dans Graphics Designer, l'aperçu avant impression de la vue actuelle permet d'imprimer toutes les actions configurées. L'aperçu avant impression contient les actions C et VBS les unes à côté des autres et qui se distinguent par des indications différentes pour le texte source (C ou VBScript).
- Impression du Script actuel: L'aperçu avant impression dans Global Script contient toujours la procédure ou l'action actuellement ouverte.

Dans WinCC, on utilise toujours des layouts d'impression prédéfinis pour la mise en page de l'aperçu avant impression. Vous pouvez également concevoir vos propres layouts d'impression et les lier par la commande "Configurer documentation de projet" sur l'onglet Travail d'impression.

Marche à suivre

1. Ouvrez Global Script ou Graphics Designer selon les Scripts que vous voulez documenter.
2. Configurez au besoin le travail d'impression avec la commande "Configurer documentation de projet".
3. La commande "Aperçu documentation de projet" vous permet d'avoir un aperçu des données qui sont imprimées.
4. Sélectionnez la commande de menu "Fichier" > "Imprimer documentation de projet" pour imprimer les données.

Voir aussi

Création et édition d'actions (Page 57)

Création et édition de procédures (Page 40)

Les éditeurs VBScript (Page 28)

Utilisation de Visual Basic Script dans WinCC (Page 12)

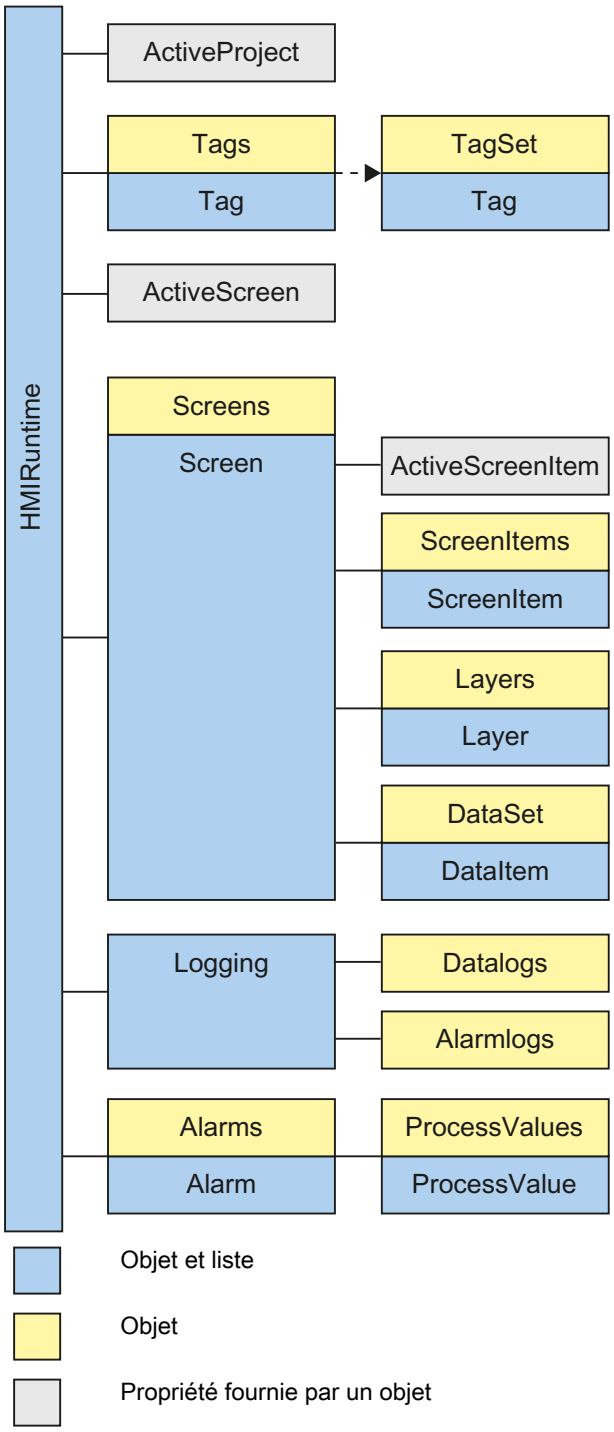
1.14 Référence VBS

1.14.1 Référence VBS

Le modèle d'objet VBS dans WinCC

Le modèle d'objet WinCC du système graphique de runtime vous permet d'accéder à des objets graphiques et à des variables en runtime.

Lorsque vous cliquez sur un nom d'objet, vous obtenez une description détaillée.



Le modèle d'objet VBS dans un type de bloc d'affichage

Dans un type de bloc d'affichage, le modèle d'objet VBS n'est pas valable pour WinCC. Il a été remplacé par un modèle entièrement nouveau.

Le modèle d'objet VBS du type de bloc d'affichage vous permet d'accéder aux objets graphiques et aux variables de bloc d'affichage du type de bloc d'affichage en runtime.

Objets

Vous accédez à tous les objets du système graphique Runtime par le biais des objets et des listes : Objets graphiques, vues, niveaux et variables.

Propriétés

Par le biais des propriétés des différents objets, vous pouvez modifier de façon ciblée des objets graphiques et des variables en runtime, p. ex. valider par un clic de souris un élément de commande ou déclencher un changement de couleur à la modification d'une valeur de variable.

Méthodes

Avec les méthodes que vous appliquez aux différents objets, vous pouvez p. ex lire des valeurs de variables pour un traitement ultérieur ou afficher des messages de diagnostic en runtime.

Voir aussi

Propriété ActiveScreen (Page 295)
Type d'objet de l'objet ScreenItem (Page 150)
Méthodes (Page 680)
Propriétés (Page 292)
Objets et listes (Page 114)
Objet AlarmLogs (Page 119)
Objet DataItem (Page 120)
Objet DataLogs (Page 122)
Objet DataSet (Liste) (Page 123)
Objet HMIRuntime (Page 125)
Objet Layer (Page 127)
Objet Layers (Liste) (Page 128)
Objet ScreenItem (Page 132)
Objet ScreenItems (Liste) (Page 136)
Objet Screen (Page 138)
Objet Screens (Liste) (Page 141)
Objet Tag (Page 144)
Objet Tags (Liste) (Page 147)
Objet TagSet (Liste) (Page 149)
Propriété ActiveProject (Page 294)
Propriété ActiveScreenItem (Page 295)
Objet Logging (Page 129)
Objet d'alarme (Page 117)
Objet d'alarmes (liste) (Page 118)
Objet ProcessValue (Page 130)
Objet ProcessValues (liste) (Page 131)

1.14.2 Objets et listes

1.14.2.1 Objets et listes

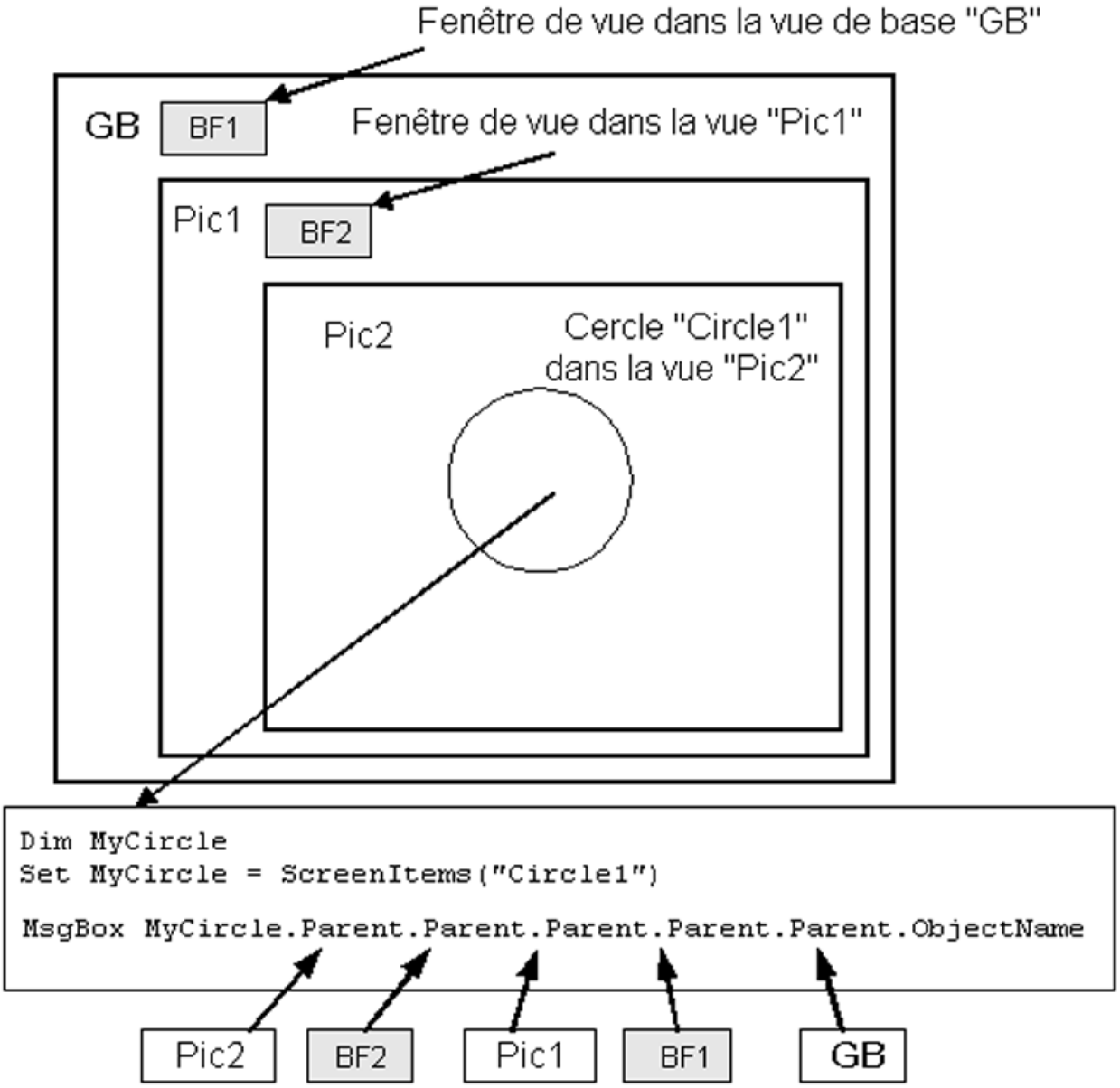
Liste récapitulative

Les objets et listes du modèle d'objet WinCC vous permettent d'accéder en Runtime aux objets graphique et aux variables.

Navigation dans un modèle d'objet

Vous accédez de façon hiérarchique aux objets au sein du modèle VBS Objet. Si vous accédez par exemple à un élément de vue au sein d'une vue, faites-le par le biais de son objet Parent (la vue environnante).

Exemple



Dans cet exemple, c'est le nom de la vue de base qui est affiché.

Accès aux objets graphiques

Dans WinCC, vous accédez aux vues, aux niveaux et aux objets graphiques dans Runtime via l'objet supérieur "HMIRuntime". L'accès aux objets et aux niveaux s'effectue toujours par la vue (Screen) dans laquelle ils se trouvent.

Accès aux variables

Dans WinCC, vous accédez directement aux variables dans Runtime via l'objet supérieur "HMIRuntime". Vous pouvez lire ou paramétrer à nouveau les valeurs des variables.

Listes

Les listes du modèle d'objet WinCC se comportent comme les collections standard de VBS. Exception : la liste "Tags" n'a pas de fonction Enum.

Objets disponibles

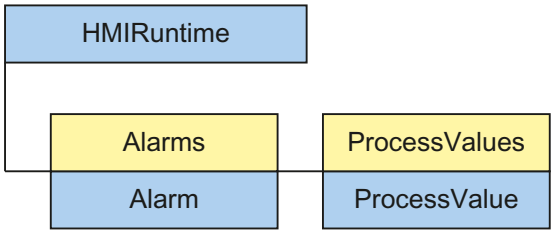
- Alarme
- Alarmes
- AlarmLogs
- DataItem
- DataLogs
- DataSet
- HMIRuntime
- Item
- Layer
- Layers
- Logging
- ProcessValues
- ProcessValue
- Project
- ScreenItem
- ScreenItems
- Screen
- Screens
- Jour
- Tags
- TagSet

Voir aussi

Objet ScreenItems (Liste) (Page 136)
Objet TagSet (Liste) (Page 149)
Objet Tags (Liste) (Page 147)
Objet Tag (Page 144)
Objet Screens (Liste) (Page 141)
Objet Screen (Page 138)
Objet ScreenItem (Page 132)
Objet Layers (Liste) (Page 128)
Objet Layer (Page 127)
Objet Item (Page 126)
Objet HMIRuntime (Page 125)

1.14.2.2 Objet d'alarme

Description



L'objet d'alarme est utilisé pour accéder à la liste des objets d'alarme.

Remarque

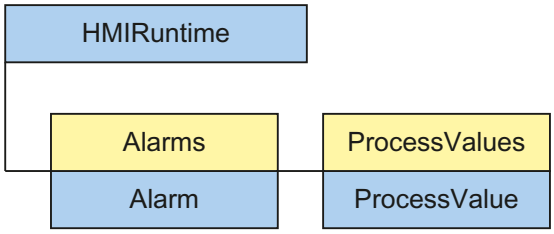
Les propriétés de l'objet d'alarme ne sont pas automatiquement actualisées lorsque les valeurs des propriétés changent.

Voir aussi

Objet d'alarmes (liste) (Page 118)

1.14.2.3 Objet d'alarmes (liste)

Description



L'objet d'alarme permet de déclencher les alarmes existantes.

Utilisation

Par la liste "Alarms", vous pouvez :

- accéder à une alarme dans la liste (méthode Item)
- générer un nouvel objet d'alarme (méthode Create)
- lire l'AlarmID de l'alarme (propriété AlarmID)
- lire l'état d'une alarme (propriété State)
- lire l'horodatage de l'alarme (propriété Timestamp)
- générer une instance de l'objet d'alarme (propriété Instance)
- lire le nom de l'ordinateur dans lequel l'alarme est apparue (propriété ComputerName)
- lire ou définir le nom de l'utilisateur qui a déclenché l'alarme (propriété UserName)
- lire ou définir les noms des blocs de valeur de process (propriété ProcessValues)
- lire ou définir le commentaire de l'alarme (propriété Comment)
- lire ou définir le préfixe de serveur de l'alarme (propriété Context)

Exemple

L'exemple suivant montre comment déclencher l'alarme configurée dans l'éditeur Alarm Logging à l'aide du numéro d'alarme "1".

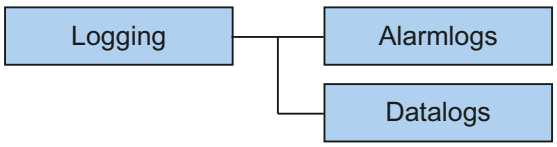
```
'VBS360
Dim MyAlarm
Set MyAlarm = HMIRuntime.Alarms(1)
MyAlarm.State = 5 'hmiAlarmStateCome + hmiAlarmStateComment
MyAlarm.Comment = "MyComment"
MyAlarm.UserName = "Hans-Peter"
MyAlarm.ProcessValues(1) = "Process Value 1"
MyAlarm.ProcessValues(4) = "Process Value 4"
MyAlarm.Create "MyApplication"
```

Voir aussi

Propriété TimeStamp (Page 595)
Propriété ComputerName (Page 367)
Propriété Context (Page 367)
Propriété State (Page 560)
Propriété AlarmID (Page 299)
Propriété Instance (Page 427)
Propriété Comment (Page 366)
Propriété UserName (Page 648)
Propriété ProcessValue (Page 517)
Objet d'alarme (Page 117)
Objet ProcessValues (liste) (Page 131)
Méthode Create (Page 687)
Méthode Item (Page 740)

1.14.2.4 Objet AlarmLogs

Description



Cet objet permet de lier à nouveau au Runtime des segments d'archive d'Alarm Logging transférés ou de supprimer à nouveau les segments d'archive d'Alarm Logging introduits. Les opérations suivantes sont alors effectuées

- les segments d'archive à introduire sont copiés dans le répertoire Common Archiving du projet WinCC, ou
- les segments d'archive auparavant introduits sont supprimés du répertoire Common Archiving.

Les paramètres vous permettent de contrôler à partir d'où les segments d'archive doivent être introduits. Vous pouvez également déterminer la période pendant laquelle les segments d'archive doivent être introduits ou supprimés. Les segments d'archive sont alors copiés dans le répertoire Common Archiving du projet.

La méthode utilisée renvoie un message d'erreur si une erreur se produit lors du traitement des segments d'archive. Vous trouverez plus d'informations à ce sujet dans la rubrique "Messages d'erreur dans la zone Base de données".

Utilisation

Les segments d'archive d'Alarm Logging auparavant transférés peuvent être reliés au Runtime (Méthode "Restore").

Les segments d'archive d'Alarm Logging auparavant introduits peuvent être supprimés du projet Runtime (Méthode "Remove")

Exemple

Dans l'exemple qui suit, les segments d'archive d'Alarm Logging sont introduits et la valeur de retour est affichée en tant que Trace.

```
'VBS187
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.AlarmLogs.Restore("D:
\Folder","2004-09-14","2004-09-20",-1) & vbNewLine
```

Voir aussi

Messages d'erreur dans la zone Bases de données (Page 790)

Méthode Restore (Page 764)

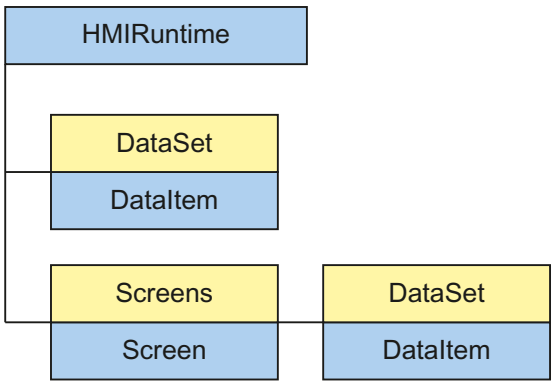
Méthode Remove (Page 758)

Objet DataLogs (Page 122)

Objet Logging (Page 129)

1.14.2.5 Objet DataItem

Description



L'objet DataItem est utilisé pour accéder au contenu de la liste DataSet. Valeurs ou références d'objets sont inscrites dans la liste comme DataItem.

L'accès s'effectue par le nom sous lequel la valeur a été ajoutée dans la liste. L'accès individuel par l'index n'est pas recommandé étant donné que l'index change à chaque fois que des

valeurs sont ajoutées et supprimées. L'index peut être utilisé pour afficher le contenu intégral de la liste. L'affichage se fait dans l'ordre alphabétique.

Remarque

Dans le cas des références d'objet, assurez-vous que les objets sont aptes au 'multithread'.

Exemple

L'exemple montre comment la valeur de 'Moteur1' est affichée en tant que Trace.

```
'VBS163
HMIRuntime.Trace "motor1: " & HMIRuntime.DataSet("motor1").Value & vbNewLine
```

L'exemple suivant énumère tous les objets DataItem de la liste DataSet. Name et Value sont affichés en tant que Trace.

```
'VBS164
Dim data
For Each data In HMIRuntime.DataSet
HMIRuntime.Trace data.Name & ": " & data.Value & vbNewLine
Next
```

Remarque

Pour les objets, il se peut que Value ne puisse être directement affiché.

Voir aussi

Objet Screen (Page 138)

Objet HMIRuntime (Page 125)

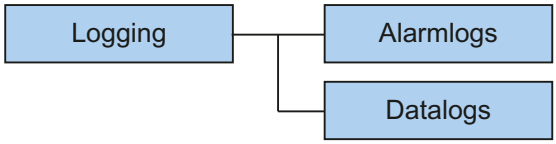
Objet DataSet (Liste) (Page 123)

Propriété Value (Page 652)

Propriété Name (Page 482)

1.14.2.6 Objet DataLogs

Description



Cet objet permet de lier à nouveau au Runtime des segments d'archive de Tag Logging transférés ou de supprimer à nouveau les segments d'archive de Tag Logging introduits. Les opérations suivantes sont alors effectuées

- les segments d'archive à introduire sont copiés dans le répertoire Common Archiving du projet WinCC, ou
- les segments d'archive auparavant introduits sont supprimés du répertoire Common Archiving.

Les paramètres vous permettent de contrôler à partir d'où les segments d'archive doivent être introduits. Vous pouvez également déterminer la période pendant laquelle les segments d'archive doivent être introduits ou supprimés. Vous pouvez en outre paramétrer le type d'archive ("Tag Logging Fast", "Tag Logging Slow", "Tag Logging Fast et Tag Logging Slow"). Les segments d'archive sont alors copiés dans le répertoire Common Archiving du projet.

La méthode utilisée renvoie un message d'erreur si une erreur se produit lors du traitement des segments d'archive. Vous trouverez plus d'informations à ce sujet dans la rubrique "Messages d'erreur dans la zone Base de données".

Utilisation

Les segments d'archive Tag Logging auparavant transférés peuvent être reliés au Runtime (Méthode "Restore").

Les segments d'archive Tag Logging peuvent être supprimés du projet Runtime (Méthode "Remove")

Exemple

Dans l'exemple qui suit, les segments d'archive Tag Logging Fast sont introduits et la valeur de retour est affichée en tant que Trace.

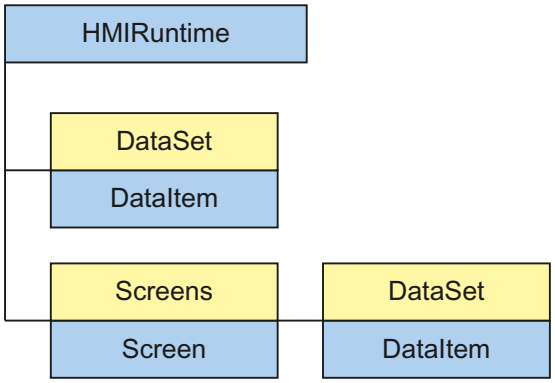
```
'VBS188
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.DataLogs.Restore("D:
\Folder","2004-09-14","2004-09-20",-1,1) & vbNewLine
```

Voir aussi

Messages d'erreur dans la zone Bases de données (Page 790)
Méthode Restore (Page 764)
Méthode Remove (Page 758)
Objet AlarmLogs (Page 119)
Objet Logging (Page 129)

1.14.2.7 Objet DataSet (Liste)

Description



L'objet DataSet permet d'échanger des données pendant plusieurs actions.

Un objet DataSet est global et défini au niveau de l'objet Screen. Il est possible d'accéder aux données depuis n'importe quelle action VBS.

L'objet DataSet de l'objet Screen doit être adressé selon la hiérarchie des vues et est maintenu tant que la vue est affichée. L'objet global est maintenu pendant toute la durée du Runtime.

L'accès s'effectue par l'objet DataItem.

Remarque

Les objets de type Screen, Screens, ScreenItem, ScreenItems, Tag et TagSet ne peuvent pas être acceptés dans la liste DataSet.
L'objet DataSet n'est compatible avec aucune des classes.

Utilisation

Ce que vous pouvez faire par le biais de la liste "DataSet" :

- Afficher ou éditer (énumérer) tous les objets de la liste.
- Afficher le nombre d'éléments contenus (Propriété "Count").

- Editer un objet déterminé de la liste (Méthode "Item").
- Ajouter un objet à la liste (Méthode "Add").
- Supprimer un objet déterminé de la liste (Méthode "Remove").
- Supprimer tous les objets de la liste (Méthode "RemoveAll").

L'accès aux éléments de la liste s'effectue de la façon suivante :

```
HMIRuntime.DataSet("Itemname")
```

S'il s'agit d'une liste spécifique de la vue, l'accès s'effectue par :

```
HMIRuntime.Screens("Screenname").DataSet("Itemname")
```

Dans une vue, vous pouvez accéder à l'objet DataSet de la vue par :

```
DataSet("Itemname")
```

Lors de l'accès, si le nom spécifié n'existe pas dans la liste, VT_Empty est renvoyé et une Exception se déclenche.

Exemple

L'exemple montre comment enregistrer, lire et supprimer une valeur dans la liste. Il est logique que ces opérations s'effectuent au cours d'actions séparées.

```
'VBS162
HMIRuntime.DataSet.Add "motor1", 23
HMIRuntime.Trace "motor1: " & HMIRuntime.DataSet("motor1").Value & vbNewLine
HMIRuntime.DataSet.Remove("motor1")
```

Voir aussi

Objet DataItem (Page 120)

Méthode RemoveAll (Page 762)

Méthode Remove (Page 758)

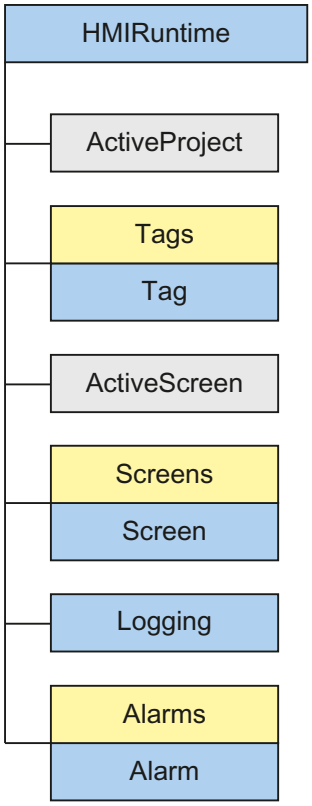
Méthode Item (Page 740)

Propriété Count (Page 368)

Méthode Add (Page 684)

1.14.2.8 Objet HMIRuntime

Description



L'objet HMIRuntime représente l'environnement graphique Runtime.

Utilisation

Ce que vous pouvez par exemple faire via l'objet "HMIRuntime" :

- lire ou paramétrer la langue Runtime actuelle (Propriété "Language").
- lire ou paramétrer le nom de la vue de base actuelle (Propriété "BaseScreenName").
- lire le chemin du projet Runtime activé (Propriété "ActiveProject").
- accéder à des variables (Propriété "Tags").
- accéder à des variables d'une liste (Propriété "DataSet").
- quitter Runtime (Méthode "Stop").
- afficher des alarmes dans une fenêtre de diagnostic (Méthode "Trace").

Exemple

La commande suivante vous permet de quitter WinCC Runtime :

```
'VBS3
HMIRuntime.Stop
```

Voir aussi

Objet Screens (Liste) (Page 141)
Objet TagSet (Liste) (Page 149)
Objet Tags (Liste) (Page 147)
Objet Logging (Page 129)
Objet DataSet (Liste) (Page 123)
Propriété Visible (Page 667)
Methode Trace (Page 781)
Propriété Tags (Page 569)
Méthode Stop (Page 780)
Propriété AlignmentLeft (Page 300)
Propriété Logging (Page 457)
Propriété Language (Page 431)
Propriété DataSet (Page 373)
Propriété CurrentContext (Page 369)
Propriété BaseScreenName (Page 320)
Propriété ActiveProject (Page 294)
Propriété ActiveScreen (Page 295)
Propriété-MenuToolBarConfig (Page 468)
Objet d'alarmes (liste) (Page 118)

1.14.2.9 Objet Item

Description

L'objet "Item" vous fournit une référence sur l'objet actuel.

Utilisation

Utilisez l'objet "Item" pour par exemple accéder aux propriétés de l'objet actuellement sélectionné dans le Graphics Designer.

Exemple

Dans l'exemple suivant, vous avez par exemple créé un rectangle. Lorsque l'objet est sélectionné, vous pouvez accéder à toutes les propriétés de l'objet actuel, ici il s'agit de régler la couleur de l'arrière-plan en rouge:

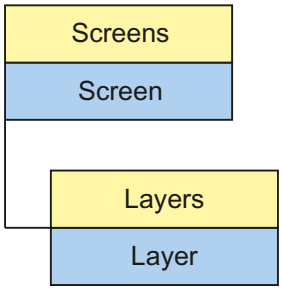
```
'VBS195
    Item.BackColor = RGB(255,0,0)
```

Voir aussi

Objets et listes (Page 114)

1.14.2.10 Objet Layer

Description



L'objet Layer vous est restitué comme résultat après l'accès à la liste des Layers.

Objet parent

Vue dans laquelle se trouve le niveau de vue.

Utilisation

L'objet Layer vous permet d'accéder aux propriétés d'un niveau entier en fonction d'événements déterminés, par exemple pour afficher ou supprimer un niveau avec des organes de commande en fonction de l'autorisation de conduite.

Ce que vous pouvez faire par le biais de l'objet "Layer":

- Afficher ou supprimer la visibilité d'un niveau (Propriété "Visible").
- Lire le nom d'un niveau (Propriété "Name").

Remarque

La propriété Layer affiche le niveau où se trouve l'objet. Le niveau "0" s'affiche Niveau"0".

En cas d'accès, les niveaux sont comptés dans VBS à partir de 1. C'est pourquoi on accède au niveau "1" avec layers(2).

Exemple

Dans l'exemple suivant, Niveau 1 est réglé sur invisible:

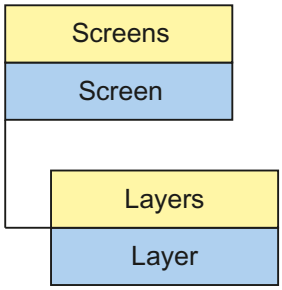
```
'VBS4
Layers(2).Visible = vbFalse
```

Voir aussi

Objet Layer (Page 127)
Propriété Visible (Page 667)
Propriété Parent (Page 501)
Propriété Name (Page 482)

1.14.2.11 Objet Layers (Liste)

Description



La liste des Layers vous permet d'accéder aux 32 niveaux du système graphique Runtime.

Objet parent

Vue dans laquelle se trouve le niveau de vue.

Utilisation

Ce que vous pouvez faire par le biais de la liste "Layers" :

- Editer tous les niveaux au sein de la liste (Propriété "_NewEnum").
- Compter tous les niveaux contenus dans la liste (Propriété "Count").
- Editer un niveau de la liste (Méthode "Item").

Les propriétés sont des propriétés standard et les méthodes d'une liste, et ne sont pas décrites en détail dans la documentation WinCC.

Voir aussi

Propriété Parent (Page 501)

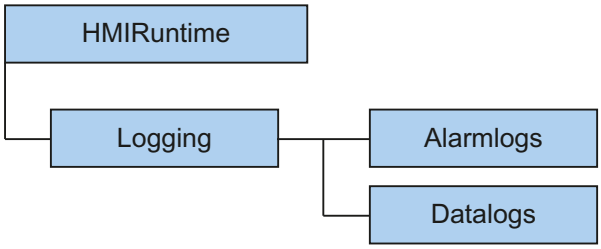
Méthode Item (Page 740)

Propriété Count (Page 368)

Objet Layer (Page 127)

1.14.2.12 Objet Logging

Description



Cet objet permet de lier à nouveau au Runtime des segments d'archive transférés ou de supprimer à nouveau les segments d'archive auparavant introduits. Les opérations suivantes sont alors effectuées

- les segments d'archive à introduire sont copiés dans le répertoire Common Archiving du projet WinCC, ou
- les segments d'archive auparavant introduits sont supprimés du répertoire Common Archiving.

Les paramètres vous permettent de contrôler à partir d'où les segments d'archive doivent être introduits. Vous pouvez également déterminer la période pendant laquelle les segments d'archive doivent être introduits ou supprimés. Les segments d'archive sont alors copiés dans le répertoire Common Archiving du projet.

La méthode utilisée renvoie un message d'erreur si une erreur se produit lors du traitement des segments d'archive. Vous trouverez plus d'informations à ce sujet dans la rubrique "Messages d'erreur dans la zone Base de données".

Utilisation

Les segments d'archive d'Alarm Logging et Tag Logging auparavant transférés peuvent être reliés au Runtime (Méthode "Restore").

Les segments d'archive d'Alarm Logging et Tag Logging auparavant introduits peuvent être supprimés du projet Runtime (Méthode "Remove").

Exemple

Dans l'exemple qui suit, les segments d'Alarm et Tag Logging sont introduits et la valeur de retour est affichée en tant que Trace.

```
'VBS189
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Restore("D:
\Folder","2004-09-14","2004-09-20",-1) & vbNewLine
```

Voir aussi

Messages d'erreur dans la zone Bases de données (Page 790)

Objet DataLogs (Page 122)

Objet AlarmLogs (Page 119)

Méthode Restore (Page 764)

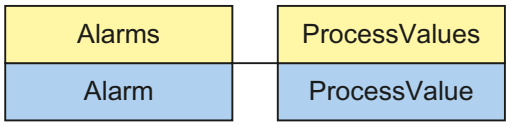
Méthode Remove (Page 758)

Propriété DataLogs (Page 373)

Propriété AlarmLogs (Page 299)

1.14.2.13 Objet ProcessValue

Description



L'objet ProcessValue est utilisé pour accéder à la liste des objets ProcessValue.

Remarque

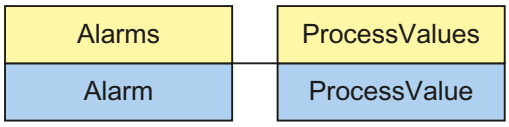
Seules les dix ProcessValues prédéfinies sont supportées.

Voir aussi

Objet ProcessValues (liste) (Page 131)

1.14.2.14 Objet ProcessValues (liste)

Description



Utilisation

Par la liste "ProcessValues", vous pouvez :

- éditer une ProcessValue de la liste (méthode Item)
- afficher ou éditer tous les objets au sein de la liste (propriété _NewEnum)
- compter les ProcessValues contenues dans la liste (propriété Count)
- lire ou définir les valeurs de l'objet ProcessValue (propriété Value)

Les propriétés sont des propriétés standard et des méthodes dans une liste et ne sont pas décrites en détail dans la documentation.

Voir aussi

Objet d'alarmes (liste) (Page 118)

Objet ProcessValue (Page 130)

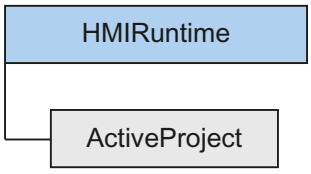
Propriété Count (Page 368)

Propriété Value (Page 652)

Méthode Item (Page 740)

1.14.2.15 Objet Project

Description



Cet objet vous permet d'interroger des informations du projet Runtime actuel.

L'objet Project est restitué comme résultat d'ActiveProject.

Utilisation

Ce que vous pouvez faire par le biais de l'objet "Project" :

- lire le chemin du projet Runtime actuel (Propriété "Path").
- lire le nom du projet Runtime actuel sans l'indication du chemin, ni de l'extension de fichier (Propriété "Name").

Exemple

L'exemple suivant affiche le nom et le chemin du projet Runtime actuel en tant que Trace :

```
'VBS159
HMIRuntime.Trace "Name: " & HMIRuntime.ActiveProject.Name & vbNewLine
HMIRuntime.Trace "Path: " & HMIRuntime.ActiveProject.Path & vbNewLine
```

Voir aussi

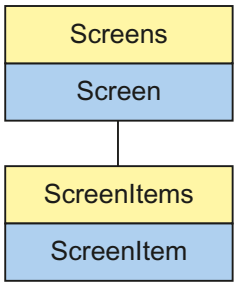
Propriété ActiveProject (Page 294)

Propriété Name (Page 482)

Propriété Path (Page 504)

1.14.2.16 Objet ScreenItem

Description



L'objet ScreenItem vous est restitué comme résultat après l'accès à la liste des ScreenItems.

Objet parent

Vue dans laquelle se trouve l'élément de vue.

Utilisation

L'objet ScreenItem vous permet d'accéder aux propriétés des objets graphiques au sein d'une vue, et ce en fonction d'événements déterminés.

Ce que vous pouvez par exemple faire via l'objet "ScreenItem" :

- Afficher ou désactiver la visibilité d'un objet (Propriété "Visible").
- Libérer ou bloquer un objet pour la commande (Propriété "Enabled").
- Modifier la largeur et la hauteur d'un objet (Propriété "Height" et "Width").
- Modifier la position d'un objet (Propriété "Top" et "Left").
- Lire et définir le niveau où se trouve un objet graphique (Propriété "Layer").
- Lire ou définir le nom d'un objet graphique (Propriété "ObjectName").
- Attribuer une référence à la vue supérieure (Propriété "Parent").

Avec la méthode "Activate", le focus est positionné sur l'objet ScreenItem respectif. Une erreur est générée si le focus ne peut pas être positionné, p. ex. parce que l'objet ne peut être manipulé. L'erreur peut être analysée par une fonction de traitement des erreurs (On Error Resume Next).

Réglages possibles pour ScreenItem

L'objet "ScreenItem" peut contenir les types d'objet suivants :

Objets standard	Objets complexes	Objets Windows	Objets tuyaux	Contrôles	Autres
Ellipse	Bargraphes tridimensionnels	Bouton	Double raccord en T	Siemens HMI Symbol Library	Objet utilisateur
Arc d'ellipse	Fenêtre d'application	Case à cocher	Tuyau polygonal	WinCC AlarmControl	Groupe
Segment d'ellipse	Bargraphe	Case d'option	Coude de tuyau	WinCC Digital/ Analog Clock Control	
Cercle	Fenêtre de vue	Bouton rond	Raccord en T	WinCC FunctionTrendControl	
Arc de cercle	Contrôle	Slider		WinCC Gauge Control	
Segment de cercle	Champ d'E/S			WinCC OnlineTrendControl	
Trait	Instance de bloc d'affichage			WinCC OnlineTableControl	
Polygone	Objet graphique			WinCC Push Button Control	
Trait polygonal	Champ de combinaison			WinCC RulerControl	

Objets standard	Objets complexes	Objets Windows	Objets tuyaux	Contrôles	Autres
Rectangle	Champ de liste			WinCC Slider Control	
Rectangle arrondi	Texte multiligne			WinCC UserArchiveControl	
Connecteur	Objet OLE				
	Signalisation groupée				
	Liste de textes				
	Indicateur d'état				

Vous trouverez sous "Types d'objet ScreenItem" une description détaillé de chacun des types d'objet. La propriété "Type" de l'objet ScreenItem vous permet d'accéder aux types d'objet via l'identification du type VBS.

Propriétés des objets

L'objet "ScreenItem" dispose, selon le réglage, de diverses propriétés. Les propriétés que possède chaque type d'objet de ScreenItem sont indiquées ci-après :

```
Sub OnClick(ByVal Item)
Dim obj
Set obj = ScreenItems("Circle1").
End Sub
```

Si vous accédez à un type d'objet déterminé, ses propriétés standard sont complétées par d'autres propriétés :

```
Sub OnClick(ByVal Item)
Dim obj
Set obj = ScreenItems("Circle1")
obj.
End
    Activate
    BackColor
    BackFlashColorOff
    BackFlashColorOn
    BorderBackColor
    BorderColor
    BorderFlashColorOff
    BorderFlashColorOn
    BorderStyle
    BorderWidth
```

Ces propriétés supplémentaires sont mentionnées dans la description des divers types d'objet.

Exemple

Dans l'exemple suivant, vous réglez en Runtime le rayon d'un cercle sur 2 en cliquant sur la souris :

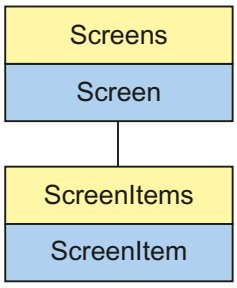
```
Sub OnClick(ByVal Item)
'VBS5
Dim objCircle
Set objCircle= ScreenItems("Circle1")
objCircle.Radius = 2
End Sub
```

Voir aussi

Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété Parent (Page 501)
Propriété Left (Page 449)
Propriété Layer (Page 433)
Propriété Height (Page 417)
Propriété Enabled (Page 382)
Méthode Activate (Page 682)
Exemple : Pour lire les valeurs variable (Page 799)
Exemple : Pour écrire les valeurs variable (Page 797)
Propriétés (Page 292)
Objets et listes (Page 114)
Type d'objet de l'objet ScreenItem (Page 150)

1.14.2.17 Objet ScreenItems (Liste)

Description



La liste "ScreenItems" vous permet de référencer un objet sur la vue.

Objet parent

Vue dans laquelle se trouve l'élément de vue.

Utilisation

Ce que vous pouvez faire par le biais de la liste "ScreenItems" :

- Afficher ou éditer tous les objets (tous les objets d'une vue) (Propriété "_NewEnum").
- Compter les objet d'une vue (Propriété "Count").
- Editer un objet déterminé de la liste (Méthode "Item").

Les propriétés sont des propriétés standard et les méthodes d'une Collection, et ne sont pas décrites en détail dans la documentation WinCC.

Particularités de l'objet ScreenItem

Si vous incorporez un contrôle externe (ActiveX-Control ou objet OLE) dans WinCC, il se peut que les propriétés du contrôle incorporé aient les mêmes noms que les propriétés générales de l'objet ScreenItem. Dans ce cas, ce sont les propriétés ScreenItem qui ont priorité.

Vous pouvez cependant accéder aux propriétés de contrôle incorporé via la propriété "object" :

La propriété "object" existe uniquement pour les ActiveX Controls et les objets OLE.

Exemple :

```
'Control1 is an embedded ActiveX-Control with property "type"
'VBS196
Dim Control
Set Control=ScreenItems("Control1")
Control.object.type

'Control1 is a WinCC-Control
'VBS197
Dim Control
Set Control=ScreenItems("Control1")
Control.type
```

Exemple

Dans l'exemple suivant, vous affichez dans une boîte à messages le nom des objets contenus sur l'image actuelle :

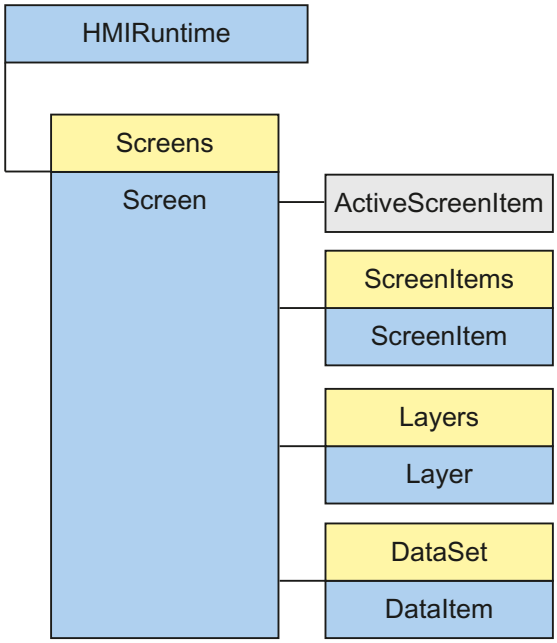
```
Sub OnClick(ByVal Item)
'VBS6
  Dim lngAnswer
  Dim lngIndex
  lngIndex = 1
  For lngIndex = 1 To ScreenItems.Count
    lngAnswer = MsgBox(ScreenItems(lngIndex).Objectname, vbOKCancel)
    If vbCancel = lngAnswer Then Exit For
  Next
  End Sub
```

Voir aussi

Propriété Count (Page 368)
Exemple : Pour lire les valeurs variable (Page 799)
Exemple : Pour écrire les valeurs variable (Page 797)
Objet ScreenItem (Page 132)
Propriété Parent (Page 501)
Méthode Item (Page 740)

1.14.2.18 Objet Screen

Description



L'objet Screen vous est restitué comme résultat après l'accès à la liste des Screens. Vous pouvez également éditer directement en Runtime toutes les propriétés et méthodes de cet objet. L'objet "Screen" représente une vue WinCC en Runtime et contient toutes les propriétés du document et de l'affichage de vue.

L'objet "Screen" contient en outre :

- une liste de tous les objets graphiques contenus dans la vue en question, et auxquels vous pouvez accéder via l'objet "ScreenItems"
- une liste de tous les niveaux contenus dans la vue en question, et auxquels vous pouvez accéder via l'objet "Layers"

Objet parent

La fenêtre de vue dans laquelle est incorporé l'objet Screen.

Si l'objet Screen est la vue de base, l'objet Parent n'est alors pas défini et positionné sur zéro.

Utilisation

Ce que vous pouvez par exemple faire via l'objet "Screen" :

- Libérer ou bloquer une vue pour la commande (Propriété "Enabled").
- Modifier la largeur et la hauteur d'une vue (Propriété "Width" et "Height").
- Editer le zoom d'une vue (Propriété "Zoom").
- Modifier le motif de remplissage, la couleur d'arrière-plan et la couleur du motif de remplissage (Propriétés "Fillstyle", "Backcolor" et "Fillcolor").

Remarque

En cas de changement de vue, les références ouvertes ne sont plus valables pour les vues n'existant plus. Vous ne pouvez donc plus travailler avec ces références.

Exemple

Dans l'exemple suivant, on augmente de 20 Pixel la largeur de la première vue en Runtime :

```
'VBS7
Dim objScreen
Set objScreen = HMIRuntime.Screens(1)
MsgBox "Screen width before changing: " & objScreen.Width
objScreen.Width = objScreen.Width + 20
MsgBox "Screen width after changing: " & objScreen.Width
```

Remarques sur CrossReference

Toutes les vues auxquelles vous accédez avec la formule standard

```
HMIRuntime.BaseScreenName = "Screenname"
```

Sont automatiquement saisies par la CrossReference de WinCC pour être mentionnées dans les propriétés de la vue.

Si vous accédez à des vues avec d'autres formules dans votre code, vous pouvez les présenter à CrossReference via la section suivante :

```
' ' WINCC:SCREENNAME_SECTION_START
Const ScreenNameInAction = "ScreenName"
' WINCC:SCREENNAME_SECTION_END
Vous pouvez insérer cette section dans des actions VBS autant de fois que vous le voulez.
```

Remarque

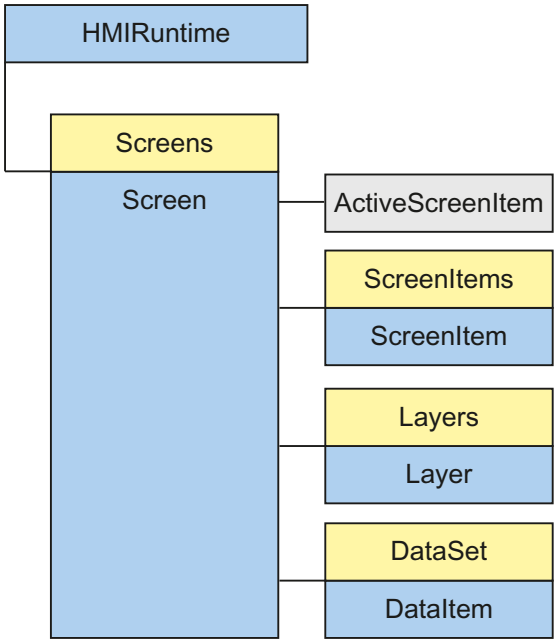
Pour des raisons de compatibilité avec les versions ultérieures, il est toujours préférable d'écrire les noms de vue sans l'extension de fichier ".PDL" .

Voir aussi

Propriété ScreenItems (Page 533)
Méthode Refresh (Page 758)
Méthode Activate (Page 682)
Propriété Zoom (Page 680)
Propriété Width (Page 669)
Propriété Parent (Page 501)
Propriété ObjectSizeDeclutteringMin (Page 487)
Propriété ObjectSizeDeclutteringMax (Page 486)
Propriété ObjectSizeDeclutteringEnable (Page 486)
Propriété ObjectName (Page 485)
Propriété Layers (Page 449)
Propriété DataSet (Page 373)
Propriété LayerDeclutteringEnable (Page 449)
Propriété Height (Page 417)
Propriété FillStyle (Page 394)
Propriété FillColor (Page 392)
Propriété ExtendedZoomingEnable (Page 390)
Propriété Enabled (Page 382)
Propriété BackColor (Page 313)
Propriété ActiveScreenItem (Page 295)
Propriété AccessPath (Page 293)

1.14.2.19 Objet Screens (Liste)

Description



La technique des fenêtres de vue vous permet d'ouvrir simultanément plusieurs vues dans WinCC Runtime, avec cependant une seule vue de base. La liste "Screens" permet d'accéder en Runtime à toutes les vues ouvertes via le nom de vue. La liste Screens contient également toutes les vues invisibles.

Utilisation

Lorsque vous configurez un projet multiposte, il vous faut spécifier le ServerPrefix quand vous accédez à une vue qui ne se trouve pas sur l'ordinateur.

Ce que vous pouvez faire par le biais de la liste "Screens" :

- afficher ou éditer toutes les vues au sein de la liste (Propriété "_NewEnum").
- compter les vues d'un projet (Propriété "Count").
- éditer une vue déterminée de la liste (Méthode "Item").
- faire redessiner toutes les vues visibles (Méthode "Refresh").

Les propriétés sont des propriétés standard et les méthodes d'une Collection, et ne sont pas décrites en détail dans la documentation WinCC.

Le code d'accès nécessaire dans l'environnement VBS, dans l'instruction HMIRuntime.Screens(<Code d'accès>) doit satisfaire à la description syntaxique suivante :

```
[<Nom de la vue de base>.]<Nom de la fenêtre de vue>[:<Nom de la
vue>] ...
```

```
.<Nom de la fenêtre de vue>[:<Nom de la vue>]
```

Cela signifie que :

- Le code d'accès exprime la hiérarchie des vues.
- Dans le code, l'omission des noms de vue est possible sur chaque position.
- La propriété "AccessPath" de l'objet "Screen" correspond au code d'accès intégral.
- Pour des raisons de compatibilité avec les versions ultérieures, il est toujours préférable d'écrire les noms de vue sans l'extension de fichier ".PDL" .
- Vous pouvez accéder à la vue de base via le code d'accès "".

L'accès à la vue de base avec Index 1 a en outre été défini.

Exemples

Les vues sont adressées en indiquant la hiérarchie dans la liste. Deux possibilités se présentent alors, avec ou sans utilisation du nom de vue. Dans les exemples suivants, une vue de base "BaseScreenName" est configurée avec une fenêtre de vue "ScreenWindow". Une vue "ScreenName" se trouve dans la fenêtre de vue.

Adressage avec utilisation du nom de vue

```
'VBS8
Set objScreen = HMIRuntime.Screens("BaseScreenName.ScreenWindow:ScreenName")
```

Adressage sans utilisation du nom de vue

```
'VBS9
Set objScreen = HMIRuntime.Screens("ScreenWindow")
```

Référencer la vue de base de différentes manières

```
'VBS10
Set objScreen = HMIRuntime.Screens(1)

'VBS11
Set objScreen = HMIRuntime.Screens("")

'VBS12
Set objScreen = HMIRuntime.Screens("BaseScreenName")
```

Voir aussi

Objet ScreenItem (Page 132)

Méthode Refresh (Page 758)

Méthode Item (Page 740)

Propriété Count (Page 368)

1.14.2.20 Objet SmartTags

Description

Dans le type de bloc d'affichage, le composant "HMIRuntime" a été désactivé. Le nouveau composant "SmartTags" a été ajouté pour le type de bloc d'affichage. Avec l'objet SmartTags, vous pouvez dynamiser le type de bloc d'affichage. Vous ne pouvez accéder qu'aux variables de bloc d'affichage et aux propriétés du type de bloc d'affichage. Vous ne pouvez pas accéder au stock normal de variables WinCC. Le stock normal de variables WinCC n'est pas disponible dans le type de bloc d'affichage.

Utilisation

Avec l'objet "SmartTags", vous pouvez :

- Accéder aux variables de bloc d'affichage dans un type de bloc d'affichage.
  Syntaxe : SmartTags("<Nom de variable>")
- Accéder aux propriétés d'un type de bloc d'affichage.
  Syntaxe : SmartTags("Properties\<Propertyname>")

Exemple 1

Dans un type de bloc d'affichage, vous ajoutez un rectangle et un bouton. Définissez une variable de bloc d'affichage var1. Reliez la propriété "Width" du rectangle avec la variable de bloc d'affichage var1. Dynamisez l'événement "OnClick" du bouton comme suit avec VBS.

```
'VBS306
Dim w
w = SmartTags("var1")
w = w + 10
SmartTags("var1") = w
```

Si vous activez le runtime, la variable de bloc d'affichage est incrémentée de 10 à chaque clic sur le bouton. La largeur du rectangle est alors augmentée de 10.

Lecture et écriture directes avec référence d'objet

Dans l'exemple ci-dessous, vous déterminez une référence "w" à "var1" au moyen de l'objet SmartTags.

Le référençage donne l'avantage de pouvoir accéder directement à la variable "var1".

```
'VBS307
Dim w
Set w = SmartTags("var1")
w.value = w.value + 10
```

Exemple 2

Dans un type de bloc d'affichage, vous ajoutez un rectangle et un bouton. Définissez la propriété spécifique à l'instance "wide". Reliez la propriété "Width" du rectangle avec la propriété spécifique à l'instance "wide". Dynamisez l'événement "OnClick" du bouton comme suit avec VBS :

```
'VBS308
Dim w
w = SmartTags("Properties\wide")
SmartTags("Properties\wide") = w + 50
```

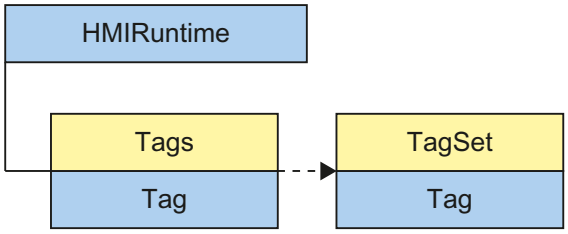
Si vous activez le runtime, la propriété spécifique à l'instance "wide" est augmentée de 50 à chaque clic sur le bouton. La largeur du rectangle est alors augmentée de 50.

Voir aussi

Propriété SmartTag (Page 553)

1.14.2.21 Objet Tag

Description



Un objet Tag (variable) est restitué via la liste "Tags". Vous pouvez accéder à toutes les propriétés et méthodes d'une variable via l'objet Tag.

Lors de la création d'un objet Tag, toutes les propriétés sont initialisées avec les valeurs suivantes :

- Value = VT_EMPTY
- Name = nom de variable
- QualityCode = BAD NON-SPECIFIC
- TimeStamp = 0

- LastError = 0
- ErrorDescription = " "

Remarque

Vous trouverez un récapitulatif des Codes qualité possibles en consultant WinCC Information System sous le mot repère "Communication" > "Diagnostic" ou "Communication" > "Codes qualité".

Utilisation

Ce que vous pouvez faire par le biais de l'objet "Tag" :

- Lire des informations sur la variable (Propriétés "Name", "QualityCode", "TimeStamp", "LastError" et "ErrorDescription")
- Attribuer une valeur à une variable (Méthode "Write", Propriété "Value")
- Lire la valeur d'une variable (Méthode "Read", Propriété "Value")

Lecture de la valeur d'une variable "Tag1" :

```
'VBS13
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read()
MsgBox objTag.Value
```

Déclaration de variables dans WinCC

Pour éviter des fautes d'orthographe dans les variables, définissez toujours les variables internes dans VB Script via l'instruction "Dim".

Lors de la création d'une nouvelle action, la commande "Option explicit" non supprimable s'inscrit automatiquement dans la zone de déclaration.

N'utilisez pas la commande "Option explicit" dans votre code, sinon il pourrait y avoir des erreurs d'exécution.

Exemple : Déclaration d'une variable VBScript "lngVar" :

```
'VBS14
Dim lngVar
lngVar = 5
MsgBox lngVar
```

Remarque

Les noms de variable ne doivent pas contenir de caractères spéciaux.

Notez que lors de la création d'une variable, cette dernière ne contient aucune valeur (Value = VT_EMPTY). Initialisez les variables après la déclaration avec une valeur correspondante.

Remarques sur CrossReference

Toutes les variables que vous adressez avec la formule standard

```
HMIRuntime.Tags("Tagname")
```

Sont automatiquement saisies par la CrossReference de WinCC pour être mentionnées dans les propriétés de la vue.

Si vous accédez à des variables avec d'autres formules dans votre code, vous pouvez les présenter à CrossReference via la section suivante :

```
' ' WINCC:TAGNAME_SECTION_START
Const TagNameInAction = "TagName"
' WINCC:TAGNAME_SECTION_END
```

Vous pouvez insérer cette section dans des actions VBS autant de fois que vous le voulez.

Remarque

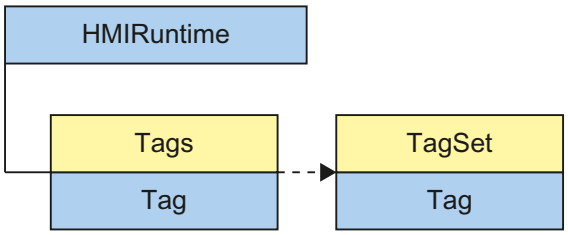
Il n'y a aucune garantie que CrossReference saisisse des noms de variable composés.

Voir aussi

Propriété Name (Page 482)
Exemple : Pour lire les valeurs variable (Page 799)
Exemple : Pour écrire les valeurs variable (Page 797)
Methode Write (Page 783)
Methode Read (Page 753)
Propriété Value (Page 652)
Propriété TimeStamp (Page 595)
Propriété QualityCode (Page 518)
Propriété LastError (Page 432)
Propriété ErrorDescription (Page 386)

1.14.2.22 Objet Tags (Liste)

Description



La liste "Tags" vous permet d'accéder aux variables dans WinCC Runtime. Un objet de type "Tag" est retransmis comme résultat après l'accès à la liste "Tags". Il est possible d'accéder à toutes les propriétés et méthodes d'une variable via l'objet Tag.

Remarque

"Tags" est une liste dont l'étendue des fonctions est restreint. Il n'est pas possible d'accéder via l'index aux variables au sein de la liste, mais uniquement via le nom de variable. Les méthodes standard get_Count et get_NewEnum ne peuvent être utilisées pour la liste Tags.

Utilisation

L'accès aux variables de la liste s'effectue de la façon suivante :

```
HMIRuntime.Tags("Tagname")
```

L'accès aux variables (objets Tag) en écriture et en lecture est déclaré via la liste Tags. Pour que l'accès en écriture et en lecture puisse être exécuté sans erreur, il faut que les variables correspondantes soient disponibles dans le stock de variables WinCC.

Dans VBS, vous pouvez accéder directement aux variables via leur nom et définir et lire des valeurs. Il vous faudra cependant toujours accéder aux variables par la liste Tags si vous voulez demander des propriétés supplémentaires pour les variables, tel le code de qualité ou l'horodatage. L'objet Tag retransmis vous permet d'accéder à toutes les propriétés et méthodes des variables. Vous devez constituer une instance pour l'objet, afin p.ex. d'écrire une variable binaire avec HMIRuntime.Tags("Variable").Value=TRUE.

La méthode "CreateTagSet" permet de créer un objet "TagSet" qui permet l'accès simultané à plusieurs variables.

Exemple

Deux possibilités se présentent lorsque vous créez des variables :

- Avec spécification du Serverprefix : pour les variables des systèmes multiposte qui ne sont pas inscrites localement.
- Utilisation directe du nom de variable : pour les variables qui sont inscrites localement, sur l'ordinateur.

Spécification du Serverprefix :

```
'VBS15
Dim objTag
Set objTag = HMIRuntime.Tags("Serverprefix::Tagname")
Lorsque vous spécifiez directement le Serverprefix, la propriété "ServerPrefix" se voit
attribuer la valeur correspondante.
```

Spécification du nom de variable

```
'VBS16
Dim objTag
Set objTag = HMIRuntime.Tags("Tagname")
Si vous utilisez uniquement le nom de variable, les propriétés "ServerPrefix" et
"TagPrefix" se voient attribuer les valeurs du contexte actuel (de la fenêtre de vue
actuelle).
```

Voir aussi

Exemple : Pour lire les valeurs variable (Page 799)

Exemple : Pour écrire les valeurs variable (Page 797)

Méthode Item (Page 740)

Méthode CreateTagSet (Page 687)

Objet Tag (Page 144)

1.14.2.23 Objet TagSet (Liste)

Description

L'objet "TagSet" permet par un seul appel d'accéder simultanément à plusieurs variables. Les performances sont alors améliorées et la charge de communication est moins lourde que lors de l'accès individuel aux diverses variables.

Utilisation

L'objet TagSet permet les opérations suivantes :

- Ajouter des variables à la liste (Méthode "Add").
- Accéder aux objets Tag de la liste et à leurs propriétés (Méthode "Item")
- Ecrire toutes les variables de la liste (Méthode "Write")
- Lire toutes les variables de la liste (Méthode "Read")
- Supprimer l'une ou l'autre variable de la liste (Méthode "Remove")
- Supprimer toutes les variables de la liste (Méthode "RemoveAll").

L'accès aux variables de la liste s'effectue de la façon suivante :

```
'VBS169
Dim myTags
myTags = HMIRuntime.Tags.CreateTagSet
myTags("Tagname")
```

Pour que l'accès aux variables de la liste (objets Tag) en lecture et en écriture puisse être exécuté correctement, il faut que les variables correspondantes soient disponibles dans le stock de variables WinCC.

Si une erreur se produit lors de l'accès en lecture ou en écriture, la méthode utilisée renvoie un message d'erreur par les propriétés "LastError" et "ErrorDescription".

L'écriture et la lecture synchrones de varaibles sont possibles. Si l'on positionne le paramètre optionnel "Writemode" sur 1, les variables de process sont écrites directement dans l'AS, p.ex. "group.Write 1". Si l'on positionne le paramètre optionnel "Readmode" sur 1, les variables du process ne sont pas connectées, mais directement lues dans l'AS ou dans le canal p.ex. "group.Read 1".

Exemple

L'exemple suivant montre comment créer un objet TagSet, ajouter des variables et écrire des valeurs.

```
'VBS168
Build a Reference to the TagSet Object
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
'Add Tags to the Collection
group.Add "Motor1"
group.Add "Motor2"
'Set the Values of the Tags
group("Motor1").Value = 3
group("Motor2").Value = 9
'Write the Values to the DataManager
group.Write
```

Voir aussi

Propriété LastError (Page 432)
Exemple : Pour lire les valeurs variable (Page 799)
Exemple : Pour écrire les valeurs variable (Page 797)
Methode Write (Page 783)
Méthode RemoveAll (Page 762)
Méthode Remove (Page 758)
Methode Read (Page 753)
Méthode Item (Page 740)
Propriété ErrorDescription (Page 386)
Propriété Count (Page 368)
Méthode Add (Page 684)
Objet Tags (Liste) (Page 147)
Objet Tag (Page 144)

1.14.3 Type d'objet de l'objet ScreenItem

1.14.3.1 Type d'objet de l'objet ScreenItem

Introduction

Vous trouverez ci-après tous les types disponibles pour l'objet "ScreenItem".

Les réglages de l'objet "ScreenItem" représentent tous les objets graphiques disponibles dans WinCC Graphics Designer.

Les types d'objet se répartissent selon la disposition dans Graphics Designer dans les groupes suivants:

- Objets standard
- Objets complexes
- Objets Windows
- Objets tuyaux
- Contrôles

comme c'était le cas jusqu'ici, les Types d'objets

- Objet utilisateur
- Groupe

Voir aussi

Objet ScreenItems (Liste) (Page 136)

Objet ScreenItem (Page 132)

Groupe (Page 291)

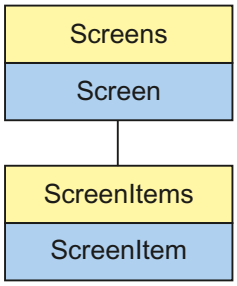
Objet utilisateur (Page 290)

Contrôles (Page 224)

1.14.3.2 Objets standard

Ellipse

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Ellipse"

Désignation du type sous VBS

HMIEllipse

Utilisation

Dans l'exemple suivant, l'objet appelé "Ellipse1" est décalé de 10 Pixel vers la droite :

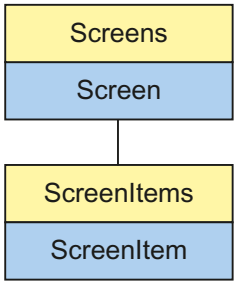
```
'VBS17
Dim objEllipse
Set objEllipse = ScreenItems("Ellipse1")
objEllipse.Left = objEllipse.Left + 10
```

Voir aussi

Propriété FillStyle (Page 394)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété RadiusWidth (Page 520)
Propriété RadiusHeight (Page 520)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOn (Page 316)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)
Propriété Layer (Page 433)

Arc d'ellipse

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Arc d'ellipse".

Désignation du type sous VBS

HMIEllipticalArc

Utilisation

Dans l'exemple suivant, l'objet appelé "EllipseArc1" est décalé de 10 Pixel vers la droite :

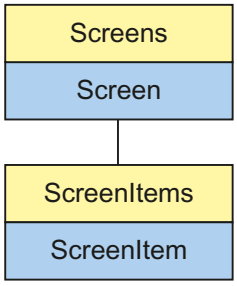
```
'VBS18
Dim objEllipseArc
Set objEllipseArc = ScreenItems("EllipseArc1")
objEllipseArc.Left = objEllipseArc.Left + 10
```

Voir aussi

Propriété RadiusHeight (Page 520)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété StartAngle (Page 559)
Propriété RadiusWidth (Page 520)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashBorderColor (Page 398)
Propriété EndAngle (Page 384)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété Layer (Page 433)

Segment d'ellipse

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Segment d'ellipse"

Désignation du type sous VBS

HMIEllipseSegment

Utilisation

Dans l'exemple suivant, l'objet appelé "EllipseSegment1" est décalé de 10 Pixel vers la droite :

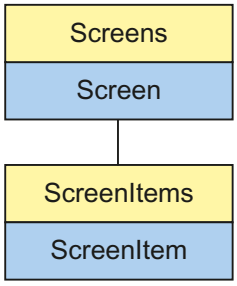
```
'VBS19
Dim objEllipseSeg
Set objEllipseSeg = ScreenItems("EllipseSegment1")
objEllipseSeg.Left = objEllipseSeg.Left + 10
```

Voir aussi

Objet Layer (Page 127)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété StartAngle (Page 559)
Propriété RadiusWidth (Page 520)
Propriété RadiusHeight (Page 520)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété EndAngle (Page 384)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOn (Page 316)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)
Propriété Layer (Page 433)

Cercle

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Cercle"

Désignation du type sous VBS

HMICircle

Utilisation

Dans l'exemple suivant, l'objet appelé "Circle1" est décalé de 10 Pixel vers la droite :

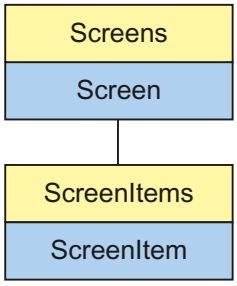
```
'VBS20
Dim objCircle
Set objCircle= ScreenItems("Circle1")
objCircle.Left = objCircle.Left + 10
```

Voir aussi

Propriétés (Page 292)
Propriété BorderStyle (Page 334)
Méthode Activate (Page 682)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Radius (Page 519)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOn (Page 316)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)
Propriété Layer (Page 433)

Arc de cercle

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Arc de cercle"

Désignation du type sous VBS

HMICircularArc

Utilisation

Dans l'exemple suivant, l'objet appelé "CircularArc1" est décalé de 10 Pixel vers la droite :

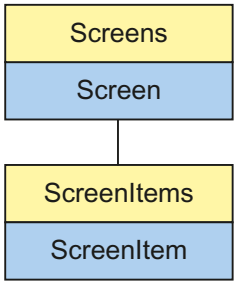
```
'VBS21
Dim objCircularArc
Set objCircularArc = ScreenItems("CircularArc1")
objCircularArc.Left = objCircularArc.Left + 10
```

Voir aussi

Propriété StartAngle (Page 559)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Radius (Page 519)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashBorderColor (Page 398)
Propriété EndAngle (Page 384)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété Layer (Page 433)

Segment de cercle

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Segment de cercle"

Désignation du type sous VBS

HMICircleSegment

Utilisation

Dans l'exemple suivant, l'objet appelé "PieSegment1" est décalé de 10 Pixel vers la droite :

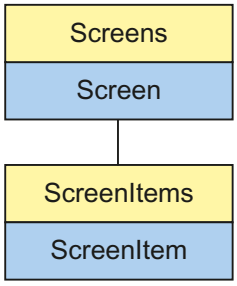
```
'VBS22
Dim objCircleSeg
Set objCircleSeg = ScreenItems("PieSegment1")
objCircleSeg.Left = objCircleSeg.Left + 10
```

Voir aussi

Propriété Type (Page 638)
Propriété BorderColor (Page 332)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété StartAngle (Page 559)
Propriété Radius (Page 519)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété EndAngle (Page 384)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOn (Page 316)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)
Propriété Layer (Page 433)

Trait

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Trait"

Désignation du type sous VBS

HMILine

Utilisation

Dans l'exemple suivant, l'objet appelé "Line1" est décalé de 10 Pixel vers la droite :

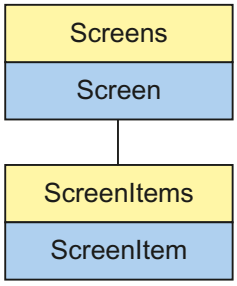
```
'VBS23
Dim objLine
Set objLine = ScreenItems("Line1")
objLine.Left = objLine.Left + 10
```

Voir aussi

Propriété PasswordLevel (Page 503)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété RotationAngle (Page 524)
Propriété ReferenceRotationTop (Page 522)
Propriété ReferenceRotationLeft (Page 521)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashBorderColor (Page 398)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderEndStyle (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété Layer (Page 433)

Polygone

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Polygone"

Désignation du type sous VBS

HMIPolygon

Utilisation

Dans l'exemple suivant, l'objet appelé "Polygon1" est décalé de 10 Pixel vers la droite :

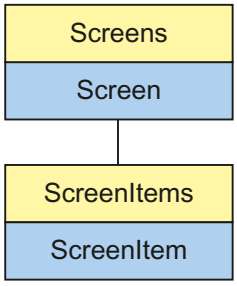
```
'VBS24
Dim objPolygon
Set objPolygon = ScreenItems("Polygon1")
objPolygon.Left = objPolygon.Left + 10
```

Voir aussi

Propriété ReferenceRotationTop (Page 522)
Propriété BackFlashColorOn (Page 316)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété RotationAngle (Page 524)
Propriété ReferenceRotationLeft (Page 521)
Propriété PointCount (Page 513)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Index (Page 425)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)
Propriété ActualPointTop (Page 297)
Propriété ActualPointLeft (Page 296)

Trait polygonal

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Trait polygonal"

Désignation du type sous VBS

HMIPolyLine

Utilisation

Dans l'exemple suivant, l'objet appelé "Polyline1" est décalé de 10 Pixel vers la droite :

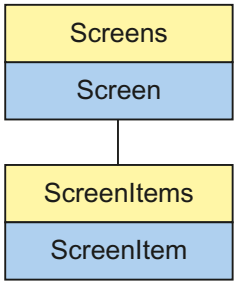
```
'VBS25
Dim objPolyline
Set objPolyline = ScreenItems("Polyline1")
objPolyline.Left = objPolyline.Left + 10
```

Voir aussi

Objet Layer (Page 127)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété RotationAngle (Page 524)
Propriété ReferenceRotationTop (Page 522)
Propriété ReferenceRotationLeft (Page 521)
Propriété PointCount (Page 513)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Propriété Index (Page 425)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashBorderColor (Page 398)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderEndStyle (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété ActualPointTop (Page 297)
Propriété ActualPointLeft (Page 296)
Propriété Layer (Page 433)

Rectangle

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Rectangle"

Désignation du type sous VBS

HMIRectangle

Utilisation

Dans l'exemple suivant, l'objet appelé "Rectangle1" est décalé de 10 Pixel vers la droite :

```
'VBS26
Dim objRectangle
Set objRectangle = ScreenItems("Rectangle1")
objRectangle.Left = objRectangle.Left + 10
```

Remarques sur le traitement des erreurs

Rectangle et rectangle arrondi sont représentés dans le modèle d'objet sur un seul type "HMIRectangle". Etant donné que ces deux objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime) via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

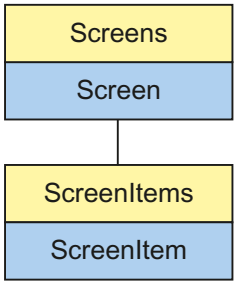
```
Sub OnClick(ByVal Item)
'VBS27
Dim objScreenItem
'
'Activation of errorhandling:
On Error Resume Next
For Each objScreenItem In ScreenItems
If "HMIRectangle" = objScreenItem.Type Then
'
'=== Property "RoundCornerHeight" only available for RoundRectangle
objScreenItem.RoundCornerHeight = objScreenItem.RoundCornerHeight * 2
If 0 <> Err.Number Then
HMIRuntime.Trace objScreenItem.Name & ": no RoundedRectangle" & vbCrLf
'
'Delete error message
    Err.Clear
    End If
    End If
Next
On Error Goto 0  'Deactivation of errorhandling
End Sub
```

Voir aussi

Propriétés (Page 292)
Propriété BorderFlashColorOn (Page 334)
Méthode Activate (Page 682)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOn (Page 316)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)
Propriété Layer (Page 433)

Rectangle arrondi

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Rectangle arrondi".

Désignation du type sous VBS

HMIRoundRectangle

Utilisation

Dans l'exemple suivant, l'objet appelé "RoundedRectangle1" est décalé de 10 Pixel vers la droite :

```
'VBS28
Dim objRoundedRectangle
Set objRoundedRectangle = ScreenItems("RoundedRectangle1")
objRoundedRectangle.Left = objRoundedRectangle.Left + 10
```

Remarques sur le traitement des erreurs

Rectangle et rectangle arrondi sont représentés dans le modèle d'objet sur un seul type "HMIRectangle". Etant donné que ces deux objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime) via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

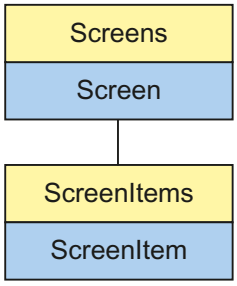
```
Sub OnClick(ByVal Item)
'VBS29
Dim objScreenItem
On Error Resume Next    'Activation of errorhandling
For Each objScreenItem In ScreenItems
If "HMIRectangle" = objScreenItem.Type Then
'
'=== Property "RoundCornerHeight" available only for RoundRectangle
objScreenItem.RoundCornerHeight = objScreenItem.RoundCornerHeight * 2
If 0 <> Err.Number Then
HMIRuntime.Trace objScreenItem.ObjectName & ": no RoundedRectangle" & vbCrLf
Err.Clear    'Delete errormessage
    End If
    End If
Next
On Error Goto 0  'Deactivation of errorhandling
End Sub
```

Voir aussi

Propriété FlashBackColor (Page 398)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété RoundCornerWidth (Page 525)
Propriété RoundCornerHeight (Page 525)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOn (Page 316)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)
Propriété Layer (Page 433)

Texte statique

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Texte statique"

Désignation du type sous VBS

HMITextField

Utilisation

Dans l'exemple suivant, l'objet appelé "StaticText1" est décalé de 10 Pixel vers la droite :

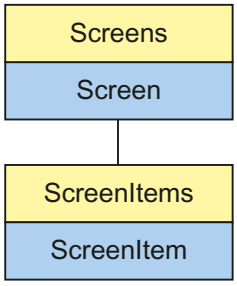
```
'VBS30
Dim objStaticText
Set objStaticText = ScreenItems("StaticText1")
objStaticText.Left = objStaticText.Left + 10
```

Voir aussi

Propriété ObjectName (Page 485)
Propriété BorderFlashColorOn (Page 334)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Liste de textes (Page 206)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété Orientation (Page 499)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété ForeFlashColorOn (Page 409)
Propriété ForeFlashColorOff (Page 409)
Propriété ForeColor (Page 408)
Propriété FontUnderline (Page 407)
Propriété FontSize (Page 407)
Propriété FontName (Page 406)
Propriété FontItalic (Page 405)
Propriété FontBold (Page 405)
Propriété FlashRateForeColor (Page 401)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashForeColor (Page 398)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOff (Page 333)

Connecteur

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Connecteur"

Désignation du type sous VBS

HMIConnector

Utilisation

Dans l'exemple suivant, l'objet appelé "Connector1" est décalé de 10 Pixel vers la droite :

```
'VBS31
Dim objConnector
Set objConnector = ScreenItems("Connector1")
objConnector.Left = objConnector.Left + 10
```

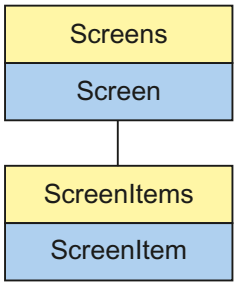
Voir aussi

Objet ScreenItems (Liste) (Page 136)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété TopConnectedObjectName (Page 615)
Propriété TopConnectedConnectionPointIndex (Page 615)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Parent (Page 501)
Propriété Orientation (Page 499)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Propriété Layer (Page 433)
Propriété Height (Page 417)
Propriété Enabled (Page 382)
Propriété BottomConnectedObjectName (Page 335)
Propriété BottomConnectedConnectionPointIndex (Page 335)

1.14.3.3 Objets complexes

Bargraphes tridimensionnels

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Bargraphes tridimensionnels"

Désignation du type sous VBS

HMIBar

Utilisation

Dans l'exemple suivant, l'objet appelé "3DBar1" est décalé de 10 Pixel vers la droite :

```
'VBS32
Dim objBar
Set objBar = ScreenItems("3DBar1")
objBar.Left = objBar.Left + 10
```

Remarques sur le traitement des erreurs

Les bargraphes et bargraphes tridimensionnels sont représentés dans le modèle d'objet sur un seul type "HMIBar". Etant donné que ces deux objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime) via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

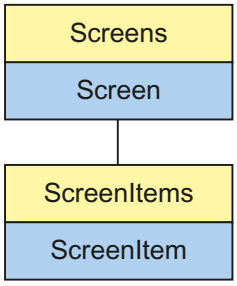
```
'VBS148
Sub OnClick(ByVal Item)
Dim objScreenItem
'
'Activation of errorhandling:
On Error Resume Next
For Each objScreenItem In ScreenItems
If "HMIBar" = objScreenItem.Type Then
'
'=== Property "Layer00Value" only available for 3D bar
objScreenItem.Layer00Value = objScreenItem.Layer00Value * 2
If 0 <> Err.Number Then
HMIRuntime.Trace objScreenItem.Name & ": no 3D bar" & vbCrLf
'
'Delete error message
    Err.Clear
    End If
    End If
Next
On Error Goto 0  'Deactivation of errorhandling
End Sub
```

Voir aussi

Propriété Type (Page 638)
Propriété Layer08Color (Page 440)
Propriété BorderStyle (Page 334)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété ZeroPointValue (Page 679)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Process (Page 516)
Propriété PredefinedAngles (Page 515)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Min (Page 479)
Propriété Max (Page 461)
Propriété LightEffect (Page 450)
Propriété Left (Page 449)
Propriété Layer10Value (Page 448)
Propriété Layer09Value (Page 448)
Propriété Layer08Value (Page 448)
Propriété Layer07Value (Page 447)
Propriété Layer06Value (Page 447)
Propriété Layer05Value (Page 447)
Propriété Layer04Value (Page 447)
Propriété Layer03Value (Page 446)
Propriété Layer02Value (Page 446)
Propriété Layer01Value (Page 446)
Propriété Layer00Value (Page 445)
Propriété Layer10Color (Page 440)
Propriété Layer09Color (Page 440)
Propriété Layer07Color (Page 439)
Propriété Layer06Color (Page 439)
Propriété Layer05Color (Page 439)
Propriété Layer04Color (Page 438)
Propriété Layer03Color (Page 438)
Propriété Layer02Color (Page 438)

Fenêtre d'application

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Fenêtre d'application"

Désignation du type sous VBS

HMIApplicationWindow

Utilisation

Dans l'exemple suivant, l'objet appelé "ApplicationWindow1" est décalé de 10 Pixel vers la droite :

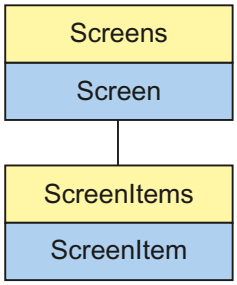
```
'VBS33
Dim objAppWindow
Set objAppWindow = ScreenItems("ApplicationWindow1")
objAppWindow.Left = objAppWindow.Left + 10
```

Voir aussi

Propriétés (Page 292)
Méthode Activate (Page 682)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété WindowBorder (Page 670)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété Template (Page 570)
Propriété Parent (Page 501)
Propriété OnTop (Page 489)
Propriété ObjectName (Page 485)
Propriété Moveable (Page 480)
Propriété MaximizeButton (Page 461)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété Enabled (Page 382)
Propriété CloseButton (Page 347)
Propriété Caption (Page 339)
Propriété Application (Page 304)

Bargraphe

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Bargraphe"

Désignation du type sous VBS

HMIBar

Utilisation

Dans l'exemple suivant, l'objet appelé "Bar1" est décalé de 10 Pixel vers la droite :

```
'VBS34
Dim objBar
Set objBar = ScreenItems("Bar1")
objBar.Left = objBar.Left + 10
```

Remarques sur le traitement des erreurs

Les bargraphes et bargraphes tridimensionnels sont représentés dans le modèle d'objet sur un seul type "HMIBar". Etant donné que ces deux objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime) via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

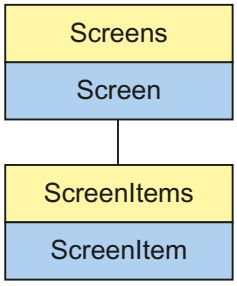
```
'VBS147
Sub OnClick(ByVal Item)
Dim objScreenItem
'
'Activation of errorhandling:
On Error Resume Next
For Each objScreenItem In ScreenItems
If "HMIBar" = objScreenItem.Type Then
'
'=== Property "LimitHigh4" only available for bar
objScreenItem.LimitHigh4 = objScreenItem.LimitHigh4 * 2
If 0 <> Err.Number Then
HMIRuntime.Trace objScreenItem.Name & ": no bar" & vbCrLf
'
'Delete error message
    Err.Clear
    End If
    End If
Next
On Error Goto 0  'Deactivation of errorhandling
End Sub
```

Voir aussi

Propriété ToolTipText (Page 613)
Objet Layer (Page 127)
Propriété ColorChangeType (Page 351)
Propriété Average (Page 311)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété ZeroPointValue (Page 679)
Propriété ZeroPoint (Page 679)
Propriété Width (Page 669)
Propriété WarningLow (Page 669)
Propriété WarningHigh (Page 668)
Propriété Visible (Page 667)
Propriété TypeWarningLow (Page 643)
Propriété TypeWarningHigh (Page 642)
Propriété TypeToleranceLow (Page 642)
Propriété TypeToleranceHigh (Page 642)
Propriété TypeLimitLow5 (Page 642)
Propriété TypeLimitLow4 (Page 641)
Propriété TypeLimitHigh5 (Page 641)
Propriété TypeLimitHigh4 (Page 641)
Propriété TypeAlarmLow (Page 640)
Propriété TypeAlarmHigh (Page 640)
Propriété Type (Page 638)
Propriété TrendColor (avant WinCC V7) (Page 619)
Propriété Trend (Page 616)
Propriété Top (Page 614)
Propriété ToleranceLow (Page 601)
Propriété ToleranceHigh (Page 600)
Propriété ScalingType (Page 531)
Propriété Scaling (Page 531)
Propriété ScaleTicks (Page 530)
Propriété ScaleColor (Page 530)
Propriété RightComma (Page 523)
Propriété Process (Page 516)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Min (Page 479)

Fenêtre de vue

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Fenêtre de vue"

Désignation du type sous VBS

HMIScreenWindow

Utilisation

Dans l'exemple suivant, l'objet appelé "ScreenWindow1" est décalé de 10 Pixel vers la droite :

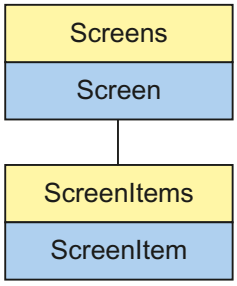
```
'VBS35
Dim objScrWindow
Set objScrWindow = ScreenItems("ScreenWindow1")
objScrWindow.Left = objScrWindow.Left + 10
```

Voir aussi

Propriété ServerPrefix (Page 544)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Zoom (Page 680)
Propriété WindowBorder (Page 670)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété UpdateCycle (Page 645)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété TagPrefix (Page 568)
Propriété ScrollPositionY (Page 535)
Propriété ScrollPositionX (Page 534)
Propriété ScrollBars (Page 534)
Propriété ScreenName (Page 533)
Propriété Screens (Page 533)
Propriété Parent (Page 501)
Propriété OnTop (Page 489)
Propriété OffsetTop (Page 488)
Propriété OffsetLeft (Page 487)
Propriété ObjectName (Page 485)
Propriété Moveable (Page 480)
Propriété MaximizeButton (Page 461)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété Enabled (Page 382)
Propriété CloseButton (Page 347)
Propriété CaptionText (Page 341)
Propriété Caption (Page 339)
Propriété AdaptSize (Page 298)
Propriété AdaptPicture (Page 298)
Propriété-MenuToolBarConfig (Page 468)

Contrôle

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Control"

Le type d'objet Control adopte toujours les propriétés du type de contrôle que vous avez choisi. Dans le cas des Controls mis à disposition par WinCC, veuillez consulter les propriétés dans la description des Controls respectifs.

Dans le cas des Controls d'une autre marque, les propriétés y sont intégrées et ne font donc pas partie de cette description. Vous pouvez cependant sortir les propriétés Control via les propriétés "Item".

Désignation du type sous VBS

Désignation du type spéciale WinCC ou ProgID indépendante de la version

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 10 Pixel vers la droite :

```
'VBS36
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left + 10
```

Particularité

Dans le cas des Controls mis à disposition par WinCC, une identification spéciale est retransmise comme type. Elle se trouve sous la rubrique "Identification du type dans VBS" dans les descriptions individuelles de WinCC Controls.

Utilisation de Controls d'une autre marque

Pour les Controls WinCC d'une autre marque, c'est l'identification ProgID indépendante de la version qui est retransmise comme type.

A partir de ProgID, vous pouvez déterminer l'identification ProgID dépendant de la version ou le "User friendly Name" : Dans l'exemple suivant, "Control1" est un Control incorporé dans la

vue et qui retransmet déjà l'identification ProgID indépendante de la version, via la propriété Type.

Remarque

Etant donné que ce ne sont pas tous les Controls qui ont des ProgID dépendant de la version, il vous faut intégrer une fonction de traitement des erreurs pour sortir l'identification ProgID dépendant de la version ou le UserFriendlyName. Si une fonction de traitement des erreurs est utilisée, le code se termine immédiatement sans résultat si aucune ProgID n'est trouvée.

Comment déterminer l'identification ProgID dépendant de la version :

```
'VBS37
Dim objControl
Dim strCurrentVersion
Set objControl = ScreenItems("Control1")
strCurrentVersion = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type &
"\CurVer\")
MsgBox strCurrentVersion
```

Remarque

Pour que l'exemple donné ci-dessus fonctionne, il vous faut intégrer un contrôle multimédia dans la vue.

Déterminez comme suit le UserFriendlyName :

```
'VBS38
Dim objControl
Dim strFriendlyName
Set objControl = ScreenItems("Control1")
strFriendlyName = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type & "\")
MsgBox strFriendlyName
```

Remarque

Pour que l'exemple donné ci-dessus fonctionne, il vous faut intégrer un contrôle multimédia dans la vue.

Si vous utilisez un Control n'ayant pas été mis à disposition par WinCC, il se peut que les propriétés apportées dans ce Control portent les mêmes noms que les propriétés ScreenItem générales. Dans ce cas, ce sont les propriétés ScreenItem qui ont la priorité. Vous pouvez accéder aux propriétés "cachées" d'un Control d'une autre marque par le biais de la propriété supplémentaire "object". Vous accédez donc par exemple aux propriétés d'un Control d'une autre marque sous la forme suivante :

```
Control.object.type
```

En cas d'identité de nom, les propriétés de l'objet ScreenItem sont utilisées si vous recourez à la forme suivante :

```
Control.type
```

WinCC Controls disponibles

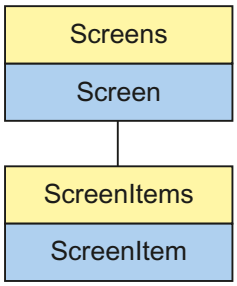
- WinCC Alarm Control
- WinCC Digital/Analog Clock
- WinCC FunctionTrendControl
- WinCC Gauge Control
- WinCC Media Control
- WinCC OnlineTableControl
- WinCC OnlineTrendControl
- WinCC Push Button Control
- WinCC Slider Control
- WinCC UserArchiveControl
- HMI Symbol Library

Voir aussi

Propriété Object (Page 484)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Propriété Layer (Page 433)
Propriété Height (Page 417)
Propriété Enabled (Page 382)

Champ d'E/S

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Champ E/S"

Désignation du type sous VBS

HMIIOField

Utilisation

Dans l'exemple suivant, l'objet appelé "IOField1" est décalé de 10 Pixel vers la droite :

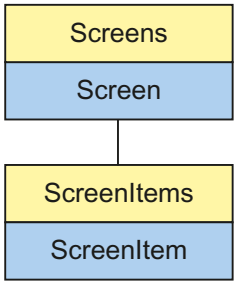
```
'VBS39
Dim objIOField
Set objIOField = ScreenItems("IOField1")
objIOField.Left = objIOField.Left + 10
```

Voir aussi

Propriété OperationMessage (Page 490)
Propriété EditAtOnce (Page 382)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété OutputValue (Page 500)
Propriété OutputFormat (Page 500)
Propriété Orientation (Page 499)
Propriété OperationReport (Page 498)
Propriété ObjectName (Page 485)
Propriété LimitMin (Page 452)
Propriété LimitMax (Page 452)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété HiddenInput (Page 418)
Propriété Height (Page 417)
Propriété ForeFlashColorOn (Page 409)
Propriété ForeFlashColorOff (Page 409)
Propriété ForeColor (Page 408)
Propriété FontUnderline (Page 407)
Propriété FontSize (Page 407)
Propriété FontName (Page 406)
Propriété FontItalic (Page 405)
Propriété FontBold (Page 405)
Propriété FlashRateForeColor (Page 401)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashForeColor (Page 398)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillColor (Page 392)

Instance de bloc d'affichage

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Instance de bloc d'affichage".

Désignation du type sous VBS

HMIFaceplateObject

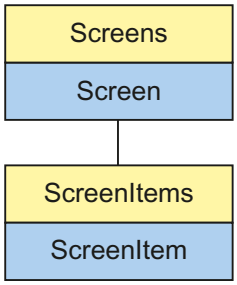
Utilisation

Dans l'exemple suivant, l'objet appelé ""FaceplateInstance1" est décalé de 10 pixels vers la droite :

```
'VBS309
Dim objFaceplateObject
Set objFaceplateObject = ScreenItems("FaceplateInstanz1")
objFaceplateObject.Left = objFaceplateObject.Left + 10
```

Objet graphique

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Objet graphique"

Désignation du type sous VBS

HMIGraphicView

Utilisation

Dans l'exemple suivant, l'objet appelé "GraphicObject1" est décalé de 10 Pixel vers la droite :

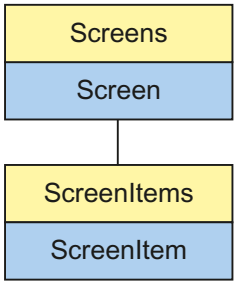
```
'VBS40
Dim objGraphicView
Set objGraphicView= ScreenItems("GraphicObject1")
objGraphicView.Left = objGraphicView.Left + 10
```

Voir aussi

Propriété Parent (Page 501)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété PicUseTransColor (Page 513)
Propriété PictureName (Page 510)
Propriété PicTransColor (Page 509)
Propriété PicReferenced (Page 508)
Propriété PasswordLevel (Page 503)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BackFlashColorOn (Page 316)
Propriété BackFlashColorOff (Page 315)
Propriété BackColor (Page 313)

Champ de combinaison

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Champ de combinaison".

Désignation du type sous VBS

HMIComboBox

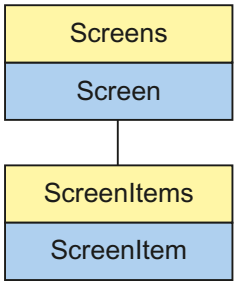
Utilisation

Dans l'exemple suivant, l'objet portant le nom "ComboBox1" est déplacé sur la droite de 10 pixels :

```
'VBS21
Dim objComboBox
Set objComboBox = ScreenItems("ComboBox1")
objComboBox.Left = objComboBox.Left + 10
```

Champ de liste

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Champ de liste".

Désignation du type sous VBS

HMIListBox

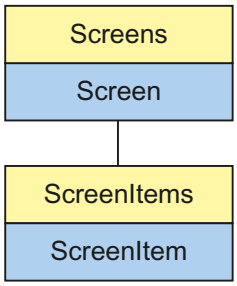
Utilisation

Dans l'exemple suivant, l'objet portant le nom "ListBox1" est déplacé sur la droite de 10 pixels :

```
'VBS21
Dim objListBox
Set objListBox = ScreenItems("ListBox1")
objListBox.Left = objListBox.Left + 10
```

Texte multiligne

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Texte multiligne".

Désignation du type sous VBS

HMIMultiLineEdit

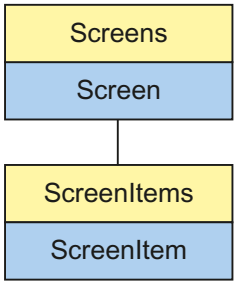
Utilisation

Dans l'exemple suivant, l'objet portant le nom "MultiLineEdit1" est déplacé sur la droite de 10 pixels :

```
'VBS21
Dim objMultiLineEdit
Set objMultiLineEdit = ScreenItems("MultiLineEdit1")
objMultiLineEdit.Left = objMultiLineEdit.Left + 10
```

Objet OLE

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Objet OLE" La valeur de retour est de type STRING.

Désignation du type sous VBS

ProgID indépendante de la version

Utilisation

Dans l'exemple suivant, l'objet appelé "OLEElement1" est décalé de 10 Pixel vers la droite :

```
'VBS41
Dim objOLEElement
Set objOLEElement = ScreenItems("OLEElement1")
objOLEElement.Left = objOLEElement.Left + 10
```

Particularité

Pour les objets OLE, c'est l'identification ProgID indépendante de la version qui est retransmise comme type.

A partir de ProgID, vous pouvez déterminer l'identification ProgID dépendant de la version ou le "User friendly Name": Dans l'exemple suivant, "OLEObject1" est un Control incorporé dans la vue et qui retransmet déjà l'identification ProgID indépendante de la version, via la propriété Type.

Remarque

Etant donné que ce ne sont pas tous les Controls qui ont des ProgID dépendant de la version, il vous faut intégrer une fonction de traitement des erreurs pour sortir l'identification ProgID dépendant de la version ou le UserFriendlyName. Si une fonction de traitement des erreurs est utilisée, le code se termine immédiatement sans résultat si aucune ProgID n'est trouvée.

Comment déterminer l'identification ProgID dépendant de la version:

```
'VBS42
Dim objControl
Dim strCurrentVersion
Set objControl = ScreenItems("OLEElement1")
strCurrentVersion = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type &
"\CurVer\")
MsgBox strCurrentVersion
```

Remarque

Pour que l'exemple ci-dessus fonctionne, il vous faut incorporer un document Word dans la vue en tant qu'objet OLE.

Déterminez le User friendly Name comme suit:

```
'VBS43
Dim objControl
Dim strFriendlyName
Set objControl = ScreenItems("OLEElement1")
strFriendlyName = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type & "\")
MsgBox strFriendlyName
```

Remarque

Pour que l'exemple ci-dessus fonctionne, il vous faut incorporer un document Word dans la vue en tant qu'objet OLE.

Utilisation des objets OLE

Si vous utilisez un objet OLE, il se peut que les propriétés apportées par cet objet OLE portent les mêmes noms que les propriétés ScreenItem générales. Dans ce cas, ce sont les propriétés ScreenItem qui ont priorité. Vous pouvez accéder aux propriétés "cachées" d'un objet OLE par le biais de la propriété supplémentaire "object". Vous accédez donc par exemple aux propriétés d'un objet OLE sous la forme suivante :

```
OLEObjekt.object.type
```

Si vous utilisez uniquement la forme

```
OLEObjet.type
```

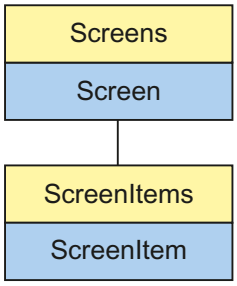
Ce sont les propriétés de l'objet ScreenItem qui seront utilisées si les noms sont identiques.

Voir aussi

Propriété Height (Page 417)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Parent (Page 501)
Propriété Object (Page 484)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Propriété Layer (Page 433)
Propriété Enabled (Page 382)

Signalisation groupée

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Signalisation groupée"

Désignation du type sous VBS

HMIGroupDisplay

Utilisation

Dans l'exemple suivant, l'objet appelé "GroupDisplay1" est décalé de 10 Pixel vers la droite :

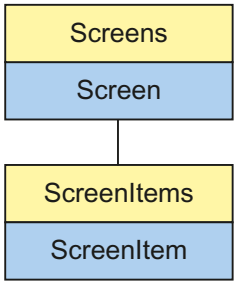
```
'VBS44
Dim objGroupDisplay
Set objGroupDisplay = ScreenItems("GroupDisplay1")
objGroupDisplay.Left = objGroupDisplay.Left + 10
```

Voir aussi

Méthode Activate (Page 682)
Propriété MCKQBackColorOn (Page 465)
Propriété FontBold (Page 405)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété UserValue4 (Page 650)
Propriété UserValue3 (Page 649)
Propriété UserValue2 (Page 649)
Propriété UserValue1 (Page 649)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété SignificantMask (Page 552)
Propriété SameSize (Page 530)
Propriété Relevant (Page 522)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété MessageClass (Page 476)
Propriété MCText (Page 467)
Propriété MCKQTextFlash (Page 467)
Propriété MCKQTextColorOn (Page 466)
Propriété MCKQTextColorOff (Page 466)
Propriété MCKQBackFlash (Page 466)
Propriété MCKQBackColorOff (Page 465)
Propriété MCKOTextFlash (Page 465)
Propriété MCKOTextColorOn (Page 465)
Propriété MCKOTextColorOff (Page 464)
Propriété MCKOBackFlash (Page 464)
Propriété MCKOBackColorOn (Page 464)
Propriété MCKOBackColorOff (Page 463)
Propriété MCGUTextFlash (Page 463)
Propriété MCGUTextColorOn (Page 463)
Propriété MCGUTextColorOff (Page 463)
Propriété MCGUBackFlash (Page 462)
Propriété MCGUBackColorOn (Page 462)
Propriété MCGUBackColorOff (Page 462)

Liste de textes

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Liste d'objets"

Désignation du type sous VBS

HMISymbolicIOField

Utilisation

Dans l'exemple suivant, l'objet appelé "TextList1" est décalé de 10 Pixel vers la droite :

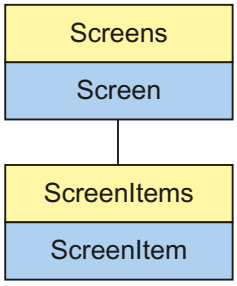
```
'VBS45
Dim objSymIO
Set objSymIO = ScreenItems("TextList1")
objSymIO.Left = objSymIO.Left + 10
```

Voir aussi

Propriété Type (Page 638)
Propriété FontUnderline (Page 407)
Propriété BackFlashColorOff (Page 315)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété UnselTextColor (Page 645)
Propriété UnselBGColor (Page 644)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété SelTextColor (Page 541)
Propriété SelBGColor (Page 536)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété OutputValue (Page 500)
Propriété Orientation (Page 499)
Propriété OperationReport (Page 498)
Propriété OperationMessage (Page 490)
Propriété ObjectName (Page 485)
Propriété NumberLines (Page 483)
Propriété ListType (Page 454)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété LanguageSwitch (Page 431)
Propriété ItemBorderWidth (Page 428)
Propriété ItemBorderStyle (Page 428)
Propriété ItemBorderColor (Page 428)
Propriété ItemBorderBackColor (Page 427)
Propriété Height (Page 417)
Propriété ForeFlashColorOn (Page 409)
Propriété ForeFlashColorOff (Page 409)
Propriété ForeColor (Page 408)
Propriété FontSize (Page 407)
Propriété FontName (Page 406)
Propriété FontItalic (Page 405)
Propriété FontBold (Page 405)
Propriété FlashRateForeColor (Page 401)

Indicateur d'état

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Indicateur d'état"

Désignation du type sous VBS

HMIGraphicIOField

Utilisation

Dans l'exemple suivant, l'objet appelé "StatusDisplay1" est décalé de 10 Pixel vers la droite :

```
'VBS46
Dim objGraphicIO
Set objGraphicIO= ScreenItems("StatusDisplay1")
objGraphicIO.Left = objGraphicIO.Left + 10
```

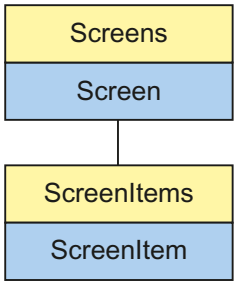
Voir aussi

Objet Layer (Page 127)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Propriété Index (Page 425)
Propriété Height (Page 417)
Propriété FlashRateFlashPic (Page 401)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashPicUseTransColor (Page 400)
Propriété FlashPicture (Page 399)
Propriété FlashPicTransColor (Page 399)
Propriété FlashPicReferenced (Page 399)
Propriété FlashFlashPicture (Page 398)
Propriété FlashBorderColor (Page 398)
Propriété Enabled (Page 382)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)
Propriété BorderFlashColorOff (Page 333)
Propriété BorderColor (Page 332)
Propriété BorderBackColor (Page 332)
Propriété BasePicUseTransColor (Page 320)
Propriété BasePicture (Page 320)
Propriété BasePicTransColor (Page 319)
Propriété BasePicReferenced (Page 319)

1.14.3.4 Objets Windows

Bouton

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Bouton"

Désignation du type sous VBS

HMIButton

Utilisation

Dans l'exemple suivant, l'objet appelé "Button1" est décalé de 10 Pixel vers la droite :

```
'VBS47
Dim cmdButton
Set cmdButton = ScreenItems("Button1")
cmdButton.Left = cmdButton.Left + 10
```

Remarques sur le traitement des erreurs

Les boutons et boutons-poussoirs sont représentés dans le modèle d'objet sur un seul type "HMIButton". Etant donné que ces objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime) via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

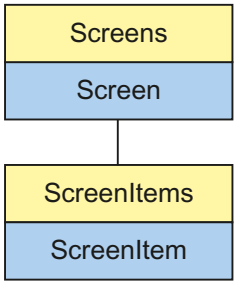
```
Sub OnClick(ByVal Item)
'VBS48
Dim objScreenItem
On Error Resume Next    'Activation of errorhandling
For Each objScreenItem In ScreenItems
If objScreenItem.Type = "HMIButton" Then
'
'=== Property "Text" available only for Standard-Button
objScreenItem.Text = "Windows"
If 0 <> Err.Number Then
HMIRuntime.Trace objScreenItem.ObjectName & ": no Windows-Button" & vbCrLf
Err.Clear    'Delete error message
    End If
    End If
Next
On Error Goto 0  'Deactivation of errorhandling
End Sub
```

Voir aussi

Propriété Top (Page 614)
Propriété FlashBorderColor (Page 398)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété WindowsStyle (Page 671)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété ToolTipText (Page 613)
Liste de textes (Page 206)
Propriété PictureUp (Page 511)
Propriété PictureDown (Page 510)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété Orientation (Page 499)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Hotkey (Page 422)
Propriété Height (Page 417)
Propriété ForeFlashColorOn (Page 409)
Propriété ForeFlashColorOff (Page 409)
Propriété ForeColor (Page 408)
Propriété FontUnderline (Page 407)
Propriété FontSize (Page 407)
Propriété FontName (Page 406)
Propriété FontItalic (Page 405)
Propriété FontBold (Page 405)
Propriété FlashRateForeColor (Page 401)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashForeColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)

Case à cocher

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Case à cocher"

Désignation du type sous VBS

HMICheckBox

Utilisation

Dans l'exemple suivant, l'objet appelé "CheckBox1" est décalé de 10 Pixel vers la droite :

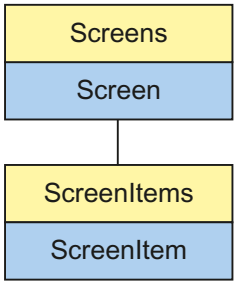
```
'VBS49
Dim chkCheckBox
Set chkCheckBox = ScreenItems("CheckBox1")
chkCheckBox.Left = chkCheckBox.Left + 10
```

Voir aussi

Propriété FontSize (Page 407)
Propriété BackColor (Page 313)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Liste de textes (Page 206)
Propriété Process (Page 516)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété Orientation (Page 499)
Propriété OperationMessage (Page 490)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Index (Page 425)
Propriété Height (Page 417)
Propriété ForeFlashColorOn (Page 409)
Propriété ForeFlashColorOff (Page 409)
Propriété ForeColor (Page 408)
Propriété FontUnderline (Page 407)
Propriété FontName (Page 406)
Propriété FontItalic (Page 405)
Propriété FontBold (Page 405)
Propriété FlashRateForeColor (Page 401)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashForeColor (Page 398)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)

Case d'option

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Case d'option"

Désignation du type sous VBS

HMIOptionGroup

Utilisation

Dans l'exemple suivant, l'objet appelé "RadioBox1" est décalé de 10 Pixel vers la droite :

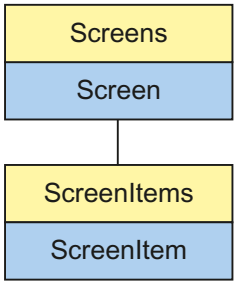
```
'VBS50
Dim objOptionGroup
Set objOptionGroup = ScreenItems("RadioBox1")
objOptionGroup.Left = objOptionGroup.Left + 10
```

Voir aussi

Propriété ForeColor (Page 408)
Propriété BackFlashColorOn (Page 316)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Liste de textes (Page 206)
Propriété Process (Page 516)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété Orientation (Page 499)
Propriété OperationMessage (Page 490)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Index (Page 425)
Propriété Height (Page 417)
Propriété ForeFlashColorOn (Page 409)
Propriété ForeFlashColorOff (Page 409)
Propriété FontUnderline (Page 407)
Propriété FontSize (Page 407)
Propriété FontName (Page 406)
Propriété FontItalic (Page 405)
Propriété FontBold (Page 405)
Propriété FlashRateForeColor (Page 401)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashForeColor (Page 398)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)

Bouton rond

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Bouton rond"

Désignation du type sous VBS

HMISwitch

Utilisation

Dans l'exemple suivant, l'objet appelé "RoundButton1" est décalé de 10 Pixel vers la droite :

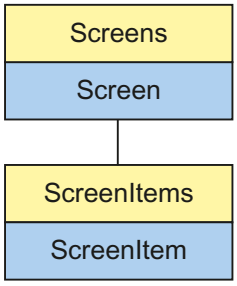
```
'VBS51
Dim objSwitch
Set objSwitch= ScreenItems("RoundButton1")
objSwitch.Left = objSwitch.Left + 10
```

Voir aussi

Propriété PicDownUseTransColor (Page 508)
Propriété BorderColorTop (Page 333)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Toggle (Page 600)
Propriété Radius (Page 519)
Propriété Pressed (Page 515)
Propriété PicUpUseTransColor (Page 512)
Propriété PicUpTransparent (Page 512)
Propriété PicUpReferenced (Page 512)
Propriété PictureUp (Page 511)
Propriété PictureDown (Page 510)
Propriété PictureDeactivated (Page 509)
Propriété PicDownTransparent (Page 508)
Propriété PicDownReferenced (Page 507)
Propriété PicDeactUseTransColor (Page 507)
Propriété PicDeactTransparent (Page 507)
Propriété PicDeactReferenced (Page 506)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété Enabled (Page 382)

Slider

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Slider"

Désignation du type sous VBS

HMISlider

Utilisation

Dans l'exemple suivant, l'objet appelé "Slider1" est décalé de 10 Pixel vers la droite :

```
'VBS53
Dim sldSlider
Set sldSlider = ScreenItems("Slider1")
sldSlider.Left = sldSlider.Left + 10
```

Remarques sur le traitement des erreurs

Slider et WinCC Slider Controls sont représentés dans un modèle d'objet sur un seul type "HMISlider". Etant donné que ces objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime) via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

```
Sub OnClick(Byval Item)
'VBS194
Dim ScreenItem
' activating error handling:
On Error Resume Next
For Each ScreenItem In ScreenItems
If ScreenItem.Type = "HMISlider" Then
'=== Property "BevelColorUp" only exists for a WinCC Slider Control
ScreenItem.BevelColorUp = 1
If (Err.Number <> 0) Then
HMIRuntime.Trace(ScreenItem.ObjectName + ": no Windows-Slider" + vbCrlf)
' delete error message
    Err.Clear
    End If
'=== Property "BorderStyle" only exists for a Windows-Slider
ScreenItem.BorderStyle = 1
If (Err.Number <> 0) Then
HMIRuntime.Trace(ScreenItem.ObjectName + ": no WinCC Slider Control" + vbCrlf)
    Err.Clear
    End If
    End If
Next
On Error GoTo 0  ' deactivating error handling
End Sub
```

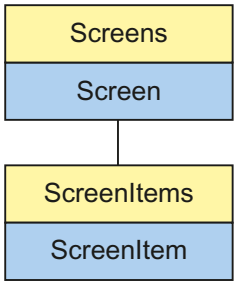
Voir aussi

Propriété Height (Page 417)
Propriété BackColorBottom (Page 315)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Propriété WindowsStyle (Page 671)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété SmallChange (Page 553)
Propriété Process (Page 516)
Propriété PasswordLevel (Page 503)
Propriété Parent (Page 501)
Propriété OperationReport (Page 498)
Propriété OperationMessage (Page 490)
Propriété ObjectName (Page 485)
Propriété Min (Page 479)
Propriété Max (Page 461)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété FlashRateBorderColor (Page 401)
Propriété FlashRateBackColor (Page 400)
Propriété FlashBorderColor (Page 398)
Propriété FlashBackColor (Page 398)
Propriété FillStyle (Page 394)
Propriété FillingIndex (Page 393)
Propriété Filling (Page 393)
Propriété FillColor (Page 392)
Propriété ExtendedOperation (Page 390)
Propriété Enabled (Page 382)
Propriété Direction (Page 380)
Propriété ColorTop (Page 353)
Propriété ColorBottom (Page 351)
Propriété BorderBackColor (Page 336)
Propriété BorderWidth (Page 334)
Propriété BorderStyle (Page 334)
Propriété BorderFlashColorOn (Page 334)

1.14.3.5 Objets tuyaux

Tuyau polygonal

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Tuyau polygonal".

Désignation du type sous VBS

HMITubePolyline

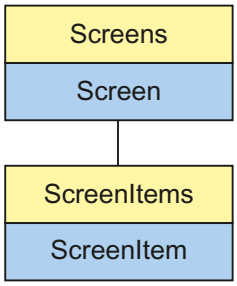
Utilisation

Dans l'exemple suivant, l'objet portant le nom "TubePolyline1" est déplacé sur la droite de 10 pixels :

```
'VBS24
Dim objTubePolyline
Set objTubePolyline = ScreenItems("TubePolyline1")
objTubePolyline.Left = objTubePolyline.Left + 10
```

Raccord en T

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Raccord en T".

Désignation du type sous VBS

HMITubeTeeObject

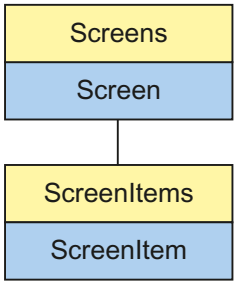
Utilisation

Dans l'exemple suivant, l'objet portant le nom "TubeTeeObject1" est déplacé sur la droite de 10 pixels :

```
'VBS21
Dim objTubeTeeObject
Set objTubeTeeObject = ScreenItems("TubeTeeObject1")
objTubeTeeObject.Left = objTubeTeeObject.Left + 10
```

Double raccord en T

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Double raccord en T".

Désignation du type sous VBS

HMITubeDoubleTeeObject

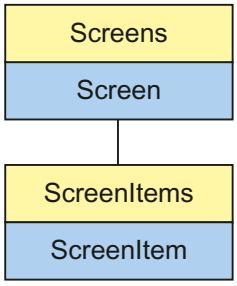
Utilisation

Dans l'exemple suivant, l'objet portant le nom "TubeDoubleTeeObject1" est déplacé sur la droite de 10 pixels :

```
'VBS21
Dim objTubeDoubleTeeObject
Set objTubeDoubleTeeObject = ScreenItems("TubeDoubleTeeObject1")
objTubeDoubleTeeObject.Left = objTubeDoubleTeeObject.Left + 10
```

Coude de tuyau

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Coude de tuyau".

Désignation du type sous VBS

HMITubeArcObject

Utilisation

Dans l'exemple suivant, l'objet portant le nom "TubeArcObject1" est déplacé sur la droite de 10 pixels :

```
'VBS24
Dim objTubeArcObject
Set objTubeArcObject = ScreenItems("TubeArcObject1")
objTubeArcObject.Left = objTubeArcObject.Left + 10
```

1.14.3.6 Contrôles

Contrôles

Particularités des contrôles

Pour les Controls WinCC d'une autre marque, c'est l'identification ProgID indépendante de la version qui est retransmise comme type.

A partir de ProgID, vous pouvez déterminer l'identification ProgID dépendant de la version ou le "User friendly Name" : Dans l'exemple suivant, "Control1" est un Control incorporé dans la

vue et qui retransmet déjà l'identification ProgID indépendante de la version, via la propriété Type.

Remarque

Etant donné que ce ne sont pas tous les Controls qui ont des ProgID dépendant de la version, il vous faut intégrer une fonction de traitement des erreurs pour sortir l'identification ProgID dépendant de la version ou le UserFriendlyName. Si une fonction de traitement des erreurs est utilisée, le code se termine immédiatement sans résultat si aucune ProgID n'est trouvée.

Comment déterminer l'identification ProgID dépendant de la version :

```
'VBS153
Dim objControl
Dim strCurrentVersion
Set objControl = ScreenItems("Control1")
strCurrentVersion = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type &
"\CurVer\")
MsgBox strCurrentVersion
```

Remarque

Pour que l'exemple donné ci-dessus fonctionne, il vous faut intégrer un contrôle multimédia dans la vue.

Déterminez le User friendly Name comme suit :

```
'VBS154
Dim objControl
Dim strFriendlyName
Set objControl = ScreenItems("Control1")
strFriendlyName = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type & "\")
MsgBox strFriendlyName
```

Remarque

Pour que l'exemple donné ci-dessus fonctionne, il vous faut intégrer un contrôle multimédia dans la vue.

Restrictions de VBS pour les dynamisations des Controls

Si des Controls doivent être dynamisés avec VBS, les conditions suivantes doivent au préalable être remplies :

Méthodes

La déclaration "ByRef" peut uniquement être implémentée en tant que "Variant" (ByRef xxx as Variant)

La déclaration "ByVal" peut uniquement être implémentée avec des types de variables (ByVal xxx as Long)

Propriétés

La déclaration "ByRef" peut uniquement être implémentée en tant que "Variant" (ByRef xxx as Variant)

La déclaration "ByVal" peut uniquement être implémentée avec des types de variables (ByVal xxx as Long)

Evénements

La déclaration "ByRef" n'est pas autorisée.

La déclaration "ByVal" peut uniquement être implémentée en tant que "Variant" (ByVal xxx as Variant)

Arrays

Si vous utilisez des Arrays, il vous faut déclarer ces derniers avec (ByRef xxx As Variant).

Pour que les Arrays puissent être transmis, il faut activer entre eux une variable Variant en tant que variable intermédiaire supplémentaire selon le schéma suivant :

```
'VBS151
Dim arrayPoints(200)
Dim vArrayCoercion      'Variant for array Coercion
' Make the VBS Array compatibile with the OLE Automation
vArrayCoercion = (arrayPoints)
objTrendControl.DataXY = vArrayCoercion   ' this array will occur in the control
```

Utilisation de Controls d'une autre marque

Si vous utilisez un Control n'ayant pas été mis à disposition par WinCC, il se peut que les propriétés apportées dans ce Control portent les mêmes noms que les propriétés ScreenItem générales. Dans ce cas, ce sont les propriétés ScreenItem qui ont priorité. Vous pouvez accéder aux propriétés "cachées" d'un Control d'une autre marque par le biais de la propriété supplémentaire "object". Vous accédez donc par exemple aux propriétés d'un Control d'une autre marque sous la forme suivante :

```
Control.object.type
```

Si vous recourez à la forme suivante, les propriétés de l'objet ScreenItem sont utilisées en cas d'identité de nom :

```
Control.type
```

Paramètres doubles

Lorsqu'un Control propre à WinCC est utilisé, il peut arriver également qu'un paramètre avec le nom "Item" existe dans les prototypes d'événement. Dans ce cas, le nom du paramètre est renommé en "ObjectItem" dans les prototypes VBS présentés. Si ce nom aussi existe déjà, le nom est différencié par ajout de chiffres.

WinCC Controls disponibles

- HMI Symbol Library
- WinCC AlarmControl
- WinCC Alarm Control (avant WinCC V7)
- WinCC Digital/Analog Clock
- WinCC FunctionTrendControl
- WinCC Function Trend Control (avant WinCC V7)
- WinCC Gauge Control
- WinCC Media Control
- WinCC OnlineTableControl
- WinCC Online Table Control (avant WinCC V7)
- WinCC OnlineTrendControl
- WinCC Online Trend Control (avant WinCC V7)
- WinCC Push Button Control
- WinCC RulerControl
- WinCC Slider Control
- WinCC UserArchiveControl

Voir aussi

HMI Symbol Library (Page 246)
WinCC Slider Control (Page 275)
WinCC Push Button Control (Page 268)
WinCC Online Trend Control (avant WinCC V7) (Page 288)
WinCC Online Table Control (avant WinCC V7) (Page 286)
WinCC Gauge Control (Page 258)
WinCC Function Trend Control (avant WinCC V7) (Page 284)
WinCC Digital Analog Clock (Page 251)
WinCC Alarm Control (avant WinCC V7) (Page 282)
WinCC UserArchiveControl (Page 279)
WinCC RulerControl (Page 272)
WinCC OnlineTrendControl (Page 264)
WinCC OnlineTableControl (Page 260)
WinCC FunctionTrendControl (Page 254)
WinCC AlarmControl (Page 248)

Listes des contrôles

Objet Column (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Column" permet de configurer les propriétés des colonnes dans WinCC UserArchiveControl.

Utilisation dans les Controls

- WinCC UserArchiveControl (Page 279)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "colobj.ColumnName", le nom "Column" est ignoré : "colobj.Name".

ColumnAlias (Page 355)	ColumnFlagUnique (Page 357)	ColumnPosition (Page 359)	ColumnSort (Page 362)
ColumnAlign (Page 355)	ColumnHideText (Page 358)	ColumnPrecisions (Page 360)	ColumnSortIndex (Page 363)
ColumnAutoPrecisions (Page 355)	ColumnHideTitleText (Page 358)	ColumnReadAccess (Page 360)	ColumnStartValue (Page 363)
ColumnCaption (Page 356)	ColumnIndex (Page 358)	ColumnReadonly (Page 360)	ColumnStringLength (Page 363)
ColumnCount (Page 356)	ColumnLeadingZeros (Page 358)	ColumnRepos (Page 361)	ColumnTimeFormat (Page 363)
ColumnDateFormat (Page 356)	ColumnLength (Page 359)	ColumnShowDate (Page 362)	ColumnType (Page 364)
ColumnDMVarName (Page 356)	ColumnMaxValue (Page 359)	ColumnShowIcon (Page 362)	ColumnVisible (Page 365)
ColumnExponentialFormat (Page 357)	ColumnMinValue (Page 359)	ColumnShowTitleIcon (Page 362)	ColumnWriteAccess (Page 365)
ColumnFlagNotNull (Page 357)	ColumnName (Page 359)		

Voir aussi

Méthode GetColumn (Page 690)

Méthode GetColumnCollection (Page 691)

Objet HitlistColumn (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "HitlistColumn" permet de configurer les blocs d'alarme utilisés dans la liste d'occurrences de WinCC AlarmControl.

Utilisation dans les Controls

- WinCC AlarmControl (Page 248)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "hitlistobj.HitlistColumnName", le nom "HitlistColumn" est ignoré : "hitlistobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

HitlistColumnAdd (Page 418)	HitlistColumnRepos (Page 419)	HitListMaxSourceItems (Page 421)
HitlistColumnCount (Page 418)	HitlistColumnSort (Page 420)	HitListMaxSourceItemsWarn (Page 421)
HitlistColumnIndex (Page 419)	HitlistColumnSortIndex (Page 420)	HitListRelTime (Page 421)
HitlistColumnName (Page 419)	HitlistColumnVisible (Page 420)	HitListRelTimeFactor (Page 421)
HitlistColumnRemove (Page 419)	HitListDefaultSort (Page 420)	HitListRelTimeFactorType (Page 422)

Voir aussi

Méthode GetHitlistColumn (Page 693)

Méthode GetHistlistColumnCollection (Page 694)

Objet MessageBlock (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "MessageBlock" permet de configurer les blocs d'alarme dans WinCC AlarmControl.

Utilisation dans les Controls

- WinCC AlarmControl (Page 248)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "messageobj.MessageBlockName", le nom "MessageBlock" est ignoré : "messageobj.Name".

MessageBlockAlign (Page 468)	MessageBlockFlashOn (Page 470)	MessageBlockLength (Page 473)	MessageBlockShowIcon (Page 474)
MessageBlockAutoPrecisions (Page 468)	MessageBlockHideText (Page 471)	MessageBlockName (Page 473)	MessageBlockShowTitleIcon (Page 474)
MessageBlockCaption (Page 469)	MessageBlockHideTitleText (Page 471)	MessageBlockPrecisions (Page 473)	MessageBlockTextId (Page 475)
MessageBlockCount (Page 469)	MessageBlockID (Page 471)	MessageBlockSelected (Page 473)	MessageBlockTimeFormat (Page 475)
MessageBlockDateFormat	MessageBlockIndex (Page 472)	MessageBlockShowDate (Page 474)	MessageBlockType (Page 475)
MessageBlockExponentialFormat (Page 470)	MessageBlockLeadingZeros (Page 472)		

Voir aussi

Méthode GetMessageBlock (Page 695)

Méthode GetMessageBlockCollection (Page 696)

Objet MessageColumn (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "MessageColumn" permet de configurer les blocs d'alarme utilisés dans les listes d'alarmes de WinCC AlarmControl.

Utilisation dans les Controls

- WinCC AlarmControl (Page 248)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "messagecol.MessageColumnName", le nom "MessageColumn" est ignoré : "messagecol.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

MessageColumnAdd (Page 476)	MessageColumnName (Page 477)	MessageColumnSort (Page 478)
MessageColumnCount (Page 476)	MessageColumnRemove (Page 477)	MessageColumnSortIndex (Page 478)
MessageColumnIndex (Page 477)	MessageColumnRepos (Page 477)	MessageColumnVisible (Page 478)

Voir aussi

Méthode GetMessageColumn (Page 697)

Méthode GetMessageColumnCollection (Page 698)

Objet OperatorMessage (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "OperatorMessage" permet de configurer les messages de conduite affichés dans WinCC AlarmControl.

Utilisation dans les Controls

- WinCC AlarmControl (Page 248)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "opmessobj.OperatorMessageName", le nom "OperatorMessage" est ignoré : "opmessobj.Name".

OperatorMessageID (Page 490)	OperatorMessageSource5 (Page 493)	OperatorMessageSourceType3 (Page 496)
OperatorMessageIndex (Page 490)	OperatorMessageSource6 (Page 493)	OperatorMessageSourceType4 (Page 496)
OperatorMessageName (Page 491)	OperatorMessageSource7 (Page 494)	OperatorMessageSourceType5 (Page 496)
OperatorMessageNumber (Page 491)	OperatorMessageSource8 (Page 494)	OperatorMessageSourceType6 (Page 497)
OperatorMessageSelected (Page 492)	OperatorMessageSource9 (Page 494)	OperatorMessageSourceType7 (Page 497)
OperatorMessageSource1 (Page 492)	OperatorMessageSource10 (Page 495)	OperatorMessageSourceType8 (Page 497)
OperatorMessageSource2 (Page 492)	OperatorMessageSourceType1 (Page 495)	OperatorMessageSourceType9 (Page 498)

OperatorMessageSource3 (Page 492)	OperatorMessageSourceType2 (Page 495)	OperatorMessageSourceType10 (Page 498)
OperatorMessageSource4 (Page 493)		

Voir aussi

Méthode GetOperatorMessage (Page 699)

Méthode GetOperatorMessageCollection (Page 700)

Objet Row (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Row" permet d'accéder aux lignes des Controls basés sur des tables. L'objet Row se réfère aux données Runtime dans les tables.

Utilisation dans les Controls

WinCC AlarmControl (Page 248)	WinCC OnlineTableControl (Page 260)
WinCC RulerControl (Page 272)	WinCC UserArchiveControl (Page 279)

Méthodes disponibles de l'objet

SelectAll (Page 767)	SelectRow (Page 768)
UnselectAll (Page 782)	UnselectRow (Page 783)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "rowobj.RowCellCount", le nom "Row" est ignoré : "rowobj.CellCount".

RowCellCount (Page 525)	RowCellText (Page 526)
RowCount (Page 526)	RowNumber (Page 526)

Voir aussi

Méthode GetRow (Page 701)

Méthode GetRowCollection (Page 702)

Méthode GetSelectedRow (Page 709)

Méthode GetSelectedRows (Page 710)

Objet RulerBlock (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Block" permet de configurer les blocs de WinCC RulerControl.

Utilisation dans les Controls

- WinCC RulerControl (Page 272)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "rulerblockobj.BlockName", le nom "Block" est ignoré : "rulerblockobj.Name".

BlockAlign (Page 326)	BlockHideText (Page 328)	BlockPrecisions (Page 330)
BlockAutoPrecisions (Page 327)	BlockHideTitleText (Page 328)	BlockShowDate (Page 330)
BlockCaption (Page 327)	BlockID	BlockShowIcon (Page 330)
BlockCount (Page 327)	BlockIndex (Page 329)	BlockShowTitleIcon (Page 331)
BlockDateFormat (Page 327)	BlockLength (Page 329)	BlockTimeFormat (Page 331)
BlockExponentialFormat (Page 328)	BlockName (Page 330)	BlockUseSourceFormat (Page 331)

Voir aussi

Méthode GetRulerBlock (Page 703)

Méthode GetRulerBlockCollection (Page 704)

Objet RulerColumn (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Column" permet de configurer les colonnes de la fenêtre règle dans WinCC RulerControl.

Utilisation dans les Controls

- WinCC RulerControl (Page 272)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "rulercolobj.ColumnName", le nom "Column" est ignoré : "rulercolobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

ColumnAdd (Page 355)	ColumnName (Page 359)	ColumnSort (Page 362)
ColumnCount (Page 356)	ColumnRemove (Page 360)	ColumnSortIndex (Page 363)
ColumnIndex (Page 358)	ColumnRepos (Page 361)	ColumnVisible (Page 365)

Voir aussi

Méthode GetRulerColumn (Page 706)

Méthode GetRulerColumnCollection (Page 707)

Objet StatisticAreaColumn (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Column" permet de configurer les colonnes de la fenêtre plage de statistiques dans WinCC RulerControl.

Utilisation dans les Controls

- WinCC RulerControl (Page 272)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "statareacolobj.ColumnName", le nom "Column" est ignoré : "statareacolobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

ColumnAdd (Page 355)	ColumnName (Page 359)	ColumnSort (Page 362)
ColumnCount (Page 356)	ColumnRemove (Page 360)	ColumnSortIndex (Page 363)
ColumnIndex (Page 358)	ColumnRepos (Page 361)	ColumnVisible (Page 365)

Voir aussi

Méthode GetStatisticAreaColumn (Page 711)

Méthode GetStatisticAreaColumnCollection (Page 712)

Objet StatisticResultColumn (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Column" permet de configurer les colonnes de la fenêtre statistiques dans WinCC RulerControl.

Utilisation dans les Controls

- WinCC RulerControl (Page 272)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "statrescolobj.ColumnName", le nom "Column" est ignoré : "statrescolobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

ColumnAdd (Page 355)	ColumnName (Page 359)	ColumnSort (Page 362)
ColumnCount (Page 356)	ColumnRemove (Page 360)	ColumnSortIndex (Page 363)
ColumnIndex (Page 358)	ColumnRepos (Page 361)	ColumnVisible (Page 365)

Voir aussi

Méthode GetStatisticResultColumn (Page 713)

Méthode GetStatisticResultColumnCollection (Page 714)

Objet StatusbarElement (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "StatusbarElement" permet de configurer les propriétés de la barre d'état des Controls.

Utilisation dans les Controls

WinCC AlarmControl (Page 248)	WinCC FunctionTrendControl (Page 254)	WinCC OnlineTableControl (Page 260)
WinCC OnlineTrendControl (Page 264)	WinCC RulerControl (Page 272)	WinCC UserArchiveControl (Page 279)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "statusbarobj.StatusbarElementName", le nom "StatusbarElement" est ignoré : "statusbarobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

StatusbarElementAdd (Page 561)	StatusbarElementIndex (Page 562)	StatusbarElementText (Page 563)
StatusbarElementAutoSize (Page 561)	StatusbarElementName (Page 562)	StatusbarElementTooltipText (Page 563)
StatusbarElementCount (Page 561)	StatusbarElementRemove (Page 562)	StatusbarElementUserDefined (Page 564)
StatusbarElementIconId (Page 562)	StatusbarElementRename (Page 563)	StatusbarElementVisible (Page 564)
StatusbarElementId (Page 562)	StatusbarElementRepos (Page 563)	StatusbarElementWidth (Page 564)

Voir aussi

Méthode GetStatusbarElement (Page 715)

Méthode GetStatusbarElementCollection (Page 716)

Objet TimeAxis (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "TimeAxis" permet de configurer les propriétés de l'axe des temps dans WinCC OnlineTrendControl.

Utilisation dans les Controls

- WinCC OnlineTrendControl (Page 264)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "timeaxisobj.TimeAxisName", le nom "TimeAxis" est ignoré : "timeaxisobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

TimeAxisActualize	TimeAxisIndex (Page 577)	TimeAxisRepos (Page 579)
TimeAxisAdd (Page 575)	TimeAxisInTrendColor	TimeAxisShowDate (Page 580)
TimeAxisAlign (Page 575)	TimeAxisLabel (Page 578)	TimeAxisTimeFormat (Page 580)
TimeAxisBeginTime (Page 575)	TimeAxisMeasurePoints (Page 578)	TimeAxisTimeRangeBase (Page 580)
TimeAxisColor (Page 576)	TimeAxisName (Page 578)	TimeAxisTimeRangeFactor (Page 581)
TimeAxisCount (Page 576)	TimeAxisRangeType (Page 579)	TimeAxisTrendWindow (Page 581)
TimeAxisDateFormat (Page 576)	TimeAxisRemove (Page 579)	TimeAxisVisible (Page 581)
TimeAxisEndTime (Page 576)	TimeAxisRename (Page 579)	

Voir aussi

Méthode GetTimeAxis (Page 717)

Méthode GetTimeAxisCollection (Page 719)

Objet TimeColumn (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "TimeColumn" permet de configurer les propriétés de l'axe de la colonne heure dans WinCC OnlineTableControl.

Utilisation dans les Controls

- WinCC OnlineTableControl (Page 260)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "timecolobj.TimeColumnName", le nom "TimeColumn" est ignoré : "timecolobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

TimeColumnActualize (Page 582)	TimeColumnHideText (Page 585)	TimeColumnShowDate (Page 588)
TimeColumnAdd (Page 582)	TimeColumnHideTitleText (Page 586)	TimeColumnShowIcon (Page 588)
TimeColumnAlign (Page 583)	TimeColumnIndex (Page 586)	TimeColumnShowTitleIcon (Page 589)
TimeColumnBackColor (Page 583)	TimeColumnLength (Page 586)	TimeColumnSort (Page 589)
TimeColumnBeginTime (Page 584)	TimeColumnMeasurePoints (Page 586)	TimeColumnSortIndex (Page 589)
TimeColumnCaption (Page 584)	TimeColumnName (Page 587)	TimeColumnTimeFormat (Page 589)
TimeColumnCount (Page 584)	TimeColumnRangeType (Page 587)	TimeColumnTimeRangeBase (Page 590)
TimeColumnDateFormat (Page 584)	TimeColumnRemove (Page 587)	TimeColumnTimeRangeFactor (Page 590)
TimeColumnEndTime (Page 585)	TimeColumnRename (Page 587)	TimeColumnUseValueColumnColors (Page 591)
TimeColumnForeColor (Page 585)	TimeColumnRepos (Page 588)	TimeColumnVisible (Page 591)

Voir aussi

Méthode GetTimeColumn (Page 720)

Méthode GetTimeColumnCollection (Page 721)

Objet ToolbarButton (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "ToolbarButton" permet de configurer les propriétés de la barre d'outils des Controls.

Utilisation dans les Controls

WinCC AlarmControl (Page 248)	WinCC FunctionTrendControl (Page 254)	WinCC OnlineTableControl (Page 260)
WinCC OnlineTrendControl (Page 264)	WinCC RulerControl (Page 272)	WinCC UserArchiveControl (Page 279)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "toolbarobj.ToolbarButtonName", le nom "ToolbarButton" est ignoré : "toolbarobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

ToolbarButtonActive (Page 603)	ToolbarButtonId (Page 607)	ToolbarButtonRename (Page 609)
ToolbarButtonAdd (Page 603)	ToolbarButtonIndex (Page 607)	ToolbarButtonRepos (Page 609)
ToolbarButtonBeginGroup (Page 603)	ToolbarButtonLocked (Page 608)	ToolbarButtonTooltipText (Page 609)
ToolbarButtonCount (Page 607)	ToolbarButtonName (Page 608)	ToolbarButtonUserDefined (Page 609)
ToolbarButtonEnabled (Page 607)	ToolbarButtonPasswordLevel (Page 608)	ToolbarButtonVisible (Page 610)
ToolbarButtonHotKey (Page 607)	ToolbarButtonRemove (Page 608)	

Voir aussi

Méthode GetToolbarButton (Page 722)

Méthode GetToolbarButtonCollection (Page 723)

Objet Trend (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Trend" permet de configurer les propriétés des courbes. Vous renseignez la courbe avec la méthode "InsertData" et vous la supprimez avec la méthode "RemoveData". La méthode "GetRulerData" permet d'accéder aux données à un endroit précis de la courbe.

Utilisation dans les Controls

WinCC FunctionTrendControl (Page 254)	WinCC OnlineTrendControl (Page 264)

Méthodes disponibles de l'objet

GetRulerData (Page 708)	InsertData (Page 739)	RemoveData (Page 764)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "trendobj.Trendname", le nom "Trend" est ignoré : "trendobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

Propriétés dans WinCC FunctionTrendControl et WinCC OnlineTrendControl

TrendAdd (Page 617)	TrendLineWidth (Page 622)	TrendRemove (Page 626)
TrendColor (Page 619)	TrendLowerLimit (Page 622)	TrendRename (Page 626)
TrendCount (Page 619)	TrendLowerLimitColor (Page 622)	TrendRepos (Page 626)
TrendExtendedColorSet	TrendLowerLimitColoring (Page 622)	TrendTrendWindow (Page 629)
TrendFill (Page 620)	TrendName (Page 623)	TrendUncertainColor (Page 629)
TrendFillColor (Page 620)	TrendPointColor (Page 623)	TrendUncertainColoring (Page 629)
TrendIndex (Page 620)	TrendPointStyle	TrendUpperLimit (Page 629)
TrendLabel (Page 621)	TrendPointWidth (Page 624)	TrendUpperLimitColor (Page 630)
TrendLineStyle (Page 621)	TrendProvider (Page 624)	TrendUpperLimitColoring (Page 630)
TrendLineType (Page 621)	TrendProviderCLSID (Page 625)	TrendVisible (Page 631)

Propriétés dans WinCC OnlineTrendControl

TrendAutoRangeBeginTagName (Page 617)	TrendAutoRangeSource (Page 618)	TrendValueAlignment
TrendAutoRangeBeginValue (Page 617)	TrendSelectTagName (Page 626)	TrendValueAxis (Page 631)
TrendAutoRangeEndTagName (Page 617)	TrendTagName (Page 627)	TrendValueUnit
TrendAutoRangeEndValue (Page 618)	TrendTimeAxis (Page 628)	

Propriétés dans WinCC FunctionTrendControl

TrendActualize (Page 617)	TrendSelectTagNameX (Page 627)	TrendTimeRangeBase (Page 628)
TrendBeginTime (Page 618)	TrendSelectTagNameY (Page 627)	TrendTimeRangeFactor (Page 628)
TrendEndTime (Page 619)	TrendTagNameX (Page 627)	TrendXAxis (Page 638)
TrendMeasurePoints (Page 623)	TrendTagNameY (Page 628)	TrendYAxis (Page 638)
TrendRangeType		

Voir aussi

Méthode GetTrend (Page 724)

Méthode GetTrendCollection (Page 725)

Objet TrendWindow (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "TrendWindow" permet de configurer les propriétés de la fenêtre des courbes.

Utilisation dans les Controls

WinCC FunctionTrendControl (Page 254)	WinCC OnlineTrendControl (Page 264)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "trendwndobj.TrendWindowName", le nom "TrendWindow" est ignoré : "trendwndobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

Propriétés dans WinCC FunctionTrendControl et WinCC OnlineTrendControl

TrendWindowAdd (Page 631)	TrendWindowGridInTrendColor (Page 633)	TrendWindowRulerColor (Page 635)
TrendWindowCoarseGrid (Page 632)	TrendWindowHorizontalGrid (Page 634)	TrendWindowRulerLayer
TrendWindowCoarseGridColor (Page 632)	TrendWindowIndex (Page 634)	TrendWindowRulerStyle (Page 636)
TrendWindowCount (Page 632)	TrendWindowName (Page 634)	TrendWindowRulerWidth (Page 636)
TrendWindowFineGrid (Page 632)	TrendWindowRemove (Page 634)	TrendWindowSpacePortion (Page 636)
TrendWindowFineGridColor (Page 633)	TrendWindowRename (Page 635)	TrendWindowVerticalGrid (Page 637)
TrendWindowForegroundTrendGrid (Page 633)	TrendWindowRepos (Page 635)	TrendWindowVisible (Page 638)

Propriétés dans WinCC OnlineTrendControl

TrendWindowStatisticRulerColor (Page 636)	TrendWindowStatisticRulerStyle (Page 637)	TrendWindowStatisticRulerWidth (Page 637)

Voir aussi

Méthode GetTrendWindow (Page 727)

Méthode GetTrendWindowCollection (Page 728)

Objet ValueAxis (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "ValueAxis" permet de configurer les propriétés de l'axe des valeurs dans WinCC OnlineTrendControl.

Utilisation dans les Controls

- WinCC OnlineTrendControl (Page 264)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "valueaxisobj.ValueAxisName", le nom "ValueAxis" est ignoré : "valueaxisobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

ValueAxisAdd (Page 653)	ValueAxisEndValue (Page 655)	ValueAxisRemove (Page 657)
ValueAxisAlign (Page 653)	ValueAxisExponentialFormat (Page 655)	ValueAxisRename (Page 657)
ValueAxisAutoPrecisions (Page 654)	ValueAxisIndex (Page 655)	ValueAxisRepos (Page 657)
ValueAxisAutoRange (Page 654)	ValueAxisInTrendColor	ValueAxisScalingType (Page 658)
ValueAxisBeginValue (Page 654)	ValueAxisLabel (Page 656)	ValueAxisTrendWindow (Page 658)
ValueAxisColor (Page 654)	ValueAxisName (Page 656)	ValueAxisVisible (Page 658)
ValueAxisCount (Page 655)	ValueAxisPrecisions (Page 657)	

Voir aussi

Méthode GetValueAxis (Page 729)

Méthode GetValueAxisCollection (Page 730)

Objet ValueColumn (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "ValueColumn" permet de configurer les propriétés de la colonne des valeurs dans WinCC OnlineTableControl.

Utilisation dans les Controls

- WinCC OnlineTableControl (Page 260)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "valcolobj.ValueColumnName", le nom "ValueColumn" est ignoré : "valcolobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

ValueColumnAdd (Page 658)	ValueColumnHideTitleText (Page 661)	ValueColumnRepos (Page 663)
ValueColumnAlign (Page 659)	ValueColumnIndex (Page 661)	ValueColumnSelectTagName (Page 664)
ValueColumnAutoPrecisions (Page 659)	ValueColumnLength (Page 662)	ValueColumnShowIcon (Page 664)
ValueColumnBackColor (Page 660)	ValueColumnName (Page 662)	ValueColumnShowTitleIcon (Page 664)
ValueColumnCaption (Page 660)	ValueColumnPrecisions (Page 662)	ValueColumnSort (Page 664)
ValueColumnCount (Page 660)	ValueColumnProvider (Page 662)	ValueColumnSortIndex (Page 665)
ValueColumnExponentialFormat (Page 660)	ValueColumnProviderCLSID (Page 663)	ValueColumnTagName (Page 665)
ValueColumnForeColor (Page 661)	ValueColumnRemove (Page 663)	ValueColumnTimeColumn (Page 665)
ValueColumnHideText (Page 661)	ValueColumnRename (Page 663)	ValueColumnVisible (Page 666)

Voir aussi

Méthode GetValueColumn (Page 731)

Méthode GetValueColumnCollection (Page 732)

Objet XAxis (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Xaxis" permet de configurer les propriétés de l'axe X dans WinCC FunctionTrendControl.

Utilisation dans les Controls

- WinCC FunctionTrendControl (Page 254)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "xaxisobj.XAxisName", le nom "XAxis" est ignoré : "xaxisobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

XAxisAdd (Page 672)	XAxisEndValue (Page 675)	XAxisRemove (Page 676)
XAxisAlign	XAxisExponentialFormat (Page 675)	XAxisRename (Page 678)
XAxisAutoPrecisions (Page 673)	XAxisIndex (Page 678)	XAxisRepos (Page 677)
XAxisAutoRange (Page 674)	XAxisInTrendColor	XAxisScalingType (Page 677)
XAxisBeginValue (Page 674)	XAxisLabel (Page 676)	XAxisTrendWindow (Page 677)
XAxisColor (Page 674)	XAxisName (Page 676)	XAxisVisible (Page 678)
XAxisCount (Page 678)	XAxisPrecisions (Page 676)	

Voir aussi

Méthode GetXAxis (Page 734)

Méthode GetXAxisCollection (Page 735)

Objet YAxis (liste)

Description

La liste d'un Control est un conteneur de données qui peut enregistrer un nombre variable d'objets de même type.

L'objet de liste "Yaxis" permet de configurer les propriétés de l'axe Y dans WinCC FunctionTrendControl.

Utilisation dans les Controls

- WinCC FunctionTrendControl (Page 254)

Propriétés disponibles de l'objet

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "yaxisobj.YAxisName", le nom "YAxis" est ignoré : "yaxisobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

YAxisAdd (Page 672)	YAxisEndValue (Page 675)	YAxisRemove (Page 676)
YAxisAlign (Page 673)	YAxisExponentialFormat (Page 675)	YAxisRename (Page 679)
YAxisAutoPrecisions (Page 673)	YAxisIndex (Page 679)	YAxisRepos (Page 677)
YAxisAutoRange (Page 674)	YAxisInTrendColor (Page 675)	YAxisScalingType (Page 677)
YAxisBeginValue (Page 674)	YAxisLabel (Page 676)	YAxisTrendWindow (Page 677)
YAxisColor (Page 674)	YAxisName (Page 676)	YAxisVisible (Page 678)
YAxisCount (Page 678)	YAxisPrecisions (Page 676)	

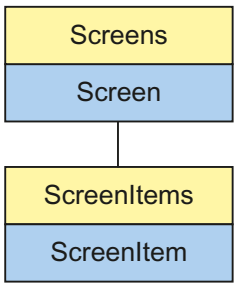
Voir aussi

Méthode GetYAxis (Page 736)

Méthode GetYAxisCollection (Page 737)

HMI Symbol Library

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "HMI Symbol Library"

Désignation du type sous VBS

HMISymbolLibrary

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 20 Pixel vers la droite :

```
'VBS64
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +20
```

Propriétés

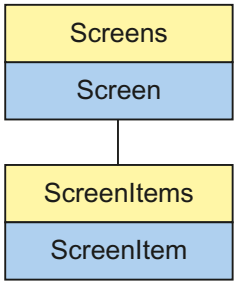
Ce type d'objet a les propriétés suivantes :

Voir aussi

Propriété Left (Page 449)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété Stretch (Page 566)
Propriété Picture (Page 509)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Object (Page 484)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété ForeColor (Page 408)
Propriété Flip (Page 402)
Propriété Enabled (Page 382)
Propriété Cursor (Page 370)
Propriété BlinkColor (Page 325)
Propriété BackStyle (Page 317)
Propriété BackColor (Page 313)

WinCC AlarmControl

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC AlarmControl" à partir de WinCC V7.0.

Désignation du type sous VBS

HMIAlarmControl

Objets de liste disponibles

HitlistColumn (Page 229)	Row (Page 233)
MessageBlock (Page 230)	StatusbarElement (Page 237)
MessageColumn (Page 231)	ToolbarButton (Page 239)
OperatorColumn (Page 232)	

Méthodes disponibles dans VBS

Activate	GetOperatorMessageCollection	MoveToLastLine	ShowHideList
ActivateDynamic	GetRow (Page 701)	MoveToLastPage	ShowHitList
AttachDB	GetRowCollection (Page 702)	MoveToNextLine	ShowInfoText
CopyRows	GetSelectedRow (Page 709)	MoveToNextPage	ShowLockDialog
DeactivateDynamic	GetSelectedRows (Page 710)	MoveToPreviousLine	ShowLockList
DetachDB	GetStatusbarElement	MoveToPreviousPage	ShowLongTermArchiveList
Export	GetStatusbarElementCollection	Print	ShowMessageList
GetHitlistColumn	GetToolbarButton	QuitHorn	ShowPropertyDialog
GetHitlistColumnCollection	GetToolbarButtonCollection	QuitSelected	ShowSelectionDialog
GetMessageBlock	HideAlarm	QuitVisible	ShowShortTermArchiveList
GetMessageBlockCollection	LockAlarm	ShowComment	ShowSortDialog

GetMessageColumn	LoopInAlarm	ShowDisplayOptionsDialog	ShowTimebaseDialog
GetMessageColumnCollection	MoveToFirstLine	ShowEmergencyQuitDialog	UnhideAlarm
GetOperatorMessage	MoveToFirstPage	ShowHelp	UnlockAlarm

Propriétés disponibles dans VBS

Si vous accédez aux propriétés à l'aide d'un objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "messagecol.MessageColumnName", le nom "MessageColumn" est ignoré : "messagecol.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

Activate	HitListRelTime	OperatorMessageSource10	StatusbarElementRepos
AllServer	HitListRelTimeFactor	OperatorMessageSourceType1	StatusbarElementText
ApplyProjectSettings	HitListRelTimeFactorType	OperatorMessageSourceType2	StatusbarElementTooltipText
AutoCompleteColumns	HorizontalGridLines	OperatorMessageSourceType3	StatusbarElementUserDefined
AutoCompleteRows	IconSpace	OperatorMessageSourceType4	StatusbarElementVisible
AutoScroll	LineColor (Page 453)	OperatorMessageSourceType5	StatusbarElementWidth
AutoSelectionColors	LineWidth (Page 454)	OperatorMessageSourceType6	StatusbarFontColor
AutoSelectionRectColor	LongTermArchiveConsistency	OperatorMessageSourceType7	StatusbarShowTooltips
BackColor	MessageBlockAlign	OperatorMessageSourceType8	StatusbarText
BorderColor	MessageBlockAutoPrecisions	OperatorMessageSourceType9	StatusbarUseBackColor
BorderWidth	MessageBlockCaption	OperatorMessageSourceType10	StatusbarVisible
Caption	MessageBlockCount	PageMode	TableColor
CellCut	MessageBlockDateFormat	PageModeMessageNumber	TableColor2
CellSpaceBottom	MessageBlockExponentialFormat	PrintJobName	TableForeColor
CellSpaceLeft	MessageBlockFlashOn	RowCellCount (Page 525)	TableForeColor2
CellSpaceRight	MessageBlockHideText	RowCellText (Page 526)	TimeBase
CellSpaceTop	MessageBlockHideTitleText	RowCount (Page 526)	TitleColor
Closeable	MessageBlockIndex	RowNumber (Page 526)	TitleCut
ColumnResize	MessageBlockLeadingZeros	RowScrollbar	TitleDarkShadowColor
ColumnTitleAlign	MessageBlockLength	RowTitleAlign	TitleForeColor
ColumnTitles	MessageBlockPrecisions	RowTitles	TitleGridLineColor
DefaultMsgFilterSQL	MessageBlockSelected	RTPersistence	TitleLightShadowColor

DefaultSort	MessageBlockShowDate	RTPersistencePasswordLevel	TitleSort
DefaultSort2	MessageBlockShowIcon	RTPersistenceType	TitleStyle
DefaultSort2Column	MessageBlockShowTitleIcon	SelectedCellColor	ToolbarAlignment
DisplayOptions	MessageBlockTextId	SelectedCellForeColor	ToolbarBackColor
DoubleClickAction	MessageBlockType	SelectedRowColor	ToolbarButtonActive
ExportDirectoryChangeable	MessageColumnAdd	SelectedRowForeColor	ToolbarButtonAdd
ExportDirectoryname	MessageColumnCount	SelectedTitleColor	ToolbarButtonBeginGroup
ExportFileExtension	MessageColumnIndex	SelectedTitleForeColor	ToolbarButtonClick
ExportFilename	MessageColumnName	SelectionColoring	ToolbarButtonCount
ExportFilenameChangeable	MessageColumnRemove	SelectionRect	ToolbarButtonEnabled
ExportFormatGuid	MessageColumnRepos	SelectionRectColor	ToolbarButtonHotKey
ExportFormatName	MessageColumnSort	SelectionRectWidth	ToolbarButtonId
ExportParameters	MessageColumnSortIndex	SelectionType	ToolbarButtonIndex
ExportSelection	MessageColumnVisible	ServerNames	ToolbarButtonLocked
ExportShowDialog	MessageListType	ShowSortButton	ToolbarButtonName
ExportXML	Moveable	ShowSortIcon	ToolbarButtonPasswordLevel
Font	MsgFilterSQL	ShowSortIndex	ToolbarButtonRemove
GridLineColor	OperatorMessageID	ShowTitle	ToolbarButtonRename
GridLineWidth	OperatorMessageIndex	Sizeable	ToolbarButtonRepos
HitlistColumnAdd	OperatorMessageName	SkinName	ToolbarButtonTooltipText
HitlistColumnCount	OperatorMessageNumber	SortSequence	ToolbarButtonUserDefined
HitlistColumnIndex	OperatorMessageSelected	StatusbarBackColor	ToolbarButtonVisible
HitlistColumnName	OperatorMessageSource1	StatusbarElementAdd	ToolbarShowTooltips
HitlistColumnRemove	OperatorMessageSource2	StatusbarElementAutoSize	ToolbarUseBackColor
HitlistColumnRepos	OperatorMessageSource3	StatusbarElementCount	ToolbarUseHotKeys
HitlistColumnSort	OperatorMessageSource4	StatusbarElementIconId	ToolbarVisible
HitlistColumnSortIndex	OperatorMessageSource5	StatusbarElementId	UseMessageColor
HitlistColumnVisible	OperatorMessageSource6	StatusbarElementIndex	UseSelectedTitleColor
HitListDefaultSort	OperatorMessageSource7	StatusbarElementName	UseTableColor2
HitListMaxSourceItems	OperatorMessageSource8	StatusbarElementRemove	VerticalGridLines
HitListMaxSourceItemsWarn	OperatorMessageSource9	StatusbarElementRename	

Exemple

on définit une sélection d'arlarmes dans un WinCC AlarmControl existant. Des propriétés sont configurées dans le script pour la courbe.

Condition

- Dans Graphics Designer, vous avez déjà inséré un "WinCC AlarmControl" portant le le "Control1" dans une vue de process. On utilise pour cet exemple la vue "C_015_Native_Alarms_Sel.pdl" du projet de démonstration.
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré l'événement "Clic de souris" avec une action VBS et le script suivant.

- Vous avez déjà configuré des alarmes dans le projet. Ou vous utilisez le projet de démonstration dont sont issues les alarmes utilisées dans l'exemple.
- Des alarmes ont déjà été déclenchées en runtime. Vous avez déjà cliqué dans le projet de démonstratio sur les boutons "arrivant" et "partant".

```
'VBS366
Sub OnClick(ByVal Item)
Dim objControl
Dim objMessColumn
Dim objMessBlock

Set objControl = ScreenItems("Control1")
objControl.ApplyProjectSettings = False
Set objMessBlock = objControl.GetMessageBlock("Date")
objMessBlock.DateFormat = "dd.MM.yy"
Set objMessColumn = objControl.GetMessageColumn("Time")
objMessColumn.Visible = False
objControl.MsgFilterSQL = "MSGNR >= 5 AND Priority = 0"
End Sub
```

Remarque

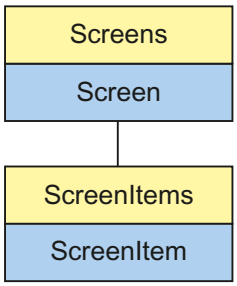
Pour plus d'exemples d'utilisation des propriétés et méthodes, référez-vous aux descriptions des méthodes Get des Controls et aux "Exemples de VBScript/Exemples dans WinCC/ Dynamisation des Controls".

Voir aussi

Contrôles (Page 224)

WinCC Digital Analog Clock

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Digital Analog Clock"

Désignation du type sous VBS

HMIClock

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 11 Pixel vers la droite :

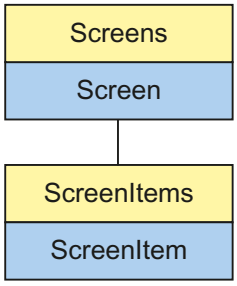
```
'VBS55
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +11
```

Voir aussi

Propriété Parent (Page 501)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété Ticks (Page 573)
Propriété TicksColor (Page 574)
Propriété SquareExtent (Page 559)
Propriété SecondNeedleWidth (Page 536)
Propriété SecondNeedleHeight (Page 535)
Propriété Picture (Page 509)
Propriété ObjectName (Page 485)
Propriété Object (Page 484)
Propriété MinuteNeedleWidth (Page 480)
Propriété MinuteNeedleHeight (Page 479)
Propriété LocaleID (Page 455)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété HourNeedleWidth (Page 423)
Propriété HourNeedleHeight (Page 423)
Propriété Height (Page 417)
Propriété Handtype (Page 416)
Propriété HandFillColor (Page 416)
Propriété ForeColor (Page 408)
Propriété Font (avant WinCC V7) (Page 404)
Propriété FocusRect (Page 403)
Propriété Enabled (Page 382)
Propriété BackStyle (Page 317)
Propriété BackColor (Page 313)
Propriété Analog (Page 303)

WinCC FunctionTrendControl

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC FunctionTrendControl" à partir de WinCC V7.0.

Désignation du type sous VBS

HMIFunctionTrendControl

Objets de liste disponibles

StatusbarElement (Page 237)	Tendance (Page 240)	XAxis (Page 245)
ToolbarButton (Page 239)	TrendWindow (Page 242)	YAxis (Page 245)

Méthodes disponibles dans VBS

Activate	GetToolbarButtonCollection	MoveAxis	ShowTrendSelection
ActivateDynamic	GetTrend	NextTrend	StartStopUpdate
AttachDB	GetTrendCollection	OneToOneView	ZoomArea
DeactivateDynamic	GetTrendWindow	PreviousTrend	ZoomInOut
DetachDB	GetTrendWindowCollection	Print	ZoomInOutX
Export	GetXAxis	ShowHelp	ZoomInOutY
GetStatusbarElement	GetXAxisCollection	ShowPropertyDialog	ZoomMove
GetStatusbarElementCollection	GetYAxis	ShowTagSelection	
GetToolbarButton	GetYAxisCollection	ShowTimeSelection	

Propriétés disponibles dans VBS

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "xaxisobj.XAxisName", le nom "XAxis" est ignoré : "xaxisobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

BackColor	StatusbarElementTooltipText	TrendLineWidth	TrendWindowVerticalGrid
BorderColor	StatusbarElementUserDefined	TrendLowerLimit	TrendWindowVisible
BorderWidth	StatusbarElementVisible	TrendLowerLimitColor	TrendXAxis
Caption	StatusbarElementWidth	TrendLowerLimitColoring	TrendYAxis
Closeable	StatusbarFontColor	TrendMeasurePoints	UseTrendNameAsLabel
ConnectTrendWindows	StatusbarShowTooltips	TrendName	XAxisAdd
ExportDirectoryChangeable	StatusbarText	TrendPointColor	XAxisAlign
ExportDirectoryname	StatusbarUseBackColor	TrendPointStyle	XAxisAutoPrecisions
ExportFileExtension	StatusbarVisible	TrendPointWidth	XAxisAutoRange
ExportFilename	TimeBase	TrendProvider	XAxisBeginValue
ExportFilenameChangeable	ToolbarAlignment	TrendProviderCLSID	XAxisColor
ExportFormatGuid	ToolbarBackColor	TrendRangeType	XAxisCount
ExportFormatName	ToolbarButtonActive	TrendRemove	XAxisEndValue
ExportSelection	ToolbarButtonAdd	TrendRename	XAxisExponentialFormat
ExportShowDialog	ToolbarButtonBeginGroup	TrendRepos	XAxisIndex
ExportParameters	ToolbarButtonClick	TrendSelectTagNameX	XAxisInTrendColor
ExportXML	ToolbarButtonCount	TrendSelectTagNameY	XAxisLabel
Font	ToolbarButtonEnabled	TrendTagNameX	XAxisName
GraphDirection	ToolbarButtonHotKey	TrendTagNameY	XAxisPrecisions
LineColor	ToolbarButtonId	TrendTimeRangeBase	XAxisRemove
LineWidth	ToolbarButtonIndex	TrendTimeRangeFactor	XAxisRename
LoadDataImmediately	ToolbarButtonLocked	TrendTrendWindow	XAxisRepos
Moveable	ToolbarButtonName	TrendUncertainColor	XAxisScalingType
Online	ToolbarButtonPasswordLevel	TrendUncertainColoring	XAxisTrendWindow
PrintJobName	ToolbarButtonRemove	TrendUpperLimit	XAxisVisible
RTPersistence	ToolbarButtonRename	TrendUpperLimitColor	XAxisAdd
RTPersistencePasswordLevel	ToolbarButtonRepos	TrendUpperLimitColoring	XAxisAlign
RTPersistenceType	ToolbarButtonTooltipText	TrendVisible	XAxisAutoPrecisions
ShowRuler	ToolbarButtonUserDefined	TrendWindowAdd	XAxisAutoRange
ShowRulerInAxis	ToolbarButtonVisible	TrendWindowCoarseGrid	XAxisBeginValue
ShowScrollbars	ToolbarShowTooltips	TrendWindowCount	XAxisColor
ShowTitle	ToolbarUseBackColor	TrendWindowCoarseGridColor	YAxisCount

Sizeable	ToolbarUseHotKeys	TrendWindowFineGrid	XAxisEndValue
ShowTrendIcon	ToolbarVisible	TrendWindowFineGridColor	XAxisExponentialFormat
SkinName	TrendActualize	TrendWindowForegroundTrendGrid	YAxisIndex
StatusbarBackColor	TrendAdd	TrendWindowGridInTrendColor	XAxisInTrendColor
StatusbarElementAdd	TrendBeginTime	TrendWindowHorizontalGrid	XAxisLabel
StatusbarElementAutoSize	TrendColor	TrendWindowIndex	XAxisName
StatusbarElementCount	TrendCount	TrendWindowName	XAxisPrecisions
StatusbarElementIconId	TrendEndTime	TrendWindowRemove	XAxisRemove
StatusbarElementId	TrendExtendedColorSet	TrendWindowRename	YAxisRename
StatusbarElementIndex	TrendFill	TrendWindowRepos	XAxisRepos
StatusbarElementName	TrendFillColor	TrendWindowRulerColor	XAxisScalingType
StatusbarElementRemove	TrendIndex	TrendWindowRulerLayer	XAxisTrendWindow
StatusbarElementRename	TrendLabel	TrendWindowRulerStyle	XAxisVisible
StatusbarElementRepos	TrendLineStyle	TrendWindowRulerWidth	
StatusbarElementText	TrendLineType	TrendWindowSpacePortion	

Exemples

Dans un WinCC FunctionTrendControl, une courbe reliée à une archive utilisateur est affichée. Différentes propriétés sont configurées dans le script pour la courbe. Concernant la liaison de données, le "StartID" de l'archive utilisateur et le nombre de points de mesures sont modifiés.

Condition

- Dans Graphics Designer, un "WinCC FunctionTrendControl" portant le nom "Control1" est inséré dans une vue de process.
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré l'événement "Clic de souris" avec une action VBS et le script suivant.
- Vous avez déjà configuré une archive utilisateur dans le projet. Ou vous utilisez le projet de démonstration dont est issue l'archive utilisateur.

```
'VBS363
Sub OnClick(ByVal Item)
Dim objFXControl
Dim objTrendWindow
Dim objTrend
Dim objXAxis
Dim objYAxis
Dim startID
Dim FXServerDataX(3)
Dim FXServerDataY(3)
' create reference to FXControl
Set objFXControl = ScreenItems("Control1")
' create reference to new window, x and y axis
Set objTrendWindow = objFXControl.GetTrendWindowCollection.AddItem("myWindow")
Set objXAxis = objFXControl.GetXAxisCollection.AddItem("myXAxis")
```

```
Set objYAxis = objFXControl.GetYAxisCollection.AddItem("myYAxis")
' assign x and y axis to the window
objXAxis.TrendWindow = objTrendWindow.Name
objYAxis.TrendWindow = objTrendWindow.Name
' add new trend
Set objTrend = objFXControl.GetTrendCollection.AddItem("myTrend1")
' configure trend data connection (UserArchive)
objTrend.Provider = 3
startID = CLng(4)
FXServerDataX(0) = "Setpoint"
FXServerDataX(1) = "ParabelX"
FXServerDataX(3) = startID
FXServerDataY(0) = "Setpoint"
FXServerDataY(1) = "ParabelY"
FXServerDataY(3) = startID
objTrend.MeasurePoints = 50
objTrend.SetTagName "Setpoint\ParabelX", "Setpoint\ParabelY", FXServerDataX, FXServerDataY
' assign trend properties
objTrend.Color = RGB(255,0,0)
objTrend.PointStyle = 1
objTrend.TrendWindow = objTrendWindow.Name
objTrend.XAxis = objXAxis.Name
objTrend.YAxis = objYAxis.Name
End Sub
```

Remarque

Pour plus d'exemples d'utilisation des propriétés et méthodes, référez-vous aux descriptions des méthodes Get des Controls et aux "Exemples de VBScript/Exemples dans WinCC/ Dynamisation des Controls".

Voir aussi

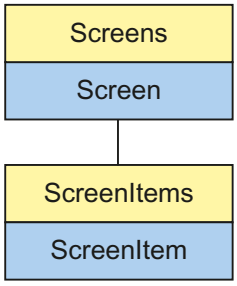
Contrôles (Page 224)

ServerDataX (Page 542)

ServerDataY (Page 543)

WinCC Gauge Control

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Gauge Control"

Désignation du type sous VBS

HMIGauge

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 14 Pixel vers la droite :

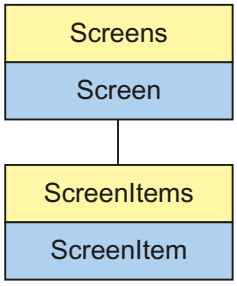
```
'VBS58
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +14
```

Voir aussi

Propriété WarningColor (Page 668)
Propriété Object (Page 484)
Propriété BackColor (Page 313)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Type d'objet de l'objet ScreenItem (Page 150)
Propriété Width (Page 669)
Propriété Warning (Page 668)
Propriété Visible (Page 667)
Propriété ValueMin (Page 666)
Propriété ValueMax (Page 666)
Propriété ValueColumnAlignment (Page 659)
Propriété UnitText (Page 644)
Propriété UnitOffset (Page 644)
Propriété UnitFont (Page 643)
Propriété UnitColor (Page 643)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété TicWidth (Page 573)
Propriété TicTextOffset (Page 573)
Propriété TicTextColor (Page 572)
Propriété TicOffset (Page 572)
Propriété TicFont (Page 571)
Propriété TicColor (Page 571)
Propriété ShowWarning (Page 551)
Propriété ShowPeak (Page 546)
Propriété ShowNormal (Page 546)
Propriété ShowDecimalPoint (Page 546)
Propriété ShowDanger (Page 545)
Propriété Rectangular (Page 521)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété NormalColor (Page 483)
Propriété NeedleColor (Page 483)
Propriété OperationMode (Page 455)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)

WinCC Media Control

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Media Control" à partir de WinCC V7.0.

Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Media Control" à partir de WinCC V7.0.

Désignation du type sous VBS

HMIMediaControl

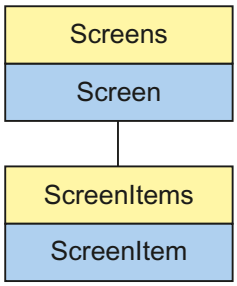
Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 16 pixels vers la droite :

```
'VBS60
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left + 16
```

WinCC OnlineTableControl

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC OnlineTableControl" à partir de WinCC V7.0.

Désignation du type sous VBS

HMIOnlineTableControl

Objets de liste disponibles

Row (Page 233)	ToolbarButton (Page 239)
StatusbarElement (Page 237)	ValueColumn (Page 244)
TimeColumn (Page 238)	

Méthodes disponibles dans VBS

Activate	GetRow (Page 701)	GetToolbarButtonCollection	Print
ActivateDynamic	GetRowCollection (Page 702)	GetValueColumn	SelectedStatisticArea
AttachDB	GetSelectedRow (Page 709)	GetValueColumnCollection	ShowColumnSelection
CalculateStatistic	GetSelectedRows (Page 710)	MoveToFirst	ShowHelp
CopyRows	GetStatusbarElement	MoveToLast	ShowPropertyDialog
DeactivateDynamic	GetStatusbarElementCollection	MoveToNext	ShowTagSelection
DetachDB	GetTimeColumn	MoveToPrevious	ShowTimeSelection
Edit	GetTimeColumnCollection	NextColumn	StartStopUpdate
Export	GetToolbarButton	PreviousColumn	

Propriétés disponibles dans VBS

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "timecolobj.TimeColumnName", le nom "TimeColumn" est ignoré : "timecolobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

AutoCompleteColumns	RTPersistence	TimeColumnAdd	ToolbarButtonId
AutoCompleteRows	RTPersistencePasswordLevel	TimeColumnAlign	ToolbarButtonIndex
AutoSelectionColors	RTPersistenceType	TimeColumnBackColor	ToolbarButtonLocked
AutoSelectionRectColor	SelectedCellColor	TimeColumnBeginTime	ToolbarButtonName
BackColor	SelectedCellForeColor	TimeColumnCaption	ToolbarButtonPasswordLevel

BorderColor	SelectedRowColor	TimeColumnCount	ToolbarButtonRemove
BorderWidth	SelectedRowForeColor	TimeColumnDateFormat	ToolbarButtonRename
Caption	SelectedTitleColor	TimeColumnEndTime	ToolbarButtonRepos
CellCut	SelectedTitleForeColor	TimeColumnForeColor	ToolbarButtonTooltipText
CellSpaceBottom	SelectionColoring	TimeColumnHideText	ToolbarButtonUserDefined
CellSpaceLeft	SelectionRect	TimeColumnHideTitleText	ToolbarButtonVisible
CellSpaceRight	SelectionRectColor	TimeColumnIndex	ToolbarShowTooltips
CellSpaceTop	SelectionRectWidth	TimeColumnLength	ToolbarUseBackColor
Closeable	SelectionType	TimeColumnMeasurePoints	ToolbarUseHotKeys
ColumnResize	ShowSortButton	TimeColumnName	ToolbarVisible
ColumnScrollbar	ShowSortIcon	TimeColumnRangeType	UseColumnBackColor
ColumnTitleAlign	ShowSortIndex	TimeColumnRemove	UseColumnForeColor
ColumnTitles	ShowTitle	TimeColumnRename	UseSelectedTitleColor
EnableEdit	Sizeable	TimeColumnRepos	UseTableColor2
ExportDirectoryChangeable	SkinName	TimeColumnShowDate	ValueColumnAdd
ExportDirectoryname	SortSequence	TimeColumnShowIcon	ValueColumnAlign
ExportFileExtension	StatusbarBackColor	TimeColumnShowTitleIcon	ValueColumnAutoPrecisions
ExportFilename	StatusbarElementAdd	TimeColumnSort	ValueColumnBackColor
ExportFilenameChangeable	StatusbarElementAutoSize	TimeColumnSortIndex	ValueColumnCaption
ExportFormatGuid	StatusbarElementCount	TimeColumnTimeFormat	ValueColumnCount
ExportFormatName	StatusbarElementIconId	TimeColumnTimeRangeBase	ValueColumnExponentialFormat
ExportParameters	StatusbarElementId	TimeColumnTimeRangeFactor	ValueColumnForeColor
ExportSelection	StatusbarElementIndex	TimeColumnUseValueColumnColors	ValueColumnHideText
ExportShowDialog	StatusbarElementName	TimeColumnVisible	ValueColumnHideTitleText
ExportXML	StatusbarElementRemove	TimeStepBase	ValueColumnIndex
Font	StatusbarElementRename	TimeStepFactor	ValueColumnLength
GridLineColor	StatusbarElementRepos	TitleColor	ValueColumnName
GridLineWidth	StatusbarElementText	TitleCut	ValueColumnPrecisions
HorizontalGridLines	StatusbarElementTooltipText	TitleDarkShadowColor	ValueColumnProvider
IconSpace	StatusbarElementUserDefined	TitleForeColor	ValueColumnProviderCLSID
LineColor	StatusbarElementVisible	TitleGridLineColor	ValueColumnRemove
LineWidth	StatusbarElementWidth	TitleLightShadowColor	ValueColumnRename
LoadDataImmediately	StatusbarFontColor	TitleSort	ValueColumnRepos
Moveable	StatusbarShowTooltips	TitleStyle	ValueColumnSelectTagName
Online	StatusbarText	ToolbarAlignment	ValueColumnShowIcon
PrintJobName	StatusbarUseBackColor	ToolbarBackColor	ValueColumnShowTitleIcon
RowCellCount (Page 525)	StatusbarVisible	ToolbarButtonActive	ValueColumnSort
RowCellText (Page 526)	TableColor	ToolbarButtonAdd	ValueColumnSortIndex
RowCount (Page 526)	TableColor2	ToolbarButtonBeginGroup	ValueColumnState

RowNumber (Page 526)	TableForeColor	ToolbarButtonClick	ValueColumnTagName
RowScrollbar	TableForeColor2	ToolbarButtonCount	ValueColumnTimeColumn
RowTitleAlign	TimeBase	ToolbarButtonEnabled	ValueColumnVisible
RowTitles	TimeColumnActualize	ToolbarButtonHotKey	VerticalGridLines

Exemple

Une colonne supplémentaire reliée à une variable d'archive est ajoutée dans un WinCC OnlineTableControl existant. Différentes propriétés sont configurées dans le script pour le Control et la colonne.

Condition

- Dans Graphics Designer, vous avez déjà inséré un "WinCC OnlineTableControl" portant le nom "Control1" dans une vue de process. Le Control est constitué d'une colonne heure et de trois colonnes de valeurs. On utilise pour cet exemple la vue "B_025_V7_Arch_TableControl.PDL" du projet de démonstration.
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré l'événement "Clic de souris" avec une action VBS et le script suivant.
- Vous avez déjà configuré une archive et une variable d'archive dans le projet. Ou vous utilisez le projet de démonstration dont est issue l'archive.

```
'VBS362
Sub OnClick(ByVal Item)
Dim objControl
Dim objTimeColumn
Dim objValueColumn
Set objControl = ScreenItems("Control1")
' Control wide spezification
objControl.ColumnResize = False
objControl.TimeBase = 1
objControl.TimeColumnTimeFormat = "HH:mm:ss tt"
objControl.TimeColumnLength = 20
' properties for Time column
Set objTimeColumn = objControl.GetTimeColumn("Time column 1")
objTimeColumn.DateFormat = "dd/MM/yy"
' properties for a new 4th value column with connection to archive tag "Trend_4"
Set objValueColumn = objControl.GetValueColumnCollection.AddItem("Trend 4")
objValueColumn.Caption = "Trend 4"
objValueColumn.Length = 10
objValueColumn.Align = 1
objValueColumn.Provider = 1
objValueColumn.TagName = "G_Archive\Trend_4"
objValueColumn.TimeColumn = "Time column 1"
End Sub
```

Remarque

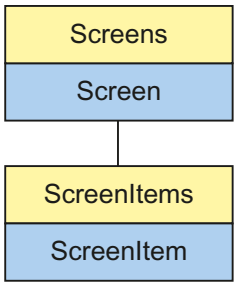
Pour plus d'exemples d'utilisation des propriétés et méthodes, référez-vous aux descriptions des méthodes Get des Controls et aux "Exemples de VBScript/Exemples dans WinCC/ Dynamisation des Controls".

Voir aussi

Contrôles (Page 224)

WinCC OnlineTrendControl

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC OnlineTrendControl" à partir de WinCC V7.0.

Désignation du type sous VBS

HMIOnlineTrendControl

Objets de liste disponibles

StatusbarElement (Page 237)	ToolbarButton (Page 239)	TrendWindow (Page 242)
TimeAxis (Page 237)	Tendance (Page 240)	ValueAxis (Page 243)

Méthodes disponibles dans VBS

Activate	GetTimeAxisCollection	MoveToFirst	ShowPropertyDialog
ActivateDynamic -Methode	GetToolbarButton (Page 722)	MoveToLast	ShowTagSelection
AttachDB-Methode	GetToolbarButtonCollection (Page 723)	MoveToNext	ShowTimeSelection

CalculateStatistic	GetTrend	MoveToPrevious	ShowTrendSelection
DeactivateDynamic	GetTrendCollection	NextTrend	StartStopUpdate
DetachDB	GetTrendWindow	OneToOneView	ZoomArea
Export	GetTrendWindowCollection	PreviousTrend	ZoomInOut
GetStatusbarElement	GetValueAxis	Print	ZoomInOutTime
GetStatusbarElementCollection	GetValueAxisCollection	ShowHelp	ZoomInOutValues
GetTimeAxis	MoveAxis	ShowPercentageAxis	ZoomMove

Propriétés disponibles dans VBS

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "trendobj.Trendname", le nom "Trend" est ignoré : "trendobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

BackColor	StatusbarElementRepos	ToolbarButtonRemove	TrendValueUnit
BorderColor	StatusbarElementText	ToolbarButtonRename	TrendVisible
BorderWidth	StatusbarElementTooltipText	ToolbarButtonRepos	TrendWindowAdd
Caption	StatusbarElementUserDefined	ToolbarButtonTooltipText	TrendWindowCoarseGrid
Closeable	StatusbarElementVisible	ToolbarButtonUserDefined	TrendWindowCoarseGridColor
ConnectTrendWindows	StatusbarElementWidth	ToolbarButtonVisible	TrendWindowCount
ExportDirectoryChangeable	StatusbarFontColor	ToolbarShowTooltips	TrendWindowFineGrid
ExportDirectoryname	StatusbarShowTooltips	ToolbarUseBackColor	TrendWindowFineGridColor
ExportFileExtension	StatusbarText	ToolbarUseHotKeys	TrendWindowForegroundTrendGrid
ExportFilename	StatusbarUseBackColor	ToolbarVisible	TrendWindowGridInTrendColor
ExportFilenameChangeable	StatusbarVisible	TrendAdd	TrendWindowHorizontalGrid
ExportFormatGuid	TimeAxisActualize	TrendAutoRangeBeginTagName	TrendWindowIndex
ExportFormatName	TimeAxisAdd	TrendAutoRangeBeginValue	TrendWindowName
ExportParameters	TimeAxisAlign	TrendAutoRangeEndTagName	TrendWindowRemove
ExportSelection	TimeAxisBeginTime	TrendAutoRangeEndValue	TrendWindowRename
ExportShowDialog	TimeAxisColor	TrendAutoRangeSource	TrendWindowRepos
ExportXML	TimeAxisCount	TrendColor	TrendWindowRulerColor
Font	TimeAxisDateFormat	TrendCount	TrendWindowRulerLayer

GraphDirection	TimeAxisEndTime	TrendExtendedColorSet	TrendWindowRulerStyle
LineColor	TimeAxisIndex	TrendFill	TrendWindowRulerWidth
LineWidth	TimeAxisInTrendColor	TrendFillColor	TrendWindowSpacePortion
LoadDataImmediately	TimeAxisLabel	TrendIndex	TrendWindowStatisticRulerColor
Moveable	TimeAxisMeasurePoints	TrendLabel	TrendWindowStatisticRulerStyle
Online	TimeAxisName	TrendLineStyle	TrendWindowStatisticRuler Width
PercentageAxis	TimeAxisRangeType	TrendLineType	TrendWindowVerticalGrid
PercentageAxisAlign	TimeAxisRemove	TrendLineWidth	TrendWindowVisible
PercentageAxisColor	TimeAxisRename	TrendLowerLimit	UseTrendNameAsLabel
PrintJobName	TimeAxisRepos	TrendLowerLimitColor	ValueAxisAdd
RTPersistence	TimeAxisShowDate	TrendLowerLimitColoring	ValueAxisAlign
RTPersistencePasswordLevel	TimeAxisTimeFormat	TrendName	ValueAxisAutoPrecisions
RTPersistenceType	TimeAxisTimeRangeBase	TrendPointColor	ValueAxisAutoRange
ShowRuler	TimeAxisTimeRangeFactor	TrendPointStyle	ValueAxisBeginValue
ShowRulerInAxis	TimeAxisTrendWindow	TrendPointWidth	ValueAxisColor
ShowScrollbars	TimeAxisVisible	TrendProvider	ValueAxisCount
ShowStatisticRuler	TimeBase	TrendProviderCLSID	ValueAxisEndValue
ShowTitle	ToolbarAlignment	TrendRemove	ValueAxisExponentialFormat
ShowTrendIcon	ToolbarBackColor	TrendRename	ValueAxisIndex
Sizeable	ToolbarButtonActive	TrendRepos	ValueAxisInTrendColor
SkinName	ToolbarButtonAdd	TrendSelectTagName	ValueAxisLabel
StatusbarBackColor	ToolbarButtonBeginGroup	TrendTagName	ValueAxisName
StatusbarElementAdd	ToolbarButtonClick	TrendTimeAxis	ValueAxisPrecisions
StatusbarElementAutoSize	ToolbarButtonCount	TrendTrendWindow	ValueAxisRemove
StatusbarElementCount	ToolbarButtonEnabled	TrendUncertainColor	ValueAxisRename
StatusbarElementIconId	ToolbarButtonHotKey	TrendUncertainColoring	ValueAxisRepos
StatusbarElementId	ToolbarButtonId	TrendUpperLimit	ValueAxisScalingType
StatusbarElementIndex	ToolbarButtonIndex	TrendUpperLimitColor	ValueAxisTrendWindow
StatusbarElementName	ToolbarButtonLocked	TrendUpperLimitColoring	ValueAxisVisible
StatusbarElementRemove	ToolbarButtonName	TrendValueAlignment	
StatusbarElementRename	ToolbarButtonPasswordLevel	TrendValueAxis	

Exemple

Dans un WinCC OnlineTrendControl, trois courbes reliées à une variable d'archive sont affichées. Différentes propriétés sont configurées dans le script pour les courbes.

Condition

- Dans Graphics Designer, un "WinCC OnlineTrendControl" portant le nom "Control1" est inséré dans une vue de process.
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré l'événement "Clic de souris" avec une action VBS et le script suivant.
- Vous avez déjà configuré une archive et une variable d'archive dans le projet. Ou vous utilisez le projet de démonstration dont est issue l'archive.

```
'VBS361
Sub OnClick(ByVal Item)
Dim objTrendControl
Dim objTrendWindow
Dim objTimeAxis
Dim objValueAxis
Dim objTrend
'create reference to TrendControl
Set objTrendControl = ScreenItems("Control1")
'create reference to new window, time and value axis
Set objTrendWindow = objTrendControl.GetTrendWindowCollection.AddItem("myWindow")
Set objTimeAxis = objTrendControl.GetTimeAxisCollection.AddItem("myTimeAxis")
Set objValueAxis = objTrendControl.GetValueAxisCollection.AddItem("myValueAxis")
'assign time and value axis to the window
objTimeAxis.TrendWindow = objTrendWindow.Name
objValueAxis.TrendWindow = objTrendWindow.Name
' assign properties to trendwindow
objTrendWindow.HorizontalGrid = False
' add new trend and assign properties
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myTrend1")
objTrend.Provider = 1
objTrend.TagName = "G_Archive\Trend_1"
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValueAxis.Name
objTrend.Color = RGB(255,0,0)
objTrend.PointStyle = 0
'add new trend and assign propertys
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myTrend2")
objTrend.Provider = 1
objTrend.TagName = "G_Archive\Trend_2"
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValueAxis.Name
objTrend.Color = RGB(0,255,0)
objTrend.LineWidth = 3
'add new trend and assign propertys
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myTrend3")
objTrend.Provider = 1
objTrend.TagName = "G_Archive\Trend_3"
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValueAxis.Name
objTrend.Color = RGB(0,0,255)
objTrend.LineType = 2
```

```
End Sub
```

Remarque

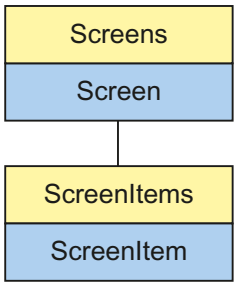
Pour plus d'exemples d'utilisation des propriétés et méthodes, référez-vous aux descriptions des méthodes Get des Controls et aux "Exemples de VBScript/Exemples dans WinCC/ Dynamisation des Controls".

Voir aussi

Contrôles (Page 224)

WinCC Push Button Control

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Push Button Control"

Désignation du type sous VBS

HMIButton

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 17 Pixel vers la droite :

```
'VBS61
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +17
```

Remarque

Les événements KeyDown, KeyUp et KeyPress ne peuvent pas être déclenchés par VBS. Si vous devez dynamiser des contrôles avec VBS, aucun des paramètres ne doit être déclaré avec ByRef.

Remarques sur le traitement des erreurs

Les boutons et boutons-poussoirs sont représentés dans le modèle d'objet sur un seul type "HMIButton". Etant donné que ces objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime) via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

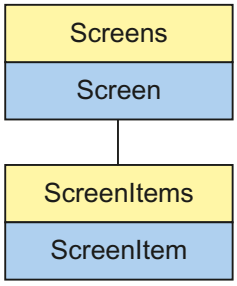
```
'VBS62
Dim objScreenItem
On Error Resume Next    'Activation of errorhandling
For Each objScreenItem In ScreenItems
If objScreenItem.Type = "HMIButton" Then
'
'=== Property "Text" available only for Standard-Button
objScreenItem.Text = "Windows"
If 0 <> Err.Number Then
HMIRuntime.Trace objScreenItem.ObjectName & ": no Windows-Button" & vbCrLf
Err.Clear    'Delete error message
    End If
'
'=== Property "Caption" available only for PushButton
objScreenItem.Caption = "Push"
If 0 <> Err.Number Then
HMIRuntime.Trace objScreenItem.ObjectName & ": no Control" & vbCrLf
    Err.Clear
    End If
    End If
Next
On Error Goto 0  'Deactivation of errorhandling
```

Voir aussi

Propriétés (Page 292)
Propriété FontName (Page 406)
Méthode Activate (Page 682)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Transparent (Page 616)
Propriété Top (Page 614)
Propriété PictureUnselected (Page 511)
Propriété PictureSelected (Page 510)
Propriété Parent (Page 501)
Propriété Outline (Page 500)
Propriété ObjectName (Page 485)
Propriété Object (Page 484)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété FrameWidth (Page 411)
Propriété FrameColorUp (Page 410)
Propriété FrameColorDown (Page 410)
Propriété ForeColor (Page 408)
Propriété FontUnderline (Page 407)
Propriété FontStrikeThru (Page 407)
Propriété FontSize (Page 407)
Propriété FontItalic (Page 405)
Propriété Font (avant WinCC V7) (Page 404)
Propriété FontBold (Page 405)
Propriété FocusRect (Page 403)
Propriété Enabled (Page 382)
Propriété Caption (Page 339)
Propriété BackColor (Page 313)
Propriété AutoSize (Page 311)

WinCC RulerControl

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC RulerControl" à partir de WinCC V7.0.

Désignation du type sous VBS

HMIRulerControl

Objets de liste disponibles

Row (Page 233)	StatisticResultColumn (Page 236)
RulerBlock (Page 234)	StatusbarElement (Page 237)
RulerColumn (Page 234)	ToolbarButton (Page 239)
StatisticAreaColumn (Page 235)	

Méthodes disponibles dans VBS

Activate	GetRulerBlock	GetStatisticAreaColumn	GetToolbarButton
ActivateDynamic	GetRulerBlockCollection	GetStatisticAreaColumnCollection	GetToolbarButtonCollection
DeactivateDynamic	GetRulerColumn	GetStatisticResultColumn	ShowHelp
Export	GetRulerColumnCollection	GetStatisticResultColumnCollection	ShowPropertyDialog
GetRow (Page 701)	GetSelectedRow (Page 709)	GetStatusbarElement	
GetRowCollection (Page 702)	GetSelectedRows (Page 710)	GetStatusbarElementCollection	

Propriétés disponibles dans VBS

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "rulercolobj.ColumnName", le nom "Column" est ignoré : "rulercolobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

AutoCompleteColumns	ColumnScrollbar	SelectedRowForeColor	TableColor2
AutoCompleteRows	ColumnSort	SelectedTitleColor	TableForeColor
AutoPositon	ColumnSortIndex	SelectedTitleForeColor	TableForeColor2
AutoSelectionColors	ColumnTitleAlign	SelectionColoring	TitleColor
AutoSelectionRectColor	ColumnTitles	SelectionRect	TitleCut
AutoShow	ColumnVisible	SelectionRectColor	TitleDarkShadowColor
BackColor	ExportDirectoryChangeable	SelectionRectWidth	TitleForeColor
BlockAlign	ExportDirectoryname	SelectionType	TitleGridLineColor
BlockAutoPrecisions	ExportFileExtension	ShareSpaceWithSourceControl	TitleLightShadowColor
BlockCaption	ExportFilename	ShowSortButton	TitleSort
BlockCount	ExportFilenameChangeable	ShowSortIcon	TitleStyle
BlockDateFormat	ExportFormatGuid	ShowSortIndex	ToolbarAlignment
BlockExponentialFormat	ExportFormatName	ShowTitle	ToolbarBackColor
BlockHideText	ExportParameters	Sizeable	ToolbarButtonActive
BlockHideTitleText	ExportSelection	SkinName	ToolbarButtonAdd
BlockID	ExportShowDialog	SortSequence	ToolbarButtonBeginGroup
BlockIndex	ExportXML	SourceControl	ToolbarButtonClick
BlockLength	Font	SourceControlType	ToolbarButtonCount
BlockName	GridLineColor	StatusbarBackColor	ToolbarButtonEnabled
BlockPrecisions	GridLineWidth	StatusbarElementAdd	ToolbarButtonHotKey
BlockShowDate	HorizontalGridLines	StatusbarElementAutoSize	ToolbarButtonId
BlockShowIcon	IconSpace	StatusbarElementCount	ToolbarButtonIndex
BlockShowTitleIcon	LineColor	StatusbarElementIconId	ToolbarButtonLocked
BlockTimeFormat	LineWidth	StatusbarElementId	ToolbarButtonName
BlockUseSourceFormat	Moveable	StatusbarElementIndex	ToolbarButtonPasswordLevel
BorderColor	PrintJobName	StatusbarElementName	ToolbarButtonRemove
BorderWidth	RowCellCount (Page 525)	StatusbarElementRemove	ToolbarButtonRename
Caption	RowCellText (Page 526)	StatusbarElementRename	ToolbarButtonRepos
CellCut	RowCount (Page 526)	StatusbarElementRepos	ToolbarButtonTooltipText
CellSpaceBottom	RowNumber (Page 526)	StatusbarElementText	ToolbarButtonUserDefined
CellSpaceLeft	RowScrollbar	StatusbarElementTooltipText	ToolbarButtonVisible
CellSpaceRight	RowTitleAlign	StatusbarElementUserDefined	ToolbarShowTooltips
CellSpaceTop	RowTitles	StatusbarElementVisible	ToolbarUseBackColor
Closeable	RTPersistence	StatusbarElementWidth	ToolbarUseHotKeys
ColumnAdd	RTPersistencePasswordLevel	StatusbarFontColor	ToolbarVisible
ColumnCount	RTPersistenceType	StatusbarShowTooltips	UseSelectedTitleColor

ColumnIndex	RulerType	StatusbarText	UseSourceBackColors
ColumnName	SelectedCellColor	StatusbarUseBackColor	UseSourceForeColors
ColumnRemove	SelectedCellForeColor	StatusbarVisible	UseTableColor2
ColumnRepos	SelectedRowColor	TableColor	VerticalGridLines
ColumnResize			

Exemple

Un WinCC RulerControl supplémentaire est ajouté dans une vue contenant un WinCC OnlineTableControl existant. Le RulerControl contient une fenêtre statistiques qui affiche les colonnes "Minimum", "Maximum" et "Moyenne". Les valeurs statistiques sont affichées pour les lignes sélectionnées du OnlineTableControl.

Condition

- Dans Graphics Designer, vous avez déjà inséré un "WinCC OnlineTableControl" portant le nom "Control1" dans une vue de process. Le Control est relié à la variable d'archive ou à la variable de process. On utilise pour cet exemple la vue "B_025_V7_Arch_TableControl.PDL" du projet de démonstration.
- Un "WinCC RulerControl" supplémentaire portant le nom "Control2" est inséré dans la vue.
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré l'événement "Clic de souris" avec une action VBS et le script suivant.
- En Runtime, vous avez sélectionné quelques lignes dans OnlineTableControl.

```
'VBS364
Sub OnClick(ByVal Item)
Dim objRulerControl
Dim objTableControl
Dim objstatColumn
Dim rows

Set objRulerControl = ScreenItems("Control2")
' Use Statistic-window
objRulerControl.RulerType = 2
objRulerControl.SourceControl = "Control1"
' In Statistic-window only columns "Name", "MinValue", MaxValue" and "Average" are shown
Set objstatColumn = objRulerControl.GetStatisticResultColumnCollection
objstatColumn.RemoveItem(4)
objstatColumn.RemoveItem(5)
objstatColumn.RemoveItem(6)
' Get the selected rows of tablecontrol and calculate statistic
Set objTableControl = ScreenItems("Control1")
Set rows = objTableControl.SelectAll
objTableControl.CalculateStatistic
End Sub
```

Remarque

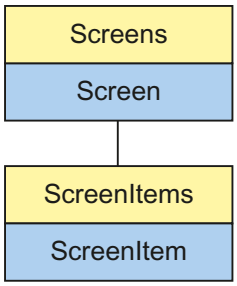
Pour plus d'exemples d'utilisation des propriétés et méthodes, référez-vous aux descriptions des méthodes Get des Controls et aux "Exemples de VBScript/Exemples dans WinCC/ Dynamisation des Controls".

Voir aussi

Contrôles (Page 224)

WinCC Slider Control

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Slider Control"

Désignation du type sous VBS

HMISlider

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 19 Pixel vers la droite :

```
'VBS63
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +19
```

Remarques sur le traitement des erreurs

Slider et WinCC Slider Controls sont représentés dans un modèle d'objet sur un seul type "HMISlider". Etant donné que ces objets ont cependant des propriétés différentes, il faudrait que vous demandiez s'il existe des propriétés (détermination dynamique du type en Runtime)

via un traitement exceptionnel. Le traitement exceptionnel est activé par l'instruction suivante pour la procédure respective :

```
On Error Resume Next
```

L'instruction fait en sorte que VBScript Engine effectue l'instruction consécutive suite à une erreur d'exécution.

Dans la ligne consécutive, le code d'erreur peut être vérifié par l'objet Err. L'instruction suivante est utilisée pour désactiver le traitement des erreurs d'exécution dans le Script :

```
On Error Goto 0
```

Le traitement des erreurs se réfère toujours au niveau de procédure. Si un Script déclenche une erreur dans une procédure, VBScript vérifie si le traitement des erreurs est implémenté sur ce niveau. Si ce n'est pas le cas, le contrôle passe au niveau supérieur (à la procédure appelant). Si là non plus il n'y a pas de traitement des erreurs, le contrôle est à nouveau transmis à un niveau supérieur. Cela continue jusqu'à ce le niveau le plus élevé des modules soit atteint ou bien jusqu'à ce que le code permettant le traitement des erreurs d'exécution soit trouvé. Si le traitement des erreurs en Runtime n'est pas activé, le contrôle est transmis au niveau principal, au traitement interne des erreurs d'exécution VBScript. Ce dernier affiche un dialogue d'erreur et interrompt le Script.

L'instruction "On Error Resume Next" peut être intégrée sur tous les niveaux (donc également dans les procédures). L'utilisation de la fonction de traitement des erreurs permet en principe de constater si l'utilisateur emploie vraiment le type d'implémentation voulu.

Elle garantit en outre que l'exécution ne sera pas interrompue en cas d'accès non correct à l'objet.

Exemple de traitement des erreurs

```
Sub OnClick(Byval Item)
'VBS193
Dim ScreenItem
' activating error handling:
On Error Resume Next
For Each ScreenItem In ScreenItems
If ScreenItem.Type = "HMISlider" Then
'=== Property "BevelColorUp" only exists for a WinCC Slider Control
ScreenItem.BevelColorUp = 1
If (Err.Number <> 0) Then
HMIRuntime.Trace(ScreenItem.ObjectName + ": no Windows-Slider" + vbCrlf)
' delete error message
    Err.Clear
    End If
'=== Property "BorderStyle" only exists for a Windows-Slider
```

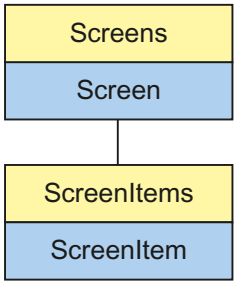
```
ScreenItem.BorderStyle = 1
If (Err.Number <> 0) Then
HMIRuntime.Trace(ScreenItem.ObjectName + ": no WinCC Slider Control" + vbCrlf)
    Err.Clear
    End If
    End If
Next
On Error GoTo 0  ' deactivating error handling
End Sub
```

Voir aussi

Propriété PictureThumb (Page 511)
Propriété BarFillColor (Page 318)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété WithLabels (Page 672)
Propriété WithAxes (Page 672)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété TickStyle (Page 574)
Propriété ThumbBackColor (Page 571)
Propriété ShowThumb (Page 550)
Propriété ShowPosition (Page 547)
Propriété ShowBar (Page 545)
Propriété RangeMin (Page 521)
Propriété RangeMax (Page 520)
Propriété Position (Page 513)
Propriété PictureBack (Page 509)
Propriété Parent (Page 501)
Propriété OuterBevelWidth (Page 500)
Propriété OuterBevelStyle (Page 499)
Propriété ObjectName (Page 485)
Propriété Object (Page 484)
Propriété LocaleID (Page 455)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété LabelColor (Page 430)
Propriété InnerBevelWidth (Page 426)
Propriété InnerBevelStyle (Page 426)
Propriété InnerBevelOffset (Page 425)
Propriété Height (Page 417)
Propriété ForeColor (Page 408)
Propriété FontPosition (Page 406)
Propriété Font (avant WinCC V7) (Page 404)
Propriété FocusWidth (Page 403)
Propriété FocusColor (Page 403)

WinCC UserArchiveControl

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC UserArchiveControl" à partir de WinCC V7.0.

Désignation du type sous VBS

HMIUserArchiveControl

Objets de liste disponibles

Column (Page 228)	StatusbarElement (Page 237)
Row (Page 233)	ToolbarButton (Page 239)

Méthodes disponibles dans VBS

Activate	GetRow (Page 701)	MoveToFirst	ServerImport
ActivateDynamic	GetRowCollection (Page 702)	MoveToLast	ShowHelp
CopyRows	GetSelectedRow (Page 709)	MoveToNext	ShowPropertyDialog
CutRows	GetSelectedRows (Page 710)	MoveToPrevious	ShowSelectArchive
DeactivateDynamic	GetStatusbarElement	PasteRows	ShowSelection
CutRows	GetStatusbarElementCollection	Print	ShowSelectTimeBase
Export	GetToolbarButton	ReadTags	ShowSort
GetColumn	GetToolbarButtonCollection	ServerExport	WriteTags
GetColumnCollection			

Propriétés disponibles dans VBS

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste. P. ex. si vous utilisez "colobj.ColumnName", le nom "Column" est ignoré : "colobj.Name".

Attention, les propriétés disponibles pour les Controls WinCC doivent s'appliquer de manière fonctionnelle comme méthodes. Ces propriétés se distinguent par leur nom, p.ex. "Add", "Remove" ou "Rename".

ArchiveName	ColumnShowIcon	RowTitles	StatusbarUseBackColor
ArchiveType	ColumnShowTitleIcon	RTPersistence	StatusbarVisible
AutoCompleteColumns	ColumnSort	RTPersistencePasswordLevel	TableColor
AutoCompleteRows	ColumnSortIndex	RTPersistenceType	TableColor2
AutoSelectionColors	ColumnStartValue	SelectArchiveName	TableForeColor
AutoSelectionRectColor	ColumnStringLength	SelectedCellColor	TableForeColor2
BackColor	ColumnTimeFormat	SelectedCellForeColor	TimeBase
BorderColor	ColumnTitleAlign	SelectedRowColor	TitleColor
BorderWidth	ColumnTitles	SelectedRowForeColor	TitleCut
Caption	ColumnType	SelectedTitleColor	TitleDarkShadowColor
CellCut	ColumnVisible	SelectedTitleForeColor	TitleForeColor
CellSpaceBottom	ColumnWriteAccess	SelectionColoring	TitleGridLineColor
CellSpaceLeft	EnableDelete	SelectionRect	TitleLightShadowColor
CellSpaceRight	EnableEdit	SelectionRectColor	TitleSort
CellSpaceTop	EnableInsert	SelectionRectWidth	TitleStyle
Closeable	ExportDirectoryChangeable	SelectionType	ToolbarAlignment
ColumnAlias	ExportDirectoryname	ShowSortButton	ToolbarBackColor
ColumnAlign	ExportFileExtension	ShowSortIcon	ToolbarButtonActive
ColumnAutoPrecisions	ExportFilename	ShowSortIndex	ToolbarButtonAdd
ColumnCaption	ExportFilenameChangeable	ShowTitle	ToolbarButtonBeginGroup
ColumnCount	ExportFormatGuid	Sizeable	ToolbarButtonClick
ColumnDateFormat	ExportFormatName	SkinName	ToolbarButtonCount
ColumnDMVarName	ExportParameters	SortSequence	ToolbarButtonEnabled
ColumnExponentialFormat	ExportSelection	StatusbarBackColor	ToolbarButtonHotKey
ColumnFlagNotNull	ExportShowDialog	StatusbarElementAdd	ToolbarButtonId
ColumnFlagUnique	ExportXML	StatusbarElementAutoSize	ToolbarButtonIndex
ColumnHideText	FilterSQL	StatusbarElementCount	ToolbarButtonLocked
ColumnHideTitleText	Font	StatusbarElementIconId	ToolbarButtonName
ColumnIndex	GridLineColor	StatusbarElementId	ToolbarButtonPasswordLevel
ColumnLeadingZeros	GridLineWidth	StatusbarElementIndex	ToolbarButtonRemove
ColumnLength	HorizontalGridLines	StatusbarElementName	ToolbarButtonRename
ColumnMaxValue	IconSpace	StatusbarElementRemove	ToolbarButtonRepos
ColumnMinValue	LineColor	StatusbarElementRename	ToolbarButtonTooltipText
ColumnName	LineWidth	StatusbarElementRepos	ToolbarButtonUserDefined
ColumnPosition (Page 359)	Moveable	StatusbarElementText	ToolbarButtonVisible
ColumnPrecisions	PrintJobName	StatusbarElementTooltipText	ToolbarShowTooltips
ColumnReadAccess	RowCellCount (Page 525)	StatusbarElementUserDefined	ToolbarUseBackColor

ColumnReadonly	RowCellText (Page 526)	StatusbarElementVisible	ToolbarUseHotKeys
ColumnRepos	RowCount (Page 526)	StatusbarElementWidth	ToolbarVisible
ColumnResize	RowNumber (Page 526)	StatusbarFontColor	UseSelectedTitleColor
ColumnScrollbar	RowScrollbar	StatusbarShowTooltips	UseTableColor2
ColumnShowDate	RowTitleAlign	StatusbarText	VerticalGridLines

Exemple

Une archive utilisateur est affichée dans un WinCC UserArchiveControl.

Les actions suivantes sont lancées via le script :

- Sélection des données
- Exporter les données
- Impression des tables

Condition

- Dans Graphics Designer, un "WinCC UserArchiveControl" portant le nom "Control1" est inséré dans une vue de process.
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré l'événement "Clic de souris" avec une action VBS et le script suivant.
- Vous avez déjà configuré une archive utilisateur dans le projet. Ou vous utilisez le projet de démonstration dont est issue l'archive utilisateur utilisée.

```
'VBS365
Sub OnClick(ByVal Item)
Dim objUAControl
Dim objColumn
Dim coll
Dim field
' create reference to UserArchivControl
Set objUAControl = ScreenItems("Control1")
' Select user archiv and general column properties
objUAControl.SelectArchiveName = True
objUAControl.ColumnResize = False
objUAControl.ColumnTitleAlign = 1
' properties for ID column
Set objColumn = objUAControl.GetColumn("ID")
objColumn.Length = 2
objColumn.Align = 0
' Select data
objUAControl.FilterSQL = "ID >=3"
'export the content as a CSV-file in the "ua" directory of the project folder
objUAControl.ServerExport
' print the control
objUAControl.PrintJobName = "UserArchiveControl - Table"
objUAControl.Print
End Sub
```

Remarque

Pour plus d'exemples d'utilisation des propriétés et méthodes, référez-vous aux descriptions des méthodes Get des Controls et aux "Exemples de VBScript/Exemples dans WinCC/ Dynamisation des Controls".

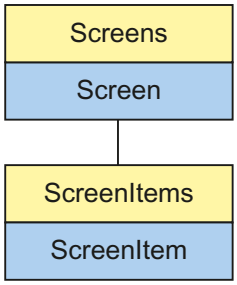
Voir aussi

Contrôles (Page 224)

Contrôles avant WinCC V7

WinCC Alarm Control (avant WinCC V7)

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Alarm Control"

Désignation du type sous VBS

HMIMessageView

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 10 Pixel vers la droite :

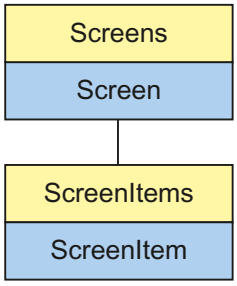
```
'VBS54
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left 10
```

Voir aussi

Propriété ProjectPath (Page 517)
Propriété BackColor (Page 313)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété WindowType (Page 671)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolbarButtons (Page 610)
Propriété Titleline (Page 599)
Propriété TitleCut (avant WinCC V7) (Page 598)
Propriété StatusbarPanes (Page 565)
Propriété ServerNames (avant WinCC V7) (Page 544)
Propriété SelectionType (avant WinCC V7) (Page 540)
Propriété SelectionRectWidth (avant WinCC V7) (Page 540)
Propriété SelectionRectColor (avant WinCC V7) (Page 539)
Propriété SelectionMode (Page 538)
Propriété PersistentRTPermission (Page 506)
Propriété PersistentRTCSPermission (Page 506)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Object (Page 484)
Propriété MsgFilterSQL (avant WinCC V7) (Page 481)
Propriété MsgCtrlFlags (Page 481)
Propriété LineTitle (Page 454)
Propriété LineHeight (Page 453)
Propriété LineFont (Page 453)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété HeaderSort (Page 417)
Propriété GridLineVert (Page 415)
Propriété GridLineHorz (Page 413)
Propriété Font (avant WinCC V7) (Page 404)
Propriété Enabled (Page 382)
Propriété ColWidth (Page 365)

WinCC Function Trend Control (avant WinCC V7)

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Function Trend Control".

Désignation du type sous VBS

HMIFunctionTrendView

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 13 Pixel vers la droite :

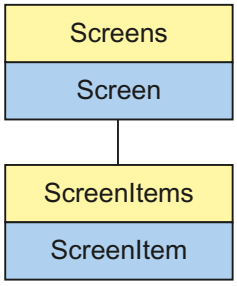
```
'VBS57
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +13
```

Voir aussi

Propriété Top (Page 614)
Propriété ScalingTypeY (Page 532)
Objet Layer (Page 127)
Propriété DesiredCurveSourceUAArchive (Page 378)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété UpperLimitValue (Page 646)
Propriété UpperLimit (Page 645)
Propriété UpperLimitColor (Page 646)
Propriété Type (Page 638)
Propriété ToolbarHotKeys (Page 611)
Propriété ToolbarButtons (Page 610)
Propriété ToolbarAlignment (avant WinCC V7) (Page 602)
Propriété Titleline (Page 599)
Propriété TimeZone (Page 597)
Propriété TimeAxisX (Page 581)
Propriété TagProviderClsid (Page 569)
Propriété SourceUAColumnY (Page 559)
Propriété SourceUAColumnX (Page 558)
Propriété SourceUAArchiveStartID (Page 558)
Propriété SourceUAArchive (Page 557)
Propriété SourceTimeRange (Page 557)
Propriété SourceTagProviderDataY (Page 557)
Propriété SourceTagProviderDataX (Page 556)
Propriété SourceTagNameY (Page 556)
Propriété SourceTagNameX (Page 556)
Propriété SourceNumberOfValues (Page 555)
Propriété SourceNumberOfUAValues (Page 555)
Propriété SourceEndTime (Page 555)
Propriété SourceBeginTime (Page 554)
Propriété ShowValuesExponentialY (Page 551)
Propriété ShowValuesExponentialX (Page 551)
Propriété ShowRulerImmediately (Page 547)
Propriété ScalingTypeX (Page 532)
Propriété RulerPrecisionY (Page 529)

WinCC Online Table Control (avant WinCC V7)

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Online Table Control"

Désignation du type sous VBS

HMITableView

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 15 Pixel vers la droite :

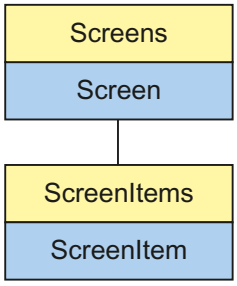
```
'VBS59
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +15
```

Voir aussi

Propriété TimeOverlap (Page 593)
Propriété ItemVisible (Page 429)
PrintBackgroundColor-propriété (Page 516)
Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Variable (Page 666)
Propriété ValueColumnAlignment (Page 659)
Propriété UpperLimitValue (Page 646)
Propriété UpperLimit (Page 645)
Propriété UpperLimitColor (Page 646)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolbarHotKeys (Page 611)
Propriété Toolbar (Page 601)
Propriété ToolbarButtons (Page 610)
Propriété ToolbarAlignment (avant WinCC V7) (Page 602)
Propriété Titleline (Page 599)
Propriété TimeZone (Page 597)
Propriété TimeRangeFactor (Page 594)
Propriété TimeRange (Page 594)
Propriété TimeRangeBase (Page 594)
Propriété TimeOverlapColor (Page 593)
Propriété TimeJump (Page 592)
Propriété TimeJumpColor (Page 592)
Propriété TimeFormat (Page 591)
Propriété TimeColumnAlignment (Page 583)
Propriété Statusbar (Page 560)
Propriété PrintJob (Page 516)
Propriété Precisions (Page 514)
Propriété PersistentRTPermission (Page 506)
Propriété PersistentRT (Page 505)
Propriété PersistentRTCSPermission (Page 506)
Propriété PersistentRTCS (Page 505)
Propriété Parent (Page 501)
Propriété Online (avant WinCC V7) (Page 489)

WinCC Online Trend Control (avant WinCC V7)

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "WinCC Online Trend Control"

Désignation du type sous VBS

HMITrendView

Utilisation

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 16 Pixel vers la droite :

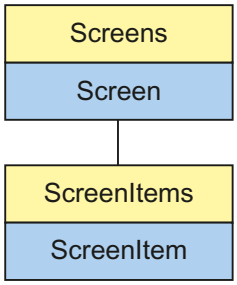
```
'VBS60
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left +16
```

Voir aussi

Propriétés (Page 292)
Propriété TimeAxis (Page 574)
Propriété LowerLimitColor (Page 460)
Propriété Caption (Page 339)
Méthode Activate (Page 682)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Contrôles (Page 224)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété UpperLimitValue (Page 646)
Propriété UpperLimit (Page 645)
Propriété UpperLimitColor (Page 646)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolbarHotKeys (Page 611)
Propriété Toolbar (Page 601)
Propriété ToolbarButtons (Page 610)
Propriété ToolbarAlignment (avant WinCC V7) (Page 602)
Propriété Titleline (Page 599)
Propriété TimeZone (Page 597)
Propriété TimeRangeFactor (Page 594)
Propriété TimeRange (Page 594)
Propriété TimeRangeBase (Page 594)
Propriété TimeOverlap (Page 593)
Propriété TimeOverlapColor (Page 593)
Propriété TimeJump (Page 592)
Propriété TimeJumpColor (Page 592)
Propriété TimeAxisFormat (Page 577)
Propriété TagName (Page 568)
Propriété Statusbar (Page 560)
Propriété ShowRulerImmediately (Page 547)
Propriété ServerData (Page 542)
Propriété RulerPrecisions (Page 528)
Propriété Replacement (Page 523)
Propriété ReplacementColor (Page 523)
Propriété RelayCurves (Page 522)
Propriété ProviderClsid (Page 517)
Propriété PrintJob (Page 516)
Propriété Precisions (Page 514)

1.14.3.7 Objet utilisateur

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Objet utilisateur"

Désignation du type sous VBS

HMIScreenModule

Utilisation

Les propriétés personnalisées d'un objet utilisateur doivent être accédées dans VBS via les noms d'attribut. Intellisense est exclusivement valide pour l'objet utilisateur en entier.

Le nom d'attribut peut être trouvé sous les propriétés des propriétés posées vers l'extérieur (clic droit de la souris sur la propriété) et peut y être modifié.

Dans l'exemple suivant, l'objet appelé "CustomizedObject1" est décalé de 10 Pixel vers la droite :

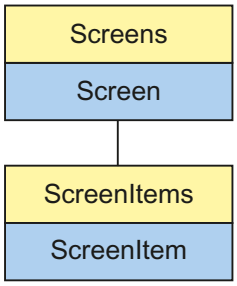
```
'VBS65
Dim objCustomObject
Set objCustomObject = ScreenItems("CustomizedObject1")
objCustomObject.Left = objCustomObject.Left + 10
```

Voir aussi

Méthode Activate (Page 682)
Propriétés (Page 292)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Type d'objet de l'objet ScreenItem (Page 150)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété Enabled (Page 382)

1.14.3.8 Groupe

Description



Type d'objet de l'objet ScreenItem. Représente l'objet graphique "Groupe"

Désignation du type sous VBS

HMIGroup

Utilisation

Dans l'exemple suivant, l'objet appelé "Group1" est décalé de 10 Pixel vers la droite :

```
'VBS66
Dim objGroup
Set objGroup = ScreenItems("Group1")
objGroup.Left = objGroup.Left + 10
```

Voir aussi

Propriétés (Page 292)
Méthode Activate (Page 682)
Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Type d'objet de l'objet ScreenItem (Page 150)
Propriété Width (Page 669)
Propriété Visible (Page 667)
Propriété Type (Page 638)
Propriété Top (Page 614)
Propriété ToolTipText (Page 613)
Propriété Parent (Page 501)
Propriété ObjectName (Page 485)
Propriété Left (Page 449)
Objet Layer (Page 127)
Propriété Height (Page 417)
Propriété Enabled (Page 382)

1.14.4 Propriétés

1.14.4.1 Propriétés

Liste récapitulative

Les propriétés des divers objets vous permettent de modifier ponctuellement des objets graphiques et des variables dans Runtime, par exemple déconnecter un organe de commande sur clic de souris, ou déclencher un changement de couleur au cas où la valeur d'une variable serait modifiée.

Vous pouvez accéder aux propriétés pour les objets graphiques via la syntaxe suivante:

```
'VBS191
Dim obj
Set obj = ScreenItems("object1")
obj.property = Valeur
```

Dans l'exemple suivant, l'objet appelé "Control1" est décalé de 10 Pixel vers la droite:

```
'VBS192
Dim obj
Set obj = ScreenItems("control1")
obj.Left = obj.Left + 10
```

1.14.4.2 A

Aa - Ad

Propriété AccessPath

Description

Affiche le chemin d'enregistrement (la hiérarchie étant spécifiée) d'un objet Screen (vue). La propriété correspond au code intégral d'accès au Screens Collections.

STRING (readonly)

Exemple

L'exemple suivant affiche le chemin de la vue "ScreenWindow1":

```
'VBS67
Dim objScreen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
MsgBox objScreen.AccessPath
```

Voir aussi

Objet ScreenItem (Page 132)

Objet Screens (Liste) (Page 141)

Propriété Activate (avant WinCC V7)

Description

Les données à représenter ne sont sélectionnées par ArchivServer que si cet attribut est activé. Pour réduire les temps d'ouverture de vue, cet attribut ne devrait pas être activé et la valeur ne devrait être modifiée dynamiquement que si besoin est.

Accès en lecture/en écriture

Pour différencier la propriété "Activate" de la méthode "Activate", on accède à la propriété via "Object".

Exemple

```
Dim ctrl
Set ctrl = ScreenItems("Control")
ctrl.Object.activate = true
```

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété Activate

Activate

Les données à représenter dans la fenêtre d'alarmes ne sont demandées au serveur d'alarmes que lorsque cet attribut est activé. Afin de réduire le temps d'ouverture de la vue à son affichage, n'activez pas cet attribut et éditez la valeur de façon dynamique en cas de besoin.

L'attribut est dynamisable avec le nom **Activate** . Le type de données est BOOLEAN.

Propriété ActiveProject

Description

Renvoie un objet de type "Project".

Voir aussi

Propriété Path (Page 504)
Propriété Name (Page 482)
Segment d'ellipse (Page 156)
Objet HMIRuntime (Page 125)

Propriété ActiveScreen

Description

Fournit une référence sur la vue qui contient l'objet ayant actuellement le focus.

Utilisation

Utilisez "ActiveScreen" pour accéder en Runtime aux propriétés de la vue qui contient l'objet actuellement focussé.

Exemple

L'exemple suivant attribue le nom de la vue actuelle à la variable "strScrName" et l'affiche dans une alarme :

```
'VBS68
Dim strScrName
strScrName = HMIRuntime.ActiveScreen.Objectname
MsgBox strScrName
```

Voir aussi

Objet Screen (Page 138)
Objet HMIRuntime (Page 125)

Propriété ActiveScreenItem

Description

Fournit une référence sur l'objet qui a actuellement le focus.

Utilisation

Utilisez "ActiveScreenItem" pour accéder en Runtime aux propriétés de l'objet qui a le focus.

Exemple

L'exemple suivant affiche le nom de l'objet qui a le focus dans la vue "ScreenWindow1" :

```
'VBS69
Dim objScreen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
MsgBox objScreen.ActiveScreenItem.ObjectName
```

Voir aussi

Objet ScreenItem (Page 132)

Objet HMIRuntime (Page 125)

Propriété Actualize

Description

La propriété "Index" référence un couple de colonnes ou une courbe. "Actualize" détermine si une représentation statique ou dynamique doit être utilisée pour ce couple de colonnes/cette courbe.

- 0: Représentation statique
- -1: Représentation dynamique

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété ActualPointLeft

Description

Détermine la coordonnée x du point d'angle actuel par rapport à l'origine de la vue (en haut, à gauche) ou la restitue. Chaque point d'angle est identifié par un indice dérivé du nombre ("PointCount") de points d'angle existants.

Une modification de la valeur peut avoir des répercussions sur les propriétés "Width" (largeur de l'objet) et "Left" (coordonnée x de la position de l'objet).

Voir aussi

Trait polygonal (Page 168)

Polygone (Page 166)

Objet ScreenItem (Page 132)

Propriété ActualPointTop

Description

Détermine la coordonnée y du point d'angle actuel par rapport à l'origine de la vue (en haut, à gauche) ou la restitue. Chaque point d'angle est identifié par un indice dérivé du nombre ("PointCount") de points d'angle existants.

Une modification de la valeur peut avoir des répercussions sur les propriétés "Height" (hauteur de l'objet) et "Top" (coordonnée y de la position).

Voir aussi

Trait polygonal (Page 168)

Polygone (Page 166)

Objet ScreenItem (Page 132)

Propriété AdaptBorder

Description

TRUE, lorsque le cadre doit, sur le plan dynamique, être adapté à la taille du texte. BOOLEAN Accès en écriture et en lecture.

Pour la liste des textes et le champ E/S: Accès en lecture uniquement.

Voir aussi

Bouton (Page 210)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété AdaptPicture

Description

Détermine si la vue représentée dans une fenêtre de vue peut ou non s'adapter à la taille de la fenêtre de vue en Runtime. Accès en lecture uniquement.

TRUE, lorsque la vue s'adapte à la taille de la fenêtre de vue.

FALSE, lorsque la vue ne s'adapte pas à la taille de la fenêtre de vue.

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété AdaptSize

Description

Détermine si la fenêtre de vue s'adapte ou non en Runtime à la vue qui y est représentée. Accès en lecture uniquement.

TRUE, lorsque la taille de la fenêtre de vue s'adapte à la vue.

FALSE, lorsque la taille de la fenêtre de vue ne s'adapte pas à la vue.

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété AdjustRuler

Description

Détermine si la fenêtre de la règle sera adaptée à chaque fois que la fenêtre de courbe s'affiche.

TRUE, si vous déplacez et cachez puis réaffichez la fenêtre de règle, cette dernière réapparaît à sa position et sa taille initiale.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

AI - Ap

Propriété AlarmID

Description

Retourne l'AlarmID de l'objet d'alarme. L'AlarmID est univoque et est attribuée par le système.

AlarmID (readonly)

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété AlarmHigh

Description

Détermine le seuil supérieur auquel l'alarme se déclenche ou le restitue.
Déterminez le type d'analyse (en pourcent ou en absolu) avec la propriété "TypeAlarmHigh".
la propriété "CheckAlarmHigh" détermine si la surveillance de ce seuil est activée.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété AlarmLogs

Description

Renvoie un objet de type "AlarmLogs".

AlarmLogs (read-only)

Voir aussi

Objet HMIRuntime (Page 125)

Propriété AlarmLow

Description

Détermine le seuil inférieur auquel l'alarme se déclenche ou le restitue.
Déterminez le type d'analyse (en pourcent ou en absolu) avec la propriété "TypeAlarmLow".
la propriété "CheckAlarmLow" détermine si la surveillance de ce seuil est activée.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété Alignment

Description

Détermine la représentation de l'échelle (gauche/droite ou haut/bas) en fonction de l'emplacement de l'objet BarGraph ou la restitue. Il faut que la propriété "Scaling" soit positionnée sur TRUE pour que l'échelle s'affiche.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété AlignmentLeft

Description

Détermine l'alignement horizontal du texte ou le restitue. Plage des valeurs de 0 à 2.

0 = gauche

1 = centré

2 = droite

Voir aussi

Signalisation groupée (Page 203)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété AlignmentTop

Description

Détermine l'alignement vertical du texte ou le restitue. Plage des valeurs de 0 à 2.

0 = haut

1 = centré

2 = bas

Voir aussi

Signalisation groupée (Page 203)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété AllowPersistence

Description

TRUE, lorsque des réglages sont possibles en ce qui concerne la persistance. BOOLEAN
Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)
WinCC Online Table Control (avant WinCC V7) (Page 286)
WinCC Function Trend Control (avant WinCC V7) (Page 284)
Objet ScreenItem (Page 132)

AllowXAxisColor-Propriétés

Description

TRUE, quand la couleur définie de l'axe X commun est affichée dans Runtime. BOOLEAN
Accès en écriture et en lecture.

Propriété AllServer (avant WinCC V7)

Description

Détermine que les données à représenter dans la fenêtre d'alarmes sont sélectionnées par tous les serveurs membres d'un système réparti et sur lesquels Alarm Logging est activé.
Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)
Objet ScreenItem (Page 132)

Propriété AllServer

Tous les serveurs - AllServer

Détermine si le système sélectionne tous les serveurs dont les packages ont été chargés et sur lesquels "Alarm Logging Runtime" est activé dans la liste de démarrage.

Valeur	Explication
TRUE	Tous les serveurs sont activés.
FALSE	Seul le serveur entré dans "Sélection de serveur" est activé.

L'attribut est dynamisable avec le nom **AllServer**. Le type de données est BOOLEAN.

Propriété Analog

Description

TRUE, lorsque l'horloge est représentée comme horloge analogique. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

AnchorRuler-Propriété

Description

TRUE, quand la fenêtre de la règle est liée fermement avec la fenêtre de courbe. BOOLEAN Accès en écriture et en lecture.

Propriété AngleAlpha

Description

Détermine l'angle de profondeur a pour l'effet tridimensionnel de l'objet "3DBarGraph" ou le restitue. Plage des valeurs en degrés de 0 à 90.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété AngleBeta

Description

Détermine l'angle de profondeur b pour l'effet tridimensionnel de l'objet "3DBarGraph" ou le restitue. Plage des valeurs en degrés de 0 à 90.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété AngleMax

Description

Détermine l'angle sur l'échelle au niveau duquel la graduation se termine ou le restitue. LONG Accès en écriture et en lecture.

Le début et la fin de la graduation s'écrivent avec les attributs "AngleMin" et "AngleMax" en degrés. AngleMin< AngleMax.

L'angle 0 degrés se trouve du côté droit du diamètre horizontal du disque gradué. Les valeurs d'angle positives sont comptées dans le sens horaire.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété AngleMin

Description

Détermine l'angle sur l'échelle au niveau duquel la graduation débute ou le restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété Application

Description

Restitue l'application Graphics-Designer lorsque la propriété Application est utilisée sans indicateur d'objet. Lorsque la propriété Application est utilisée avec indicateur d'objet, elle restitue un objet Application qui représente l'application avec laquelle l'objet défini a été créé. Accès en lecture uniquement.

Voir aussi

Fenêtre d'application (Page 183)

Objet ScreenItem (Page 132)

Propriété ApplyProjectSettings

Adopter les paramétrages de projet - ApplyProjectSettings

Cet attribut permet de déterminer si les paramétrages de projet issus de "Alarm Logging" sont repris.

Valeur	Explication
TRUE	L'option "Adopter les paramétrages de projet" est activée. Les blocs d'alarme configurés dans "Alarm Logging" sont adoptés dans AlarmControl avec toutes leurs propriétés. Les blocs d'alarme s'affichent avec ces propriétés dans la fenêtre d'alarmes.
FALSE	L'option "Adopter les paramétrages de projet" est désactivée. Vous pouvez ajouter ou supprimer des blocs d'alarme ou modifier les propriétés.

L'attribut est dynamisable avec le nom **ApplyProjectSettings**. Le type de données est BOOLEAN.

Ar - Ax

Propriété Archive

Description

La propriété "Index" référence un couple de colonnes. "Archive" définit l'archive de valeurs de processus reliée à ce couple de colonnes. Le nom de l'archive de valeurs de processus est spécifié sous la forme suivante: Servername::Archivname

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété ArchiveName

Nom - ArchiveName

Détermine l'archive utilisateur ou la vue à afficher. Ouvrez la boîte de dialogue "Package Browser" via le bouton pour sélectionner l'archive ou la vue.

L'attribut est dynamisable avec le nom **ArchiveName**. Le type de données est STRING.

Propriété ArchiveType

Type - ArchiveType

Indique si l'archive utilisateur sélectionnée est une archive ou une vue. Le champ ne peut pas être modifié.

L'attribut est dynamisable avec le nom **ArchiveType**. Le type de données est LONG.

Propriété AspectRatio

AspectRatio

Définit, pour les films, si le rapport hauteur/largeur est conservé.

L'attribut est dynamisable avec le nom **AspectRatio**. Le type de données est BOOLEAN.

Propriété Assignments

Description

Une liste contenant les affectations entre la valeur de sortie et le texte à afficher réellement. Les affectations dépendent du type de liste réglé Déterminez le type de liste avec la propriété ListType.

Accès en lecture uniquement.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété AssumeOnExit

Description

TRUE, lorsque le texte entré est validé après avoir quitté de champ de saisie (par exemple avec la touche <Tab> ou par clic de souris). BOOLEAN Accès en écriture et en lecture.

Voir aussi

Champ d'E/S (Page 193)

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété AssumeOnFull

Description

TRUE, lorsque le champ de saisie est automatiquement quitté après saisie complète (le nombre de caractères prédéterminé a été entré) et que la saisie est validée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété AutoCompleteColumns

Affichage des colonnes vides - AutoCompleteColumns

Détermine si des colonnes vides sont affichées lorsque le contrôle est plus large que les colonnes configurées.

Valeur	Explication
TRUE	Les colonnes vides sont affichées.
FALSE	Les colonnes vides ne sont pas affichées.

L'attribut est dynamisable avec le nom **AutoCompleteColumns**. Le type de données est BOOLEAN.

Propriété AutoCompleteRows

Affichage des lignes vides - AutoCompleteRows

Détermine si des lignes vides sont affichées lorsque le contrôle est plus long que le nombre des lignes configurées.

Valeur	Explication
TRUE	Les lignes vides sont affichées.
FALSE	Les lignes vides ne sont pas affichées.

L'attribut est dynamisable avec le nom **AutoCompleteRows**. Le type de données est BOOLEAN.

Propriété AutoPosition

Positionner automatiquement - AutoPosition

Détermine si RulerControl est positionné exactement en dessous du contrôle source.

Les paramétrages suivants sont disponibles :

Valeur	Explication
TRUE	RulerControl est positionné exactement en dessous du contrôle source.
FALSE	RulerControl est affiché d'une manière similaire à la position du contrôle que vous avez configurée.

L'attribut est dynamisable avec le nom **AutoPosition**. Le type de données est BOOLEAN.

Propriété Autorange

Description

TRUE, lorsque la plage de valeurs de l'axe y est automatiquement déterminée ou définie via les attributs "BeginValue" et "EndValue". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété AutorangeX

Description

TRUE, lorsque la plage des valeurs de l'axe X est automatiquement déterminée. FALSE, lorsqu'elle est définie via les attributs "BeginX" et "EndX". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété AutorangeY

Description

TRUE, lorsque la plage des valeurs de l'axe Y est automatiquement déterminée. FALSE, lorsqu'elle est définie via les attributs "BeginY" et "EndY". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété AutoScroll (avant WinCC V7)

Description

Définit comment se comporte la fenêtre d'alarmes lorsque de nouvelles alarmes arrivent. BOOLEAN Accès en écriture et en lecture.

TRUE: Une nouvelle alarme est ajoutée à la liste représentée dans la fenêtre des alarmes et automatiquement sélectionnée. La zone visible de la fenêtre d'alarmes est au besoin déplacée.

FALSE: Une nouvelle alarme n'est pas sélectionnée. La zone visible de la fenêtre d'alarmes n'est pas modifiée.

La sélection ponctuelle de lignes d'alarme n'est possible que si "AutoScroll" n'est pas activé.

La propriété "AutoScroll" est désactivé lorsque l'attribut "MsgCtrlFlag" = "-1" est positionné. Cela signifie que l'alarme la plus récente s'affichera en haut de la fenêtre d'alarmes.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété AutoScroll

Défilement automatique - AutoScroll

Définit comment se comporte la fenêtre d'alarmes lorsque de nouvelles alarmes arrivent.

La sélection de lignes d'alarme n'est possible que si "Défilement auto." n'est pas activé.

Valeur	Explication
TRUE	Lorsque "Défilement automatique" est activé, les nouvelles alarmes sont ajoutées à la liste affichée dans la fenêtre d'alarmes et sélectionnées automatiquement. La zone visible de la fenêtre d'alarmes est au besoin déplacée.
FALSE	Si "Défilement automatique" est désactivé, les nouvelles alarmes ne sont pas sélectionnées à leur apparition. La zone visible de la fenêtre d'alarmes n'est pas modifiée.

L'attribut est dynamisable avec le nom **AutoScroll**. Le type de données est BOOLEAN.

Propriété AutoSelectionColors

Couleur automatique Marquage - AutoSelectionColors

Détermine si les couleurs de marquage pour cellules et lignes sont représentées avec les couleurs définies par le système.

Valeur	Explication
TRUE	Les couleurs utilisées sont celles du système.
FALSE	Les couleurs utilisées sont celles définies par l'utilisateur.

L'attribut est dynamisable avec le nom **AutoSelectionColors**. Le type de données est BOOLEAN.

Propriété AutoSelectionRectColor

Couleur automatique Cadre de sélection - AutoSelectionRectColor

Détermine si le cadre de sélection sera représenté dans la couleur définie par le système.

Valeur	Explication
TRUE	La couleur utilisée est celle du système.
FALSE	La couleur utilisée est celle définie par l'utilisateur.

L'attribut est dynamisable avec le nom **AutoSelectionRectColors**. Le type de données est BOOLEAN.

Propriété AutoShow

Afficher/masquer automatiquement - AutoShow

Détermine si RulerControl est automatiquement affiché lorsque vous avez sélectionné les fonctions de touche pour la règle, la plage de statistiques ou les statistiques dans le contrôle source.

Si vous n'utilisez plus la règle, la plage de statistiques ou les statistiques, RulerControl est de nouveau masqué.

Valeur	Explication
TRUE	RulerControl est affiché automatiquement.
FALSE	RulerControl n'est pas affiché automatiquement.

L'attribut est dynamisable avec le nom **AutoShow**. Le type de données est BOOLEAN.

Propriété AutoSize

Description

Détermine l'adaptation en taille de l'objet ou la restitue. Les paramétrages suivants sont possibles :

- 0: Aucune adaptation de taille.
- 1: La vue (Propriétés "PictureSelected", "PictureUnselected") s'adapte au bouton de commande.
- 2: Le bouton s'adapte à la vue (Propriétés "PictureSelected", "PictureUnselected").

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété Autostart

Démarrage automatique

Définit, pour les films, si la lecture commence automatiquement.

L'attribut est dynamisable avec le nom **Autostart**. Le type de données est BOOLEAN.

Propriété Average

Average

TRUE, lorsqu'une moyenne est constituée à partir des 10 dernières valeurs. Pour constituer une nouvelle moyenne, il faut modifier une valeur. La moyenne est réinitialisée lors d'un changement de vue. S'il n'y a par exemple qu'une seule valeur disponible après le changement de vue, la moyenne suivante est constituée : (5+0+0+0+0+0+0+0+0+0)/10=0,5.

Accès en écriture et en lecture BOOLEEN.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété Axe

Description

Définit la position du bargraphe tridimensionnel dans le système de coordonnées ou restitue la valeur. Plage des valeurs de 0 à 2.

0: Le bargraphe tridimensionnel est représenté sur l'axe X.

1: Le bargraphe tridimensionnel est représenté sur l'axe Y.

2: Le bargraphe tridimensionnel est représenté sur l'axe Z.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété AxisSection

Description

Définit la distance entre deux longues sections axiales ou la restitue. La distance est alors spécifiée en unités de graduation et dépend des valeurs maximales et minimales configurées.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

1.14.4.3 B

Ba

Propriété BackBorderWidth

Description

Détermine la largeur du cadre tridimensionnel en Pixel ou la restitue. La valeur pour la largeur dépend de la taille de l'objet.

Voir aussi

Objet ScreenItem (Page 132)

Bouton (Page 210)

Bouton rond (Page 217)

Slider (Page 219)

Signalisation groupée (Page 203)

Propriété BackColor

Arrière-plan - BackColor

Indique la couleur d'arrière-plan du contrôle. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **BackColor**. Le type de données est LONG.

Propriété BackColor

Couleur d'arrière-plan (BackColor)

Définit la couleur d'arrière-plan de l'icône au moyen du dialogue "Sélection des couleurs". La couleur d'arrière-plan est représentée pour le type d'arrière-plan "Opaque".

L'attribut est dynamisable avec le nom **BackColor**. Le type de données est LONG.

Propriété BackColor

Fonction

Détermine la couleur d'arrière-plan de l'objet ou la restitue.

Pour les objets avec motif de remplissage, la couleur d'arrière-plan n'est pas affichée lorsque le motif de remplissage est paramétré sur "Transparent".

Particularité pour WinCC Slider-Control

La couleur d'arrière-plan n'est effective que si l'objet est au moins partiellement rempli.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Spécifiez la valeur décimale correspondante pour chacune des trois valeurs RVB.

Exemple :

RVB(200, 150, 100)

Exemple

L'exemple suivant positionne sur Rouge l'arrière-plan de la vue "ScreenWindow1":

```
'VBS70
Dim objScreen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
objScreen.BackColor = RGB(255, 0, 0)
```

Voir aussi

Propriété FillStyle (Page 394)

Propriété FillColor (Page 392)

Objet ScreenItem (Page 132)

Propriété BackColor2

Description

Définit la couleur de bargraphe pour l'affichage de la valeur actuelle ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété BackColor3

Description

Définit la couleur de l'arrière-plan de bargraphe ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Bargraphe (Page 184)

Propriété BackColorBottom

Description

Définit la couleur pour la partie inférieure/droite du Slider ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Slider (Page 219)

Objet ScreenItem (Page 132)

Propriété BackColorTop

Description

Définit la couleur pour la partie supérieure/gauche du Slider ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Slider (Page 219)

Objet ScreenItem (Page 132)

Propriété BackFlashColorOff

Description

Définit la couleur de l'arrière-plan d'objet pour l'état clignotant "Arrêt" ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Propriété BackFlashColorOn

Description

Définit la couleur de l'arrière-plan d'objet pour l'état clignotant "Marche" ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Propriété Background

Description

TRUE, lorsque l'arrière-plan d'un objet bargraphe tridimensionnel doit être visible. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété BackgroundPicture

Description

Restitue le nom d'image de la vue d'arrière-plan pour le disque gradué. Accès en lecture uniquement.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété BackPictureAlignment

Description

Définit ou restitue le mode de représentation de la vue d'arrière-plan dans la vue de process.

LONG Accès en écriture et en lecture.

Propriété BackPictureName

Description

Définit le chemin et le nom de fichier du papier peint dans la vue de process, ou les retourne. LONG Accès en écriture et en lecture.

Propriété BackStyle

Description

WinCC Digital/Analog Clock

Détermine le type d'arrière-plan de l'horloge analogique:

- 0: L'arrière-plan rectangulaire de l'horloge est rempli avec la couleur d'arrière-plan spécifiée.
- 1: Le cadran rond de l'horloge est rempli avec la couleur d'arrière-plan spécifiée. Vous pouvez ainsi représenter une horloge analogique ronde.
- 2: Le cadran et l'arrière plan rectangulaire sont transparents.

WinCC Gauge Control

Détermine le type d'arrière-plan de la jauge:

- 0: L'arrière-plan rectangulaire ou carré de la jauge est rempli par la couleur spécifiée pour le cadre. Le disque gradué en forme de cercle est rempli avec la couleur d'arrière-plan spécifiée.
- 1: L'arrière-plan rectangulaire ou carré de la jauge est transparent. Le disque gradué en forme de cercle est rempli avec la couleur d'arrière-plan spécifiée. Vous pouvez ainsi représenter une jauge en forme de cercle.
- 2: L'arrière-plan rectangulaire ou carré et le disque gradué sont transparents.

WinCC Slider Control

Détermine si l'arrière-plan de l'objet est transparent.

- 0: L'arrière-plan de l'objet n'est pas transparent
- 1: L'arrière-plan de l'objet est transparent

HMI Symbol Library

Détermine la transparence de l'arrière-plan du symbole. Accès en écriture et en lecture.

- 0: L'arrière-plan est transparent et n'est donc pas visible.
- 1: L'arrière-plan est visible, la couleur de l'arrière-plan est déterminée par l'attribut "Couleur d'arrière-plan".

Voir aussi

HMI Symbol Library (Page 246)

WinCC Slider Control (Page 275)

WinCC Gauge Control (Page 258)

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété BarBackColor

Description

Détermine la couleur d'arrière-plan dans la zone du curseur. La plage s'étend de "RangeMin" à "RangeMax".

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété BarDepth

Description

Détermine la profondeur de bargraphe en Pixel ou la restitue.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété BarFillColor

Description

Détermine la couleur de remplissage dans la zone du curseur. La plage s'étend de "RangeMin" à la position du curseur.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété BarHeight

Description

Détermine la hauteur de bargraphe en Pixel ou la restitue.

Voir aussi

Bargraphes tridimensionnels (Page 179)
Objet ScreenItem (Page 132)

Propriété BarWidth

Description

Détermine la largeur de bargraphe en Pixel ou la restitue.

Voir aussi

Bargraphes tridimensionnels (Page 179)
Objet ScreenItem (Page 132)

Propriété BasePicReferenced

Description

TRUE, lorsque la vue assignée doit être enregistrée dans l'objet "Indicateur d'état". Faute de quoi, seul le renvoi à l'objet respectif est enregistré. Accès en lecture uniquement.

Voir aussi

Indicateur d'état (Page 208)
Objet ScreenItem (Page 132)

Propriété BasePicTransColor

Description

Détermine quelle couleur de l'objet Bitmap assigné (.bmp, .dib) doit être configurée sur "transparent" ou restitue la couleur. LONG Accès en lecture et en écriture.
La couleur n'est configurée sur "transparent" que si la propriété "BasePicUseTransColor" est positionnée sur la valeur TRUE.

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété BasePicture

Description

Restitue la vue de base pour l'objet "Indicateur d'état". Accès en lecture uniquement.
La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.
La propriété "BasePicReferenced" détermine dans ce contexte si la vue de base doit être enregistrée ou référencée en même temps que l'objet "Indicateur d'état".

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété BasePicUseTransColor

Description

TRUE, lorsque la couleur configurée (propriété "BasePicTransColor") pour l'objet Bitmap doit être positionnée sur "transparent". BOOLEAN Accès en écriture et en lecture.

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété BaseScreenName

Fonction

Détermine la vue de base actuelle ou la restitue.

STRING (Accès en écriture et en lecture)

Un changement de vue est exécuté via

```
HMIRuntime.BaseScreenName = (<Préfixe serveur>::)<Nouvelle vue de base>
```

l'instruction.

Lors de la lecture de la propriété "BaseScreenName" c'est toujours le nom de la vue sans préfixe de serveur qui est restitué.

Remarque

Pour des raisons de compatibilité avec les versions ultérieures, il est toujours préférable d'écrire les noms de vue sans l'extension de fichier ".PDL" .

Exemple

L'exemple suivant présente un changement de vue sur "bild1.pdl":

```
HMIRuntime.BaseScreenName = "vue1"
```

Voir aussi

Objet ScreenItem (Page 132)

Objet HMIRuntime (Page 125)

Propriété BaseY

Description

Détermine la distance verticale entre le bord inférieur du bargraphe et le bord supérieur du champ de l'objet ou la restitue.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété BaseX

Description

Détermine la distance horizontale en Pixel entre le bord droit du bargraphe et le bord gauche du champ de l'objet ou la restitue.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Be - Bl

Propriété BeginTime

Description

WinCC Online Table Control

La propriété "Index" référence un couple de colonnes. "BeginTime" détermine le moment de lancement de la représentation de ce couple de colonnes. Accès en écriture et en lecture.

WinCC Online Trend Control

La propriété "Index" référence une courbe. "BeginTime" détermine le moment de lancement de la représentation de cette courbe. L'indication sera ou non exploitée selon les propriétés "TimeRange" et "CommonX".

Si vous dynamisez la plage de temporisation, utilisez le format "aaaa-mm-jj hh:mm:ss".

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété BeginValue

Description

La propriété "Index" référence une courbe. "BeginValue" détermine pour cette courbe le seuil inférieur de la plage des valeurs à représenter. L'indication sera ou non exploitée selon les propriétés "Autorange" et "CommonY".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété BeginX

Description

Détermine le seuil inférieur de l'axe X d'une courbe référencée avec la propriété "Index" ou le restitue. L'indication sera ou non exploitée selon les propriétés "AutorangeX" et "CommonX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété BeginY

Description

Détermine le seuil inférieur de l'axe Y d'une courbe référencée avec la propriété "Index" ou le restitue. L'indication sera ou non exploitée selon les propriétés "AutorangeY" et "CommonY".

Voir aussi

Objet ScreenItem (Page 132)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Propriété BevelColorDown

Description

Détermine la couleur des parties de bordure suivantes en cas de représentation tridimensionnelle du cadre:

- si la bordure est en creux ("BevelStyle" = 1): élément de bordure supérieur et gauche
- si la bordure est en relief ("BevelStyle" = 2): élément de bordure inférieur et droit

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété BevelColorUp

Description

Détermine la couleur des parties de bordure suivantes en cas de représentation tridimensionnelle du cadre:

- si la bordure est en creux ("BevelStyle" = 1): élément de bordure inférieur et droit
- si la bordure est en relief ("BevelStyle" = 2): élément de bordure supérieur et gauche

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété BevelInner

Description

Détermine l'apparence de la partie intérieure du cadre de l'objet ou la restitue. Accès en écriture et en lecture.

- 0: partie intérieur n'existe pas
- 1: apparence "en creux"
- 2: apparence "en relief"
- 3: Bordure uniformément grise
- 4 ou supérieur: Bordure de couleur uniforme, Couleur de bordure = Couleur d'arrière-plan

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété BevelOuter

Description

Détermine l'apparence de la partie extérieur du cadre de l'objet ou la restitue. Accès en écriture et en lecture.

- 0: partie intérieur n'existe pas
- 1: apparence "en creux"
- 2: apparence "en relief"

- 3: Bordure uniformément grise
- 4 ou supérieur: Bordure de couleur uniforme, Couleur de bordure = Couleur d'arrière-plan

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété BevelWidth

Description

Détermine la largeur de la bordure pour la partie intérieure du cadre (cadre intérieur) et pour la partie extérieure du cadre (cadre extérieur) en Pixel ou la restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété BitNumber

Description

Détermine le bit dont l'état doit être modifié pour déclencher une modification de valeur ou le restitue. La variable utilisée doit être de type BYTE, WORD ou DWORD.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété BlinkColor

Description

Détermine la couleur du symbole dans la vue clignotante. LONG Accès en écriture et en lecture.

Voir aussi

HMI Symbol Library (Page 246)

Objet ScreenItem (Page 132)

Propriété BlinkMode

Mode de clignotement (BlinkMode)

Définit le mode de clignotement de l'icône au Runtime.

Les paramétrages suivants sont possibles :

Valeur	Description	Signification
0	Pas de clignotement	L'icône ne clignote pas.
1	Invisible	L'icône clignote dans la couleur d'arrière-plan.
2	Ombré	L'icône clignote dans la couleur de premier plan avec un aspect ombré.
3	Opaque	L'icône clignote dans la couleur de premier plan.

L'attribut est dynamisable avec le nom **BlinkMode**. Le type de données est LONG.

Propriété BlinkSpeed

Vitesse de clignotement (BlinkSpeed)

Définit la longueur de l'intervalle de clignotement de l'icône au Runtime.

Les paramétrages suivants sont possibles :

Valeur	Description	Signification
250	Rapide	La longueur de l'intervalle de clignotement est de 250 ms.
500	Moyenne	La longueur de l'intervalle de clignotement est de 500 ms.
1000	Lente	La longueur de l'intervalle de clignotement est de 1000 ms.

L'attribut est dynamisable avec le nom **BlinkSpeed**. Le type de données est LONG.

Propriété BlockAlign

Alignement Blocs - BlockAlign

Détermine comment s'aligne la désignation du bloc dans le titre de colonne.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	à gauche	La désignation du bloc s'affiche à gauche.
1	centré	La désignation du bloc est centrée.
2	à droite	La désignation du bloc s'affiche à droite.

L'attribut est dynamisable avec le nom **BlockAlign**. Le type de données est LONG.

Propriété BlockAutoPrecisions

Chiffres après la virgule Automatique - BlockAutoPrecisions

Détermine si le nombre de chiffres après la virgule est déterminé automatiquement.

Valeur	Explication
TRUE	Le nombre de chiffres après la virgule est défini automatiquement. La valeur du champ "Chiffres après la virgule" reste sans effet.
FALSE	La valeur du champ "Chiffres après la virgule" est effective.

L'attribut est dynamisable avec le nom **BlockAutoPrecisions**. Le type de données est BOOLEAN.

Propriété BlockCaption

Désignation - BlockCaption

Permet de désigner dans le contrôle le titre de colonne pour le bloc sélectionné.

La désignation est effective dans toutes les langues Runtime.

L'attribut est dynamisable avec le nom **BlockCaption**. Le type de données est STRING.

Propriété BlockCount

BlockCount

Indique le nombre de blocs disponibles comme colonnes pour le contrôle.

L'attribut est dynamisable avec le nom **BlockCount**. Le type de données est LONG.

Propriété BlockDateFormat

Format de la date - BlockDateFormat

Détermine quel format de la date sera utilisé pour l'affichage.

Les formats de date suivants sont disponibles :

Valeur	Explication
Automatique	Le format de la date est automatiquement défini.
dd.MM.yy	Jour.Mois.Année, par exemple 24.12.07.
dd.MM.yyyy	Jour.Mois.Année, par exemple 24.12.2007.
dd/MM/yy	Jour/Mois/Année, par exemple 24/12/07.
dd/MM/yyyy	Jour/Mois/Année, par exemple 24/12/2007.

L'attribut est dynamisable avec le nom **BlockDateFormat**. Le type de données est STRING.

Propriété BlockExponentialFormat

Représentation exponentielle - BlockExponentialFormat

Détermine si les valeurs du bloc sélectionné s'affichent en représentation exponentielle.

Valeur	Explication
TRUE	Les valeurs sont affichées en représentation exponentielle.
FALSE	Les valeurs sont affichées en représentation décimale.

L'attribut est dynamisable avec le nom **BlockExponentialFormat**. Le type de données est BOOLEAN.

Propriété BlockHideText

Contenu comme texte - BlockHideText

Détermine si le contenu du bloc sélectionné doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le contenu ne s'affiche pas sous forme de texte. L'option n'est pas activée.
FALSE	Le contenu s'affiche sous forme de texte. L'option est activée.

L'attribut est dynamisable avec le nom **BlockHideText**. Le type de données est BOOLEAN.

Propriété BlockHideTitleText

Titre comme texte - BlockHideTitleText

Détermine si le titre du bloc sélectionné doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le titre ne s'affiche pas sous forme de texte. L'option n'est pas activée.
FALSE	Le titre s'affiche sous forme de texte. L'option est activée.

L'attribut est dynamisable avec le nom **BlockHideTitleText**. Le type de données est BOOLEAN.

Propriété BlockId

BlockId

Affectation définie du numéro d'identification et du bloc dans WinCC RulerControl :

Valeur	Description
0	Aucun bloc
1	Nom
2	Index
3	Désignation
4	Affichage
5	Nom de variable Y
6	Nom de variable X
7	Valeur Y
8	Valeur X/Horodatage
9	Valeur Y (seuil inférieur SI)
10	Horodatage (seuil inférieur SI)
11	Valeur Y (seuil supérieur SP)
12	Horodatage (seuil supérieur SP)
13	Minimum
14	Minimum - Horodatage
15	Maximum
16	Maximum - Horodatage
17	Moyenne
18	Différence standard
19	Intégral
20	Moyenne pondérée
21	Durée
22	Nombre de valeurs

L'attribut est dynamisable avec le nom **BlockID**. Le type de données est LONG.

Propriété BlockIndex

BlockIndex

Cet attribut fait référence à un bloc. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un bloc précis.

Les valeurs admissibles pour "BlockIndex" et "BlockCount" se situent entre 0 et -1. L'attribut "BlockCount" indique le nombre de blocs existants.

L'attribut est dynamisable avec le nom **BlockIndex**. Le type de données est LONG.

Propriété BlockLength

Longueur en caractères - BlockLength

Détermine la largeur de la colonne pour le bloc sélectionné.

L'attribut est dynamisable avec le nom **BlockLength**. Le type de données est LONG.

Propriété BlockName

Nom d'objet - BlockName

Affiche le nom pour le bloc sélectionné. Vous ne pouvez pas modifier le nom.

L'attribut est dynamisable avec le nom **BlockName**. Le type de données est STRING.

Propriété BlockPrecisions

Chiffres après la virgule - BlockPrecisions

Définit le nombre de chiffres après la virgule des valeurs de la colonne sélectionnée. Vous ne pouvez entrer la valeur que si l'option "Automatique" est désactivée.

L'attribut est dynamisable avec le nom **BlockPrecisions**. Le type de données est SHORT.

Propriété BlockShowDate

Affichage de la date - BlockShowDate

Détermine si le bloc "Heure" s'affiche dans un champ avec l'heure et la date.

Valeur	Explication
TRUE	La date et l'heure s'affichent. Le format de la date est déterminé dans le champ "Format de la date".
FALSE	L'heure s'affiche.

L'attribut est dynamisable avec le nom **BlockShowDate**. Le type de données est BOOLEAN.

Propriété BlockShowIcon

Contenu comme symbole - BlockShowIcon

Détermine si le contenu du bloc sélectionné doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le contenu s'affiche sous forme de symbole.
FALSE	Le contenu ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **BlockShowIcon**. Le type de données est BOOLEAN.

Propriété BlockShowTitleIcon

Titre comme symbole - BlockShowTitleIcon

Détermine si le titre du bloc sélectionné doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le titre s'affiche sous forme de symbole.
FALSE	Le titre ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **BlockShowTitleIcon**. Le type de données est BOOLEAN.

Propriété BlockTimeFormat

Format de l'heure - BlockTimeFormat

Détermine quel format de l'heure sera utilisé pour l'affichage.

Les formats d'heure suivants sont disponibles :

Valeur	Explication
Automatique	Le format de l'heure est automatiquement défini.
HH:mm:ss.ms	Heures:Minutes:Secondes.Millisecondes, par exemple 15:35:44.240.
hh:mm:ss tt	Heures:Minutes:Secondes AM/PM, par exemple 03:35:44 PM.
hh:mm:ss.ms tt	Heures:Minutes:Secondes.Millisecondes AM/PM, par exemple 03:35:44.240 PM.

L'attribut est dynamisable avec le nom **BlockTimeFormat**. Le type de données est STRING.

Propriété BlockUseSourceFormat

Appliquer à partir de la source - BlockUseSourceFormat

Détermine si les formatages sont acceptés par le contrôle lié.

Valeur	Explication
TRUE	Les formatages sont acceptés par le contrôle lié.
FALSE	Les formatages configurés ici sont acceptés.

L'attribut est dynamisable avec le nom **BlockUseSouceFormat**. Le type de données est BOOLEAN.

Bo - Bu

Propriété BorderBackColor

Description

Détermine la couleur d'arrière-plan du trait pour l'objet ou la restitue. LONG Accès en écriture et en lecture. La couleur d'arrière-plan n'est visible que si la propriété "BorderWidth" > 0 est positionnée.

Voir aussi

Objet ScreenItem (Page 132)

Propriété BorderColor

Description

Détermine la couleur du trait pour l'objet ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Propriété BorderColor

Couleur de bordure - BorderColor

Indique la couleur de la bordure. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **BorderColor**. Le type de données est LONG.

Propriété BorderColorBottom

Description

Définit la couleur du bord pour la partie inférieure/droite de l'objet ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Bouton (Page 210)

Bouton rond (Page 217)

Propriété BorderColorTop

Description

Définit la couleur du bord pour la partie supérieure/gauche de l'objet ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Bouton (Page 210)

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété BorderEndStyle

Description

Détermine le type des fins de trait de l'objet ou le restitue. LONG Accès en écriture et en lecture.

Voir aussi

Trait polygonal (Page 168)

Trait (Page 164)

Objet ScreenItem (Page 132)

Propriété BorderFlashColorOff

Description

Définit la couleur du trait de l'objet pour l'état clignotant "Arrêt" ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Propriété BorderFlashColorOn

Description

Définit la couleur du trait de l'objet pour l'état clignotant "Marche" ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Propriété BorderStyle

Description

Détermine le type de trait pour l'objet ou le restitue. Plage des valeurs de 0 à 4.

0 = Trait plein

1 = Trait discontinu

2 = Trait en pointillés

3 = Trait mixte

4 = Trait point point

Voir aussi

Objet ScreenItem (Page 132)

Propriété BorderWidth

Description

Détermine l'épaisseur du trait (en Pixel) pour l'objet ou la restitue.

WinCC Gauge Control:

Détermine la largeur de la partie centrale du cadre en Pixel ou la restitue.

Le cadre de l'objet se compose de trois parties. La partie centrale du cadre de l'objet est décrite par la propriété "BorderWidth".

La couleur de la partie centrale du cadre est la couleur d'arrière-plan.

Voir aussi

Objet ScreenItem (Page 132)

Propriété BorderWidth

Largeur de bordure - BorderWidth

Détermine la largeur du trait de la bordure en pixels.

L'attribut est dynamisable avec le nom **BorderWidth**. Le type de données est LONG.

Propriété BottomConnectedConnectionPointIndex

Description

Spécifie le numéro d'indice du point inférieur de connecteur ou le positionne.

Long Accès en écriture et en lecture.

Voir aussi

Connecteur (Page 178)

Objet ScreenItem (Page 132)

Propriété BottomConnectedObjectName

Description

Spécifie le nom de l'objet qui est attaché au point inférieur de connecteur ou le positionne.

Long Accès en écriture et en lecture.

Voir aussi

Connecteur (Page 178)

Objet ScreenItem (Page 132)

Propriété BoxAlignment

Description

TRUE, lorsque les champs sont alignés à droite. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Case d'option (Page 215)

Case à cocher (Page 213)

Objet ScreenItem (Page 132)

Propriété BoxCount

Description

Détermine le nombre de champs ou le restitue. Plage des valeurs de 0 à 31.

Voir aussi

Case d'option (Page 215)

Case à cocher (Page 213)

Objet ScreenItem (Page 132)

Propriété BoxType

Description

Détermine le type de champ ou le restitue. Plage des valeurs de 0 à 2.

- 0: Sortie
- 1: Entrée
- 2: Champ d'E/S

Voir aussi

Liste de textes (Page 206)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété ButtonColor

Description

Définit la couleur du curseur ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Slider (Page 219)

Objet ScreenItem (Page 132)

Propriété ButtonCommand

Description

Lorsque la valeur de ButtonCommand est modifiée, un message est envoyé au contrôle WinCC Alarm Control pour influencer l'affichage dans la fenêtre d'alarmes.

Valeur (hex); Valeur (déc); Fonction appelée :

- 0x00000001; 1; Liste d'alarmes
- 0x00000002; 2; Liste d'archives courte durée
- 0x00000004; 4; Liste d'archives longue durée
- 0x00200000; 2097152; Liste de verrouillage
- 0x00000008; 8; Acquittement avertisseurs centraux
- 0x00000010; 16; Acquittement individuel
- 0x00000020; 32; Acquittement groupé
- 0x00000040; 64; Défilement automatique
- 0x00000080; 128; Dialogue de sélection
- 0x00000100; 256; Dialogue de verrouillage
- 0x00000200; 512; Impression du journal d'alarmes
- 0x00000800; 2048; Acquittement d'urgence
- 0x00001000; 4096; Première alarme
- 0x00002000; 8192; Dernière alarme
- 0x00004000; 16384; Alarme suivante
- 0x00008000; 32768; Alarme précédente
- 0x00010000; 65536; Dialogue Texte d'info
- 0x00020000; 131072; Dialogue Commentaire
- 0x00040000; 262144; Loop in Alarm
- 0x00100000; 1048576; Impression de la vue actuelle
- 0x00400000; 4194304; Verrouiller/Valider l'alarme
- 0x00800000; 8388608; Dialogue de tri
- 0x01000000; 16777216; Dialogue Base de temps
- 0x02000000; 33554432; Liste d'occurrences
- 0x04000000; 67108864; liste des alarmes devant être masquées
- 0x08000000; 134217728; Afficher / Masquer l'alarme
- 0x10000000; 268435456; boîte de dialogue des options d'affichage

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété Button1Width

Description

Détermine la largeur de Button 1 en Pixel ou la restitue.
Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété Button2Width

Description

Détermine la largeur de Button 2 en Pixel ou la restitue.
Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété Button3Width

Description

Détermine la largeur de Button 3 en Pixel ou la restitue.
Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Voir aussi

Objet ScreenItem (Page 132)

Signalisation groupée (Page 203)

Propriété Button4Width

Description

Détermine la largeur de Button 4 en Pixel ou la restitue.
Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

1.14.4.4 C

Ca - Cl

Propriété Caption

Description

Fenêtre de vue et d'application

TRUE, lorsque la fenêtre de vue ou d'application a une barre de titre dans Runtime. Accès en lecture uniquement.

La propriété "Caption" doit être positionnée sur TRUE lorsque la fenêtre de vue ou d'application doit posséder des boutons de commande pour Maximiser et Quitter.

Avant Controls WinCC V7

Détermine le texte devant s'afficher comme libellé sur le bouton ou sur la barre de titre (Online Trend Control et Online Table Control) ou le restitue. Accès en écriture et en lecture.

Voir aussi

Contrôles (Page 224)

Fenêtre de vue (Page 188)

Fenêtre d'application (Page 183)

Objet ScreenItem (Page 132)

Propriété Caption

Texte - Caption

Détermine le texte du titre de fenêtre.

L'attribut est dynamisable avec le nom **Caption**. Le type de données est STRING.

Propriété CaptionColor

Description

Détermine la couleur du libellé ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

WinCC Gauge Control (Page 258)

Propriété CaptionFont

Description

Restitue au libellé la valeur pour la police, le type de police et la taille ainsi que pour les effets "Souligné" et "Barré". Accès en lecture uniquement.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété CaptionOffset

Description

Détermine la distance entre le texte et le haut de l'objet ou la restitue. Le texte n'est positionnable que le long du diamètre vertical du disque gradué. La valeur de l'attribut se réfère à la hauteur de l'objet et est mesurée du haut de l'objet au bas de l'inscription. Accès en écriture et en lecture.

La plage des valeurs est comprise entre 0 et 1:

0: Le bas de l'inscription se trouve à la limite supérieure de l'objet. Le texte n'est plus visible étant donné qu'il se trouve à l'extérieur de l'objet.

1: Le bas de l'inscription se trouve à la limite inférieure de l'objet.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété CaptionText

Description

Détermine le titre de fenêtre qui s'affiche en Runtime ou le restitue.
La propriété "Caption" doit pour cela être positionnée sur TRUE.

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété CellCut (avant WinCC V7)

Description

TRUE, lorsque le contenu des champs d'une ligne d'alarme est raccourci parce que la colonne n'est pas assez large. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété CellCut

Abréger les contenus - CellCut

Détermine si les contenus des cellules sont abrégés lorsque les cellules sont trop étroites.

Valeur	Explication
TRUE	Les contenus des colonnes sont abrégés.
FALSE	Les contenus des colonnes ne sont pas abrégés.

L'attribut est dynamisable avec le nom **CellCut**. Le type de données est BOOLEAN.

Propriété CellSpaceBottom

CellSpaceBottom

Cet attribut définit l'intervalle inférieur utilisé dans les cellules du tableau.

L'attribut est dynamisable avec le nom **CellSpaceBottom**. Le type de données est LONG.

Propriété CellSpaceLeft

CellSpaceLeft

Cet attribut définit l'indentation à gauche utilisée dans les cellules du tableau.

L'attribut est dynamisable avec le nom **CellSpaceLeft** . Le type de données est LONG.

Propriété CellSpaceRight

CellSpaceRight

Cet attribut définit l'indentation à droite utilisée dans les cellules du tableau.

L'attribut est dynamisable avec le nom **CellSpaceRight** . Le type de données est LONG.

Propriété CellSpaceTop

CellSpaceTop

Cet attribut définit l'intervalle supérieur utilisé dans les cellules du tableau.

L'attribut est dynamisable avec le nom **CellSpaceTop** . Le type de données est LONG.

Propriété CenterColor

Description

Détermine la couleur du centre du disque gradué (recouvrement de l'axe des aiguilles) ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété CenterScale

Description

Détermine le diamètre du centre du disque gradué (recouvrement de l'axe des aiguilles) par rapport à la plus petite valeur des attributs géométriques Largeur et Hauteur, ou le restitue. Accès en écriture et en lecture.

La plage des valeurs est comprise entre 0,03 et 1:

1: Le diamètre correspond à la plus petite valeur des attributs géométriques "Width" et "Height".

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété CheckAlarmHigh

Description

TRUE, lorsque le seuil "AlarmHigh" est surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "AlarmHigh", "ColorAlarmHigh" et "TypeAlarmHigh".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckAlarmLow

Description

TRUE, lorsque le seuil "AlarmLow" est surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "AlarmLow", "ColorAlarmLow" et "TypeAlarmLow".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckLimitHigh4

Description

TRUE, lorsque le seuil supérieur "Reserve 4" doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "LimitHigh4", "ColorLimitHigh4" et "TypeLimitHigh4".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckLimitHigh5

Description

TRUE, lorsque le seuil supérieur "Reserve 5" doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "LimitHigh5", "ColorLimitHigh5" et "TypeLimitHigh5".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckLimitLow4

Description

TRUE, lorsque le seuil inférieur "Reserve 4" doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "LimitLow4", "ColorLimitLow4" et "TypeLimitLow4".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckLimitLow5

Description

TRUE, lorsque le seuil inférieur "Reserve 5" doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "LimitLow5", "ColorLimitLow5" et "TypeLimitLow5".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckToleranceHigh

Description

TRUE, lorsque le seuil "ToleranceHigh" est surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "ToleranceHigh", "ColorToleranceHigh" et "TypeToleranceHigh".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckToleranceLow

Description

TRUE, lorsque le seuil "ToleranceLow" est surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "ToleranceLow", "ColorToleranceLow" et "TypeToleranceLow".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckWarningHigh

Description

TRUE, lorsque le seuil "WarningHigh" est surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "WarningHigh", "ColorWarningHigh" et "TypeWarningHigh".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété CheckWarningLow

Description

TRUE, lorsque le seuil "WarningLow" est surveillé. BOOLEAN Accès en lecture et en écriture. Le seuil, la représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "WarningLow", "ColorWarningLow" et "TypeWarningLow".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ClearOnError

Description

TRUE, lorsque l'entrée faite dans le champ est automatiquement supprimée au cas où elle serait erronée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété ClearOnNew

Description

TRUE, lorsque l'entrée faite dans le champ est supprimée dès que le champ d'E/S a le focus. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété Closeable (avant WinCC V7)

Description

TRUE, lorsque la fenêtre peut être fermée en Runtime. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)
WinCC Online Trend Control (avant WinCC V7) (Page 288)
WinCC Online Table Control (avant WinCC V7) (Page 286)
Objet ScreenItem (Page 132)

Propriété Closeable

Peut être fermé - Closeable

Détermine si le contrôle peut être fermé en Runtime.

Valeur	Explication
TRUE	Le contrôle peut être fermé en Runtime.
FALSE	Le contrôle ne peut pas être fermé en Runtime.

L'attribut est dynamisable avec le nom **Closeable**. Le type de données est BOOLEAN.

Propriété CloseButton

Description

TRUE, lorsque la fenêtre possède un bouton "Quitter". Accès en lecture uniquement.

Voir aussi

Fenêtre de vue (Page 188)
Fenêtre d'application (Page 183)
Objet ScreenItem (Page 132)

Co

Propriété CoarseGrid

Description

TRUE, lorsque l'axe des valeurs a de longues graduations. La distance entre deux longues graduations peut être modifiée via la propriété "CoarseGridValue". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété CoarseGridX

Description

TRUE, lorsque l'axe X a de longues graduations. La distance entre deux longues graduations peut être modifiée via la propriété "CoarseGridValueX". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété CoarseGridY

Description

TRUE, lorsque l'axe Y a de longues graduations. La distance entre deux longues graduations peut être modifiée via la propriété "CoarseGridValueY". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété CoarseGridValue

Description

Détermine la distance entre deux longues graduations. L'indication sera ou non exploitée selon la valeur de la propriété "CoarseGrid".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété CoarseGridValueX

Description

Détermine la distance entre deux graduations longues de l'axe X ou la restitue. L'indication sera ou non exploitée selon la valeur de la propriété "CoarseGridX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété CoarseGridValueY

Description

Détermine la distance entre deux graduations longues de l'axe Y ou la restitue. L'indication sera ou non exploitée selon la valeur de la propriété "CoarseGridY".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété CollectValue

Description

Contient en Runtime comme valeur d'entrée l'état actuel des classes d'alarmes actives. LONG Accès en lecture et en écriture.
La dynamisation via une variable permet par exemple de déterminer la valeur à partir de la signalisation groupée de vues inférieures du point de vue hiérarchique.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété ColMove

Description

TRUE, lorsque la disposition des colonnes peut être modifiée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété Color

Description

La propriété "Index" référence un couple de colonnes ou une courbe. "Color" définit la couleur du texte utilisée dans ce couple de colonnes ou bien la couleur de la courbe. LONG Accès en écriture et en lecture. La couleur doit être spécifiée sous la forme RVB.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ColorAlarmHigh

Description

Définit la couleur de bargraphe pour le seuil "AlarmHigh" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckAlarmHigh" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorAlarmLow

Description

Détermine la couleur du bargraphe pour le seuil "AlarmLow" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckAlarmLow" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorBottom

Description

Définit la couleur pour la butée inférieure/droite de l'objet Slider ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Slider (Page 219)

Objet ScreenItem (Page 132)

Propriété ColorChangeType

Description

TRUE, lorsqu'en cas de changement de couleur (par exemple lorsqu'un seuil est atteint) la modification doit avoir lieu segment par segment. Avec le paramètre FALSE, le changement de couleur est valable pour la totalité du bargraphe. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorLimitHigh4

Description

Définit la couleur pour le seuil supérieur "Reserve 4" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckLimitHigh4" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorLimitHigh5

Description

Définit la couleur pour le seuil supérieur "Reserve 5" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckLimitHigh5" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorLimitLow4

Description

Définit la couleur pour le seuil inférieur "Reserve 4" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "ColorLimitLow4" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorLimitLow5

Description

Définit la couleur pour le seuil inférieur "Reserve 5" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckLimitLow5" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorToleranceHigh

Description

Définit la couleur pour le seuil supérieur "ToleranceHigh" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckToleranceHigh" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorToleranceLow

Description

Définit la couleur pour le seuil inférieur "ToleranceLow" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckToleranceLow" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorTop

Description

Définit la couleur pour la butée supérieure/gauche de l'objet Slider ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Slider (Page 219)

Objet ScreenItem (Page 132)

Propriété ColorWarningHigh

Description

Définit la couleur pour le seuil supérieur "WarningHigh" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckWarningHigh" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColorWarningLow

Description

Définit la couleur pour le seuil inférieur "WarningLow" ou la restitue. LONG Accès en lecture et en écriture.
La propriété "CheckWarningLow" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ColTitle

Description

TRUE, lorsque les colones dans la fenêtre d'alarmes doivent posséder une barre de titre.
BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété ColumnAdd

Appliquer - ColumnAdd

Accepte dans la liste des colonnes sélectionnées la colonne sélectionnée dans la liste des colonnes disponibles.

L'attribut est dynamisable avec le nom **ColumnAdd**. Le type de données est STRING.

Propriété ColumnAlias

ColumnAlias

Cet attribut indique l'alias défini dans l'archive utilisateur pour le nom de la colonne.

L'attribut est dynamisable avec le nom **ColumnAlias**. Le type de données est STRING.

Propriété ColumnAlign

Alignement - ColumnAlign

Spécifie la façon dont s'aligne la colonne sélectionnée.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	à gauche	La colonne sélectionnée est affichée à gauche.
1	centré	La colonne sélectionnée est centrée.
2	à droite	La colonne sélectionnée est affichée à droite.

L'attribut est dynamisable avec le nom **ColumnAlign**. Le type de données est LONG.

Propriété ColumnAutoPrecisions

Chiffres après la virgule Automatique - ColumnAutoPrecisions

Détermine si le nombre de chiffres après la virgule est déterminé automatiquement.

Valeur	Explication
TRUE	Le nombre de chiffres après la virgule est défini automatiquement. La valeur du champ "Chiffres après la virgule" reste sans effet.
FALSE	La valeur du champ "Chiffres après la virgule" est effective.

L'attribut est dynamisable avec le nom **ColumnAutoPrecisions**. Le type de données est BOOLEAN.

Propriété ColumnCaption

Désignation - ColumnCaption

Détermine la désignation pour la colonne sélectionnée.

L'attribut est dynamisable avec le nom **ColumnCaption**. Le type de données est STRING.

Propriété ColumnCount

ColumnCount

Cet attribut indique le nombre de colonnes configurées.

L'attribut est dynamisable avec le nom **ColumnCount**. Le type de données est LONG.

Propriété ColumnDateFormat

Format de la date - ColumnDateFormat

Détermine quel format de la date sera utilisé pour l'affichage.

Les formats de date suivants sont disponibles :

Valeur	Explication
Automatique	Le format de la date est automatiquement défini.
dd.MM.yy	Jour.Mois.Année, par exemple 24.12.07.
dd.MM.yyyy	Jour.Mois.Année, par exemple 24.12.2007.
dd/MM/yy	Jour/Mois/Année, par exemple 24/12/07.
dd/MM/yyyy	Jour/Mois/Année, par exemple 24/12/2007.

L'attribut est dynamisable avec le nom **ColumnDateFormat**. Le type de données est STRING.

Propriété ColumnDMVarName

ColumnDMVarName

Cet attribut indique les noms des variables que vous avez affectées dans l'archive utilisateur de la colonne.

L'attribut est dynamisable avec le nom **ColumnDMVarName**. Le type de données est STRING.

Propriété ColumnExponentialFormat

Représentation exponentielle - ColumnExponentialFormat

Détermine si les valeurs de la colonne sélectionnée s'affichent en représentation exponentielle.

Valeur	Explication
TRUE	Les valeurs sont affichées en représentation exponentielle.
FALSE	Les valeurs sont affichées en représentation décimale.

L'attribut est dynamisable avec le nom **ColumnExponentialFormat**. Le type de données est BOOLEAN.

Propriété ColumnFlagNotNull

ColumnFlagNotNull

Cet attribut détermine si le champ de l'archive utilisateur affecté à la colonne doit avoir une valeur.

Valeur	Explication
Oui	La valeur de la colonne doit posséder une valeur.
Non	La valeur de la colonne peut posséder une valeur.

L'attribut est dynamisable avec le nom **ColumnFlagNotNull**. Le type de données est BOOLEAN.

Propriété ColumnFlagUnique

ColumnFlagUnique

Cet attribut détermine si le champ de l'archive utilisateur affecté à la colonne doit avoir une valeur univoque. Les valeurs dans cette colonne doivent être distinctes.

Valeur	Explication
TRUE	La valeur de la colonne doit posséder une valeur univoque.
FALSE	La valeur de la colonne ne doit posséder aucune valeur univoque.

L'attribut est dynamisable avec le nom **ColumnFlagUnique**. Le type de données est BOOLEAN.

Propriété ColumnHideText

Contenu comme texte - ColumnHideText

Détermine si le contenu de la colonne sélectionnée doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le contenu ne s'affiche pas sous forme de texte. L'option n'est pas activée.
FALSE	Le contenu s'affiche sous forme de texte. L'option est activée.

L'attribut est dynamisable avec le nom **ColumnHideText**. Le type de données est BOOLEAN.

Propriété ColumnHideTitleText

Titre comme texte - ColumnHideTitleText

Détermine si le titre de la colonne sélectionnée doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le titre ne s'affiche pas sous forme de texte. L'option n'est pas activée.
FALSE	Le titre s'affiche sous forme de texte. L'option est activée.

L'attribut est dynamisable avec le nom **ColumnHideTitleText**. Le type de données est BOOLEAN.

Propriété ColumnIndex

ColumnIndex

Cet attribut fait référence à une colonne du contrôle. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres propriétés à une colonne précise.

Les valeurs admissibles pour "ColumnIndex" et "ColumnCount" se situent entre 0 et -1. L'attribut "ColumnCount" indique le nombre de colonnes existantes.

L'attribut "ColumnIndex" est dynamisable via l'attribut **ColumnIndex**. Le type de données est LONG.

Propriété ColumnLeadingZeros

Avec zéros non significatifs - ColumnLeadingZeros

Détermine si les valeurs de la colonne sélectionnée s'affichent avec des zéros non significatifs. Vous déterminez le nombre des zéros non significatifs via "Nombre de positions" ou "ColumnLeadingZeros". Le nombre maximal s'élève à "11". La valeur "0" a pour effet que les zéros non significatifs ne s'affichent pas. L'option "Avec zéros non significatifs" est désactivée.

L'attribut est dynamisable avec le nom **ColumnLeadingZeros**. Le type de données est LONG.

Propriété ColumnLength

Longueur en caractères - ColumnLength

Détermine la largeur de la colonne sélectionnée.

L'attribut est dynamisable avec le nom **ColumnLength**. Le type de données est LONG.

Propriété ColumnMaxValue

ColumnMaxValue

Cet attribut indique la valeur maximale définie dans l'archive utilisateur pour la colonne.

L'attribut est dynamisable avec le nom **ColumnMaxValue**. Le type de données est STRING.

Propriété ColumnMinValue

ColumnMinValue

Cet attribut indique la valeur minimale définie dans l'archive utilisateur pour la colonne.

L'attribut est dynamisable avec le nom **ColumnMinValue**. Le type de données est STRING.

Propriété ColumnName

ColumnName

Cet attribut indique le nom de la colonne référencée via l'attribut "ColumnIndex".

L'attribut est dynamisable avec le nom **ColumnName**. Le type de données est STRING.

Propriété ColumnPosition

ColumnPosition

Cet attribut affiche la position définie dans l'archive utilisateur pour le champ.

L'attribut est dynamisable avec le nom **ColumnPosition**. Le type de données est LONG.

Propriété ColumnPrecisions

Chiffres après la virgule - ColumnPrecisions

Définit le nombre de chiffres après la virgule des valeurs de la colonne sélectionnée. Vous ne pouvez entrer la valeur que si l'option "Automatique" est désactivée.

L'attribut est dynamisable avec le nom **ColumnPrecisions**. Le type de données est SHORT.

Propriété ColumnReadAccess

ColumnReadAccess

Cet attribut indique l'autorisation définie dans l'archive utilisateur pour l'accès en lecture à la colonne. Le numéro correspond à celui affecté dans l'éditeur "User Administrator" de l'autorisation.

L'attribut est dynamisable avec le nom **ColumnReadAccess**. Le type de données est LONG.

Propriété ColumnReadonly

Protégé en écriture - ColumnReadonly

Détermine si la colonne sélectionnée est protégée en écriture.

Valeur	Explication
TRUE	La colonne est protégée en écriture.
FALSE	La colonne n'est pas protégée en écriture. Vous pouvez modifier les valeurs de la colonne en Runtime si vous activez l'option "Modifier" sur l'onglet "Général".

L'attribut est dynamisable avec le nom **ColumnReadonly**. Le type de données est BOOLEAN.

Propriété ColumnRemove

Supprimer - ColumnRemove

Supprime la colonne sélectionnée de la liste des colonnes sélectionnées et l'insère dans la liste des colonnes disponibles.

L'attribut est dynamisable avec le nom **ColumnRemove**. Le type de données est STRING.

Propriété ColumnRepos

Haut/bas - ColumnRepos

Modifie l'ordre des colonnes. "Haut" et "Bas" déplacent la colonne sélectionnée dans la liste vers le haut ou le bas. Ce qui a pour conséquence que la colonne dans le tableau est positionnée plus en avant ou plus en arrière.

L'attribut est dynamisable avec le nom **ColumnRepos**. Le type de données est LONG.

Propriété ColumnResize

Largeur modifiable - ColumnResize

Détermine si la largeur des colonnes peut être modifiée.

Valeur	Explication
TRUE	Vous pouvez modifier la largeur des colonnes.
FALSE	Vous ne pouvez pas modifier la largeur des colonnes.

L'attribut est dynamisable avec le nom **ColumnResize**. Le type de données est BOOLEAN.

Propriété ColumnSrollbar

Barre de défilement des colonnes - ColumnSrollbar

Détermine si les barres de défilement des colonnes s'affichent ou non.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	non	Les barres de défilement des colonnes ne s'affichent pas.
1	si besoin est	Les barres de défilement des colonnes s'affichent lorsque l'encombrement du contrôle en sens vertical est plus grand que la plage d'affichage disponible.
2	toujours	Les barres de défilement des colonnes s'affichent toujours.

L'attribut est dynamisable avec le nom **ColumnScrollbar**. Le type de données est LONG.

Propriété ColumnShowDate

Affichage de la date - ColumnShowDate

Détermine si le bloc "Heure" s'affiche dans un champ avec l'heure et la date.

Valeur	Explication
TRUE	La date et l'heure s'affichent. Le format de la date est déterminé dans le champ "Format de la date".
FALSE	L'heure est affichée.

L'attribut est dynamisable avec le nom **ColumnShowDate**. Le type de données est BOOLEAN.

Propriété ColumnShowIcon

Contenu comme symbole - ColumnShowIcon

Détermine si le contenu de la colonne sélectionnée doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le contenu s'affiche sous forme de symbole.
FALSE	Le contenu ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **ColumnShowIcon**. Le type de données est BOOLEAN.

Propriété ColumnShowTitleIcon

Titre comme symbole - ColumnShowTitleIcon

Détermine si le titre de la colonne sélectionnée doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le titre s'affiche sous forme de symbole.
FALSE	Le titre ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **ColumnShowTitleIcon**. Le type de données est BOOLEAN.

Propriété ColumnSort

ColumnSort

Cet attribut définit comment la colonne de l'archive utilisateur référencée dans l'attribut "ColumnIndex" est triée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	non	Aucun tri
1	croissant	Tri croissant de la plus petite à la plus grande valeur
2	décroissant	Tri décroissant de la plus grande à la plus petite valeur

L'attribut est dynamisable avec le nom **ColumnSort** . Le type de données est LONG.

Propriété ColumnSortIndex

ColumnSortIndex

Cet attribut indique l'ordre de tri de la colonne référencée dans "ColumnIndex". Si vous activez la valeur sur "0", le critère de tri est supprimé dans "ColumnSort".

L'attribut est dynamisable avec le nom **ColumnSortIndex**. Le type de données est LONG.

Propriété ColumnStartValue

ColumnStartValue

Cet attribut indique la valeur initiale définie dans l'archive utilisateur pour la colonne.

L'attribut est dynamisable avec le nom **ColumnStartValue**. Le type de données est STRING.

Propriété ColumnStringLength

ColumnStringLength

Cet attribut affiche la longueur de la chaîne de caractères définie dans l'archive utilisateur pour la colonne.

L'attribut est dynamisable avec le nom **ColumnStringLength**. Le type de données est LONG.

Propriété ColumnTimeFormat

Format de l'heure - ColumnTimeFormat

Détermine quel format de l'heure sera utilisé pour l'affichage.

Les formats d'heure suivants sont disponibles :

Valeur	Explication
Automatique	Le format de l'heure est automatiquement défini.
HH:mm:ss.ms	Heures:Minutes:Secondes.Millisecondes, par exemple 15:35:44.240.
hh:mm:ss tt	Heures:Minutes:Secondes AM/PM, par exemple 03:35:44 PM.
hh:mm:ss.ms tt	Heures:Minutes:Secondes.Millisecondes AM/PM, par exemple 03:35:44.240 PM.

L'attribut est dynamisable avec le nom **ColumnTimeFormat**. Le type de données est STRING.

Propriété ColumnTitleAlign

Alignement du titre de colonne - ColumnTitleAlign

Détermine comment les titres de colonnes sont alignés.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	à gauche	Les titres de colonnes sont alignés à gauche.
1	centré	Les titres de colonnes sont centrés.
2	à droite	Les titres de colonnes sont alignés à droite.
3	comme le contenu du tableau	Les titres de colonnes sont alignés comme le contenu correspondant de la colonne.

L'attribut est dynamisable avec le nom **ColumnTitleAlign**. Le type de données est LONG.

Propriété ColumnTitles

Afficher les titres des colonnes - ColumnTitles

Spécifie si le titre de colonne s'affiche.

Valeur	Explication
TRUE	Le titre de colonne s'affiche.
FALSE	Le titre de colonne ne s'affiche pas.

L'attribut est dynamisable avec le nom **ColumnTitles**. Le type de données est BOOLEAN.

Propriété ColumnType

Type - ColumnType

Cet attribut indique le type de données défini dans l'archive utilisateur pour la colonne sélectionnée.

L'attribut est dynamisable avec le nom **ColumnType**. Le type de données est LONG.

Propriété ColumnVisible

ColumnVisible

Détermine si la colonne référencée via l'attribut "ColumnIndex" est affichée.

Valeur	Explication
TRUE	La colonne s'affiche.
FALSE	La colonne ne s'affiche pas.

L'attribut est dynamisable avec le nom **ColumnVisible** . Le type de données est BOOLEAN.

Propriété ColumnWriteAccess

ColumnWriteAccess

Cet attribut indique l'autorisation définie dans l'archive utilisateur pour l'accès en écriture à la colonne. Le numéro correspond à celui affecté dans l'éditeur "User Administrator" de l'autorisation.

L'attribut est dynamisable avec le nom **ColumnWriteAccess**. Le type de données est LONG.

Propriété ColWidth

Description

TRUE, lorsque la largeur des colonnes de la fenêtre d'alarmes doit pouvoir être modifiée. La largeur des colonnes ne peut cependant être modifiée que si la propriété "AutoScroll" n'est pas activée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété Command

Description

TRUE, lorsqu'il faut forcer la mise à jour des valeurs affichées dans Control.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété Comment

Description

Lit ou définit le commentaire de l'objet d'alarme.

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété CommonTime

Description

TRUE, lorsqu'une colonne temps commune est utilisée dans la fenêtre de tableau. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété CommonX

Description

TRUE, lorsque les courbes de la fenêtre des courbes sont représentées avec un axe X commun. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété CommonY

Description

TRUE, lorsque les courbes de la fenêtre des courbes sont représentées avec un axe Y commun. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Propriété ComputerName

Description

Rend le nom de l'ordinateur sur lequel l'objet d'alarme a été déclenché.

ComputerName (readonly)

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété Context

Description

Lit ou définit le préfixe de serveur de l'objet d'alarme.

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété ConnectTrendWindows

Fenêtres de courbes liées - ConnectTrendWindows

Détermine si les fenêtres de courbes configurées sont liées. Ceci n'est possible que si vous avez configuré plusieurs fenêtres de courbes.

Les fenêtres de courbes liées possèdent les propriétés suivantes :

- Elles peuvent avoir un axe X commun.
- Elles ont une barre de défilement.

- Elles ont une règle.
- Les fonctions de zoom d'une fenêtre de courbes ont une influence sur les fenêtres de courbes liées.

Valeur	Description
TRUE	Toutes les fenêtres de courbes configurées sont liées.
FALSE	Les fenêtres de courbes sont représentées séparément.

L'attribut est dynamisable avec le nom **ConnectTrendWindows**. Le type de données est BOOLEAN.

Propriété ContinousChange

Description

Définit le type de transmission de la valeur réglée avec le curseur (Propriété "Position") dans Runtime:

- FALSE: La valeur de la propriété "Position" n'est transmise qu'après relâchement de la touche de souris.
- TRUE: La valeur de la propriété "Position" est transmise dès que la position du curseur est modifiée.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété Count

Description

Renvoie le nombre d'éléments figurant sur une liste.

INTEGER (Accès en lecture uniquement).

Exemple

L'exemple montre comment s'affiche le nombre des objets de la liste DataSet.

```
'VBS165
HMIRuntime.Trace "Count: " & HMIRuntime.DataSet.Count & vbNewLine
```

Dans l'exemple suivant deux variables sont insérées dans la liste TagSet et la propriété Count s'affiche en tant que Trace.

```
'VBS177
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
HMIRuntime.Trace "Count: " & group.Count & vbNewLine
```

Voir aussi

Méthode CreateTagSet (Page 687)

Objet TagSet (Liste) (Page 149)

Objet ScreenItems (Liste) (Page 136)

Objet Screens (Liste) (Page 141)

Objet Layers (Liste) (Page 128)

Objet DataSet (Liste) (Page 123)

Objet ProcessValues (liste) (Page 131)

Cu

Propriété CurrentContext

Description

Dans une fenêtre de vue est lu le serveur duquel vient la vue qui contient le Script.

La propriété "CurrentContext" peut donner des résultats différents: Dans une vue de base locale, on distingue deux cas lorsqu'il y a une fenêtre de vue représentant par exemple une vue d'un serveur:

- La propriété "CurrentContext" est utilisée dans une action de la vue dans la fenêtre de vue: En résultat, le nom symbolique de l'ordinateur (propriété Package) est restitué complété par deux deux-points "WinCCProject_MyComputer::" par exemple.
- La propriété "CurrentContext" est utilisée dans une action de la vue de base: En résultat, une chaîne de caractères vide est restituée.

Voir aussi

Objet HMIRuntime (Page 125)

Propriété Cursor

Description

Commande l'apparence du curseur en Runtime lorsqu'il se trouve au-dessus du symbole.

- 0: Le curseur a la forme d'une flèche et ne change pas d'apparence lorsqu'il est positionné au-dessus du symbole.
- 1: Le curseur est représenté par une flèche tridimensionnelle accompagnée d'un symbole éclair vert. Cela signale en Runtime que l'objet en question est manipulable.

Voir aussi

Objet ScreenItem (Page 132)

HMI Symbol Library (Page 246)

Propriété Cursor

Pointeur de la souris (Cursor)

Détermine si le pointeur de la souris s'affiche sur l'icône en Runtime.

Valeur	Explication
TRUE	Le pointeur de la souris est affiché en Runtime lorsqu'il est positionné sur l'icône.
FALSE	Le pointeur de la souris n'est pas affiché en Runtime lorsqu'il est positionné sur l'icône.

L'attribut est dynamisable avec le nom **Cursor**. Le type de données est BOOLEAN.

Propriété CursorControl

Description

TRUE, lorsqu'en mode curseur alpha ce dernier, après avoir quitté le champ, rejoint le champ suivant selon la séquence TAB. BOOLEAN Accès en écriture et en lecture.

La propriété "CursorMode" doit pour cela être positionnée sur TRUE.

Voir aussi

Liste de textes (Page 206)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété CurveForm

Description

WinCC Function Trend Control

Détermine comment les points de mesure d'une courbe référencée via la propriété "Index" doivent être liés. Accès en écriture et en lecture.

WinCC Online Trend Control

La propriété "Index" référence une courbe. "CurveForm" détermine comment les points de mesure doivent être liés.

- 0x00000012: Représentation des points de mesure.
- 0x00000014: Les points de mesure sont liés par un trait.
- 0x00000011: Les points de mesure sont liés via une courbe en escalier.
- 0x00000021: La surface est remplie sous la courbe dont les points sont liés par un trait.
- 0x00000022: La surface est remplie sous la courbe en escalier.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété CursorMode

Description

Si le "CursorMode" est mis sur "oui", vous pouvez afficher par page toutes les alarmes de l'archive cyclique dans la liste des archives longue durée. La propriété "CursorModePrefetch" vous permet de définir le nombre d'alarmes affichées par page.

L'option "Défilement automatique" doit être désactivée pour pouvoir passer entre les pages. Accès en écriture et en lecture.

Propriété CursorModePrefetch

Description

Définit le nombre d'alarmes que vous voulez afficher par page de toutes les alarmes de l'archive cyclique dans la liste des archives longue durée.

La propriété d'objet "CursorMode" doit être mise sur "oui".

Accès en écriture et en lecture.

1.14.4.5 D

Da

Propriété DangerColor

Description

Détermine la couleur de la zone de danger sur l'échelle ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété Danger

Description

Détermine la valeur graduée à partir de laquelle commence la "zone de danger" ou la restitue. La zone s'étend de la valeur "Danger" jusqu'à la fin de l'échelle. Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété DataFormat

Description

Restitue le type de données de l'objet IOField. Accès en lecture uniquement.

Plage des valeurs de 0 à 3.

0: Binaire

1: Décimal

2: Chaîne de caractères

3: Hexadécimal

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété DataIndex

Description

Restitue l'indice actuel des données de la courbe actuelle.

Remarque

La propriété est prise en charge uniquement pour les Controls avant WinCC V7.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DataLogs

Description

Renvoie un objet de type "DataLogs".

DataLogs (read-only)

Voir aussi

Objet DataLogs (Page 122)

Objet HMIRuntime (Page 125)

Propriété DataSet

Description

Renvoie un objet de type "DataSet".

DataSet (read-only)

Voir aussi

Objet DataSet (Liste) (Page 123)

Objet HMIRuntime (Page 125)

Propriété DataX

Description

Insère un enregistrement unique et doit être positionné avant l'appel de "InsertData".

Remarque

La propriété est prise en charge uniquement pour les Controls avant WinCC V7.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DataXY

Description

Insère plusieurs enregistrements en tant qu'Array avec des couples de valeurs et doit être positionné avant l'appel de "InsertData".

Les données dans Array sont validées si "DataX" est de type VT_EMPTY. Faute de quoi le système utilise dans l'attribut "InsertData" le couple de valeurs émanant de "DataX" et "DataY".

Remarque

La propriété est prise en charge uniquement pour les Controls avant WinCC V7.

Voir aussi

Exemple : Pour appeler les méthodes d'un ActiveX-Control (Page 807)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DataY

Description

Insère un enregistrement unique et doit être positionné avant l'appel de "InsertData".

Remarque

La propriété est prise en charge uniquement pour les Controls avant WinCC V7.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

De - Do

Propriété DefaultMsgFilterSQL

DefaultMsgFilterSQL

Cet attribut définit une sélection fixe des alarmes pour une instruction SQL.

Si vous définissez des sélections personnalisées supplémentaires via l'attribut "MsgFilterSQL", les instructions SQL de "DefaultMsgFilterSQL" et "MsgFilterSQL" sont reliées par "AND".

L'attribut est dynamisable avec le nom **DefaultMsgFilterSQL**. Le type de données est STRING.

Propriété DefaultPrecision

Description

Cet attribut définit le nombre de chiffres après la virgule par défaut que l'on utilise pour spécifier les valeurs de mise à l'échelle. Accès en écriture et en lecture.

Propriété DefaultRulerPrecision

Description

Cet attribut définit le nombre de chiffres après la virgule par défaut pour l'affichage d'une valeur mesurée lorsqu'elle est relevée via la fonction "Afficher valeur à cet endroit". Accès en écriture et en lecture.

Propriété DefaultSort

Tri standard - DefaultSort

Détermine le tri standard dans les colonnes de tableau.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	croissant	La liste est mise à jour à partir de la ligne la plus basse.
1	décroissant	La liste est mise à jour à partir de la ligne la plus haute.

L'attribut est dynamisable avec le nom **DefaultSort**. Le type de données est LONG.

Propriété DefaultSort2

DefaultSort2

La fonction définit le tri des colonnes du tableau si vous ne souhaitez pas utiliser le tri standard dans l'ordre "Date/Heure/Numéro". Vous devez entrer à la place dans la propriété d'objet "DefaultSort2Column" un bloc d'alarme qui effectue un tri par "Bloc d'alarme/Date/Heure/Numéro".

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	croissant	La liste est mise à jour à partir de la ligne la plus basse.
1	décroissant	La liste est mise à jour à partir de la ligne la plus haute.

L'attribut est dynamisable avec le nom **DefaultSort2**. Le type de données est LONG.

Propriété DefaultSort2Column

DefaultSort2Column

La fonction définit le tri des colonnes du tableau si vous ne souhaitez pas utiliser le tri standard dans l'ordre "Date/Heure/Numéro".

Indiquez un bloc d'alarme avec son nom d'objet.

Les colonnes du tableau sont alors triées dans l'ordre "Bloc d'alarme/Date/Heure/Numéro".

L'attribut est dynamisable avec le nom **DefaultSort2Column**. Le type de données est STRING.

Propriété DeleteData

Description

Supprime les données du tampon de données de la courbe actuelle.

TRUE: toutes les données de la courbe sont supprimées.

FALSE: le couple de valeurs pour la position "DataIndex" est supprimé.

Remarque

La propriété est prise en charge uniquement pour les Controls avant WinCC V7.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété Delta

Description

Détermine l'écart de valeur entre deux graduations principales ou la restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété DesiredCurveColor

Description

Détermine la couleur d'une courbe de consignes qui appartient à une courbe référencée via la propriété "Index". La couleur doit être spécifiée sous la forme RVB. L'indication sera ou non exploitée selon la valeur de la propriété "DesiredCurveVisible".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DesiredCurveCurveForm

Description

Détermine la forme d'une courbe de consignes qui appartient à une courbe référencée via la propriété "Index". L'indication sera ou non exploitée selon la valeur de la propriété "DesiredCurveVisible".

0x00000011: Les points de mesure sont liés par une courbe en escalier à trait plein

0x00000012: Représentation des points de mesure.

0x00000014: Les points de mesure sont représentés de façon linéaire par un trait plein

0x00000021: La surface est remplie sous la courbe dont les points sont liés par un trait.

0x00000022: La surface est remplie sous la courbe en escalier.

0x00000031: Les points de mesure sont liés par une courbe en escalier à trait discontinu

0x00000032: Les points de mesure sont représentés de façon linéaire par un trait discontinu

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DesiredCurveSourceNumberOfUAValues

Description

Détermine le nombre de couples de valeurs d'une courbe de consignes qui appartient à une courbe référencée via la propriété "Index". L'indication sera ou non exploitée selon la valeur de la propriété "DesiredCurveVisible".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DesiredCurveSourceUAArchive

Description

Détermine le nom de l'archive utilisateur à partir de laquelle les valeurs sont lues, ces valeurs de la courbe de consignes appartenant à une courbe référencée via "Index". L'indication sera ou non exploitée selon la valeur de la propriété "DesiredCurveVisible".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DesiredCurveSourceUAArchiveStartID

Description

Détermine la position initiale pour les valeurs de la courbe de consignes appartenant à une courbe référencée via "Index", position à partir de laquelle sont lues les valeurs dans l'archive utilisateur. L'indication sera ou non exploitée selon la valeur de la propriété "DesiredCurveVisible".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DesiredCurveSourceUAColumnX

Description

Détermine la colonne dans l'archive utilisateur à partir de laquelle les valeurs X sont lues, ces valeurs X de la courbe de consignes appartenant à une courbe référencée via "Index" . L'indication sera ou non exploitée selon la valeur de la propriété "DesiredCurveVisible".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DesiredCurveSourceUAColumnY

Description

Détermine la colonne dans l'archive utilisateur à partir de laquelle les valeurs Y sont lues, ces valeurs Y de la courbe de consignes appartenant à une courbe référencée via "Index" . L'indication sera ou non exploitée selon la valeur de la propriété "DesiredCurveVisible".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété DesiredCurveVisible

Description

TRUE, lorsque la courbe de consignes qui appartient à une courbe référencée via "Index" est affichée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété Direction

Description

Détermine la direction du bargraphe ou la position de l'objet Slider ou la restitue. BOOLEAN Accès en écriture et en lecture. Plage des valeurs de 0 à 3.

0 = haut

1 = bas

2 = gauche

3 = droite

Voir aussi

Slider (Page 219)

Bargraphe (Page 184)

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété DisplayOptions

Afficher des alarmes - DisplayOptions

Détermine quelles alarmes seront affichées.

Les possibilités sont les suivantes :

Valeur	Désignation
0	Toutes les alarmes
1	Uniquement les alarmes affichées
2	Uniquement les alarmes masquées

L'attribut est dynamisable avec le nom **DisplayOptions**. Le type de données est LONG.

Propriété DisplayOptions (avant WinCC V7)

Description

Établit qu'un graphique, un texte ou les deux sont assignés à un bouton.

0 Graphique ou texte : si un graphique est disponible, il est assigné au bouton, sinon un texte est assigné.
1 Graphique et texte
2 Texte seulement
3 Graphique seulement

Propriété DoubleClickAction

Action par double clic - DoubleClickAction

Détermine l'action qui sera effectuée en Runtime lors du double clic sur une ligne d'alarme.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	aucune	Aucune action ne sera exécutée.
1	Loop in Alarm	La fonction "Loop in Alarm" est appelée.
2	Ouvrir le dialogue du commentaire	La fonction de touche "Dialogue du commentaire" est appelée.
3	Ouvrir le dialogue Texte d'info	La fonction de touche "Dialogue du texte d'info" est appelée.
4	Selon les colonnes	L'action exécutée se base sur la colonne que vous avez sélectionnée par un double-clic.

L'attribut est dynamisable avec le nom **DoubleClickAction**. Le type de données est LONG.

1.14.4.6 e

Propriété Edit

Description

Active le mode d'édition pour une cellule, dans la mesure où la propriété "Editable" est positionnée sur TRUE dans la colonne correspondante.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété Editable

Description

La propriété "Index" référence un couple de colonnes. "Editable" détermine si ce couple de colonnes doit être éditable. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété EditAtOnce

Description

TRUE, lorsque l'entrée peut immédiatement avoir lieu sans aucune autre action lors du saut sur le champ avec la touche <Tab>. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Liste de textes (Page 206)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété Enabled

Fonction

Autorise ou verrouille un objet pour la conduite ou affiche la valeur correspondante. TRUE: Autoriser la conduite, FALSE: Verrouiller la conduite.

BOOLEAN Accès en écriture et en lecture.

Exemple

L'exemple suivant verrouille tous les objets de la vue "NewPDL1":

```
'VBS71
```

```
Dim objScreen
Dim objScrItem
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems.Item(lngIndex).ObjectName    'Read names of objects
Set objScrItem = objScreen.ScreenItems(strName)
objScrItem.Enabled=False    'Lock object
Next
```

Voir aussi

Objet Screen (Page 138)

Objet ScreenItem (Page 132)

Propriété EnableDelete

Supprimer - EnableDelete

Détermine si vous pouvez supprimer des données de l'archive utilisateur en Runtime.

Valeur	Explication
TRUE	Vous pouvez supprimer des données de l'archive utilisateur en Runtime.
FALSE	Vous ne pouvez supprimer aucune donnée de l'archive utilisateur en Runtime.

L'attribut est dynamisable avec le nom **EnableDelete**. Le type de données est BOOLEAN.

Propriété EnableEdit

Modifier - EnableEdit

Détermine si vous pouvez modifier les données affichées en Runtime.

Valeur	Explication
TRUE	Vous pouvez modifier les données en Runtime.
FALSE	Vous ne pouvez pas modifier les données en Runtime.

L'attribut est dynamisable avec le nom **EnableEdit**. Le type de données est BOOLEAN.

Propriété EnableInsert

Ajouter - EnableInsert

Détermine si vous pouvez ajouter des données à l'archive utilisateur en Runtime.

Valeur	Explication
TRUE	Vous pouvez ajouter des données à l'archive utilisateur en Runtime.
FALSE	Vous ne pouvez ajouter aucune donnée à l'archive utilisateur en Runtime.

L'attribut est dynamisable avec le nom **EnableInsert**. Le type de données est BOOLEAN.

Propriété EnablePopupMenu

EnablePopupMenu

Définit si le menu contextuel est activé dans le contrôle.

L'attribut est dynamisable avec le nom **EnablePopupMenu**. Le type de données est BOOLEAN.

Propriété EndAngle

Description

Détermine la fin de l'objet ou la restitue. L'indication se fait dans le sens horaire, en degrés, en commençant à 12:00 h.

Voir aussi

Segment de cercle (Page 162)

Arc de cercle (Page 160)

Segment d'ellipse (Page 156)

Arc d'ellipse (Page 154)

Objet ScreenItem (Page 132)

Propriété EndTime

Description

Control Online Table

L'attribut "Index" référence un couple de colonnes. "EndTime" détermine la fin de la représentation de ce couple de colonnes. L'indication sera ou non exploitée selon les propriétés "TimeRange" et "CommonTime". Accès en écriture et en lecture.

Control Online Trend

L'attribut "Index" référence une courbe. "EndTime" détermine la fin de la représentation de cette courbe. L'indication sera ou non exploitée selon les propriétés "Autorange", "TimeRange" et "CommonX".

Si vous dynamisez la plage de temporisation, utilisez le format "aaaa-mm-jj hh:mm:ss".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété EndValue

Description

La propriété "Index" référence une courbe. "EndValue" détermine pour cette courbe le seuil supérieur de la plage des valeurs à représenter. L'indication sera ou non exploitée selon les propriétés "Autorange" et "CommonY".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété EndX

Description

Détermine le seuil supérieur de l'axe X d'une courbe référencée avec "Index". L'indication sera ou non exploitée selon les propriétés "AutorangeX" et "CommonX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété EndY

Description

Détermine le seuil supérieur de l'axe Y d'une courbe référencée avec "Index". L'indication sera ou non exploitée selon les propriétés "AutorangeY" et "CommonY".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ErrorDescription

Fonction

Description des erreurs de la propriété "LastError". La description des erreurs n'apparaît qu'en anglais.

STRING (readonly)

Les messages d'erreur suivants sont définis :

Sortie	Description
" "	OK
"Operation Failed"	Erreur d'exécution
"Variable not found"	Erreur de variable
"Server down"	Serveur non disponible
"An error occured for one or several tags"	Multi Tag Error (Erreur pour une ou plusieurs variables)

Pour qu'ErrorDescription restitue une valeur, il faut tout d'abord exécuter un Read.

Le message "Multi Tag Error" est émis si une erreur se produit via l'objet TagSet en lisant ou en écrivant plusieurs variables. Il vous faut analyser la propriété ErrorDescription de chacune des variables pour déterminer la variable concernée par l'erreur ainsi que le type d'erreur.

Exemple

L'exemple suivant émet la description d'erreur pour la variable "Tag1" :

```
'VBS72
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objtag.Read
MsgBox objTag.ErrorDescription
```

Dans l'exemple suivant deux variables sont insérées dans la liste TagSet et la propriété ErrorDescription s'affiche en tant que Trace.

```
'VBS179
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
```

```
group.Add "Motor2"
HMIRuntime.Trace "ErrorDescription: " & group.ErrorDescription & vbNewLine
```

Vous accédez comme suit à la propriété ErrorDescription d'une variable contenue dans la liste :

```
HMIRuntime.Trace "ErrorDescription: " & group("Motor1").ErrorDescription & vbNewLine
```

Voir aussi

Propriété LastError (Page 432)

Propriété QualityCode (Page 518)

Objet TagSet (Liste) (Page 149)

Objet Tag (Page 144)

Propriété Exponent

Description

TRUE, lorsque les chiffres doivent être représentés avec des exposants (p. ex. "1,00e+000").
BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ExportDirectoryChangeable

Répertoire modifiable - ExportDirectoryChangeable

Détermine s'il sera possible de changer le répertoire pour l'exportation de données en Runtime.

Valeur	Explication
TRUE	Le répertoire pour l'exportation de données peut être changé en Runtime.
FALSE	Le répertoire pour l'exportation de données ne peut pas être changé en Runtime.

L'attribut est dynamisable avec le nom **ExportDirectoryChangeable**. Le type de données est BOOLEAN.

Propriété ExportDirectoryname

Répertoire - ExportDirectoryname

Définit le répertoire dans lequel les données Runtime sont exportées.

Le bouton de sélection vous permet de sélectionner ou de créer le répertoire voulu.

L'attribut est dynamisable avec le nom **ExportDirectoryname**. Le type de données est STRING.

Propriété ExportFileExtension

ExportFileExtension

Cet attribut définit l'extension fichier du fichier d'exportation.

Seule l'extension fichier "csv" était prise en charge jusqu'à présent.

L'attribut est dynamisable avec le nom **ExportFileExtension**. Le type de données est STRING.

Propriété ExportFilename

Nom de fichier - ExportFilename

Définit le nom du fichier dans lequel les données Runtime sont exportées.

L'attribut est dynamisable avec le nom **ExportFilename**. Le type de données est STRING.

Propriété ExportFilenameChangeable

Nom de fichier modifiable - ExportFilenameChangeable

Détermine s'il sera possible de changer le nom du fichier d'exportation en Runtime.

Valeur	Explication
TRUE	Le nom du fichier d'exportation peut être changé en Runtime.
FALSE	Le nom du fichier d'exportation ne peut pas être changé en Runtime.

L'attribut est dynamisable avec le nom **ExportFilenameChangeable**. Le type de données est BOOLEAN.

Propriété ExportFormatGuid

ExportFormatGuid

Affectation définie du numéro d'identification et du fournisseur d'exportation

L'attribut est dynamisable avec le nom **ExportFormatGuid**. Le type de données est STRING.

Propriété ExportFormatName

Format - ExportFormatName

Détermine le format de fichier pour l'exportation.

Le format de fichier "csv" est actuellement le seul format disponible pour l'exportation.

L'attribut est dynamisable avec le nom **ExportFormatName**. Le type de données est STRING.

Voir aussi

Comment exporter des données Runtime

Propriété ExportParameters

ExportParameters

Cet attribut définit les paramètres du format sélectionné via la boîte de dialogue des propriétés.

L'attribut est dynamisable avec le nom **ExportParameters**. Le type de données est VARIANT.

Propriété ExportSelection

Volume de l'exportation - ExportSelection

Définit quelles données Runtime du contrôle sont exportées.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Tout	Toutes les données Runtime du contrôle sont exportées.
1	Sélection	Les données Runtime du contrôle sélectionnés sont exportées.

L'attribut est dynamisable avec le nom **ExportSelection**. Le type de données est LONG.

Propriété ExportShowDialog

Afficher dialogue - ExportShowDialog

Détermine si le dialogue pour l'exportation des données en Runtime s'affiche.

Valeur	Explication
TRUE	Le dialogue est affiché en Runtime.
FALSE	Le dialogue n'est pas affiché en Runtime.

L'attribut est dynamisable avec le nom **ExportShowDialog**. Le type de données est BOOLEAN.

Propriété ExportXML

ExportXML

Cet attribut n'est utilisé qu'en interne.

L'attribut est dynamisable avec le nom **ExportXML**.

Propriété ExtendedOperation

Description

TRUE, lorsque le défileur est positionné sur la valeur finale correspondante (valeur minimale/valeur maximale). Cela s'effectue par clic de souris dans la zone hors du réglage actuel du défileur. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Slider (Page 219)

Objet ScreenItem (Page 132)

Propriété ExtendedZoomingEnable

Description

Active / désactive la propriété ExtendedZooming d'une vue.

ExtendedZooming permet d'agrandir ou de réduire l'affichage d'une vue de process en Runtime en utilisant la roulette de la souris.

BOOLEAN Accès en écriture et en lecture.

Exemple

Active ExtendedZooming pour la vue NewPDL1.

```
'VBS155
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
objScreen.ExtendedZoomingEnable = 1
```

Voir aussi

Objet Screen (Page 138)

1.14.4.7 f

Fe - Fl

Propriété FeatureFullscreen

FeatureFullscreen

Définit si la fonction "Plein écran" est disponible dans le contrôle.

L'attribut est dynamisable avec le nom **FeatureFullscreen**. Le type de données est BOOLEAN.

Propriété FeaturePause

FeaturePause

Définit si la fonction "Pause" est disponible dans le contrôle.

L'attribut est dynamisable avec le nom **FeaturePause**. Le type de données est BOOLEAN.

Propriété FeaturePlay

FeaturePlay

Définit si la fonction "Lecture" est disponible dans le contrôle.

L'attribut est dynamisable avec le nom **FeaturePlay**. Le type de données est BOOLEAN.

Propriété FeatureStepBackward

FeatureStepBackward

Définit si la fonction "Saut arrière" est disponible dans le contrôle.

L'attribut est dynamisable avec le nom **FeatureStepBackward**. Le type de données est BOOLEAN.

Propriété FeatureStepForward

FeatureStepForward

Définit si la fonction "Saut avant" est disponible dans le contrôle.

L'attribut est dynamisable avec le nom **FeatureStepForward**. Le type de données est BOOLEAN.

Propriété FeatureStop

FeatureStop

Définit si la fonction "Arrêt" est disponible dans le contrôle.

L'attribut est dynamisable avec le nom **FeatureStop**. Le type de données est BOOLEAN.

Propriété FeatureVolume

FeatureVolume

Définit si la fonction "Volume" est disponible dans le contrôle.

L'attribut est dynamisable avec le nom **FeatureVolume**. Le type de données est BOOLEAN.

Propriété FileName

FileName

Indique le fichier dont vous voulez afficher ou exécuter le contenu.

L'attribut est dynamisable avec le nom **FileName**. Le type de données est STRING.

Propriété FillColor

Description

Détermine la couleur du motif de remplissage pour l'objet ou la restitue.

LONG (Accès en écriture et en lecture)

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Spécifiez la valeur décimale correspondante pour chacune des trois valeurs RVB.

Exemple :

RVB(200, 150, 100)

Exemple

L'exemple suivant positionne sur Bleu la couleur de remplissage de la vue "ScreenWindow1":

```
'VBS73
Dim objScreen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
objScreen.FillStyle = 131075
```

```
objScreen.FillColor = RGB(0, 0, 255)
```

Voir aussi

Propriété FillStyle (Page 394)
Propriété BackColor (Page 313)
Objet ScreenItem (Page 132)

Propriété Filling

Description

TRUE, lorsque l'objet à bordure fermée peut être rempli (représentant par exemple le niveau de remplissage d'un réservoir). BOOLEAN Accès en écriture et en lecture.
Définissez le niveau de remplissage de l'objet avec la propriété "FillingIndex".

Voir aussi

Objet ScreenItem (Page 132)

Propriétés FillingDirection

L'attribut "Sens de remplissage" détermine dans quel sens l'objet possédant un cadre fermé doit être rempli.

De bas en haut	L'objet est rempli de bas en haut.
De haut en bas	L'objet est rempli de haut en bas.
De gauche à droite	L'objet est rempli de gauche à droite.
De droite à gauche	L'objet est rempli de droite à gauche.

L'attribut est dynamisable avec le nom FillingDirection. Le type de données est LONG.

Propriété FillingIndex

Description

Détermine la valeur en % (par rapport à la hauteur de l'objet) à laquelle l'objet à bordure fermée est rempli.
Le niveau de remplissage est représenté par la couleur d'arrière-plan actuelle. L'arrière-plan non rempli est transparent

Voir aussi

Objet ScreenItem (Page 132)

Propriété FillStyle

Description

Détermine le motif de remplissage pour l'objet ou le restitue.

LONG (Accès en écriture et en lecture)

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
< Transparent >	65536				
< Massiv >	0				
	1048576		196611		196627
	1048577		196612		196628
	1048578		196613		196629
	1048579		196614		196630
	1048832		196615		196631
	1048833		196616		196632
	1048834		196617		196633
	1048835		196618		196634
	131072		196619		196635
	131073		196620		196636
	131074		196621		196637
	131075		196622		196638
	131076		196623		196639
	196608		196624		196640

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
	196609		196625		196641
	196610		196626		196642

Exemple

L'exemple suivant positionne le motif de remplissage de la vue "Fenêtre de vue 1" sur transparent:

```
'VBS190
Dim obj
Set obj = ScreenItems("Rectangle1")
obj.FillStyle = 65536
```

Voir aussi

Propriété FillColor (Page 392)

Propriété BackColor (Page 313)

Objet Screen (Page 138)

Propriété FillStyle2

Description

Détermine le motif de remplissage du bargraphe ou la restitue.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété FillStyleAlignment

Description

Détermine l'orientation du motif de remplissage pour la vue de process.

Normal	Le motif de remplissage se réfère à la vue de process. Lors de l'étirement de la vue en runtime, aucune mise à l'échelle n'est effectuée.
Étirer (fenêtre)	Le motif de remplissage se réfère à la fenêtre dans le Graphics Designer. Lors de l'étirement de la vue en runtime, la mise à l'échelle est effectuée.

Propriété FilterSQL

FilterSQL

Cet attribut définit une sélection des données de l'archive utilisateur pour une instruction SQL.

L'attribut est dynamisable avec le nom **FilterSQL**. Le type de données est STRING.

Propriété FineGrid

Description

TRUE, lorsque l'axe des valeurs a des graduations courtes. La distance entre deux graduations courtes peut être modifiée via la propriété "FineGridValue". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété FineGridValue

Description

Détermine la distance entre deux graduations courtes. L'indication sera ou non exploitée selon la valeur de la propriété "FineGrid".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété FineGridValueX

Description

Détermine la distance entre deux graduations courtes sur l'échelle de l'axe X. L'indication sera ou non exploitée selon la valeur de la propriété "FineGridX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété FineGridValueY

Description

Détermine la distance entre deux graduations courtes sur l'échelle de l'axe Y. L'indication sera ou non exploitée selon la valeur de la propriété "FineGridX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété FineGridX

Description

TRUE, lorsque l'axe X a des graduations courtes. La distance entre deux graduations courtes peut être modifiée via la propriété "FineGridValueX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété FineGridY

Description

TRUE, lorsque l'axe Y a des graduations courtes. La distance entre deux graduations courtes peut être modifiée via la propriété "FineGridValueY".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété FlashBackColor

Description

TRUE, lorsque le clignotement de l'arrière-plan de l'objet est activé. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Propriété FlashBorderColor

Description

TRUE, lorsque le clignotement du trait de l'objet est activé. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Propriété FlashFlashPicture

Description

TRUE, lorsque le clignotement de la vue clignotante est activé. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété FlashForeColor

Description

TRUE, lorsque le clignotement du texte est activé. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Champ d'E/S (Page 193)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Objet ScreenItem (Page 132)

Propriété FlashPicReferenced

Description

TRUE, lorsque la vue clignotante assignée doit être enregistrée. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. Accès en lecture uniquement.

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété FlashPicTransColor

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à la vue clignotante doit être configurée sur "transparent" ou restitue la couleur. LONG Accès en lecture et en écriture.
La couleur n'est configurée sur "transparent" que si la propriété "FlashPicUseTransColor" est positionnée sur la valeur TRUE.

Voir aussi

Objet ScreenItem (Page 132)

Indicateur d'état (Page 208)

Propriété FlashPicture

Description

Restitue la vue clignotante. Accès en lecture uniquement.
La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

La propriété "FlashPicReferenced" détermine dans ce contexte si la vue clignotante doit être enregistrée ou référencée en même temps que l'objet .

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété FlashPicUseTransColor

Description

TRUE, lorsque la couleur configurée (propriété "FlashPicTransColor") pour l'objet Bitmap assigné à la vue clignotante doit être positionnée sur "transparent". BOOLEAN Accès en écriture et en lecture.

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété FlashRate

Description

Détermine la fréquence de clignotement ou la restitue. Plage des valeurs de 0 à 2.

0 = lent

1 = moyen

2 = rapide

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété FlashRateBackColor

Description

Définit la fréquence de clignotement pour l'arrière-plan de l'objet ou la restitue. Plage des valeurs de 0 à 2.

0 = lent

1 = moyen

2 = rapide

Voir aussi

Objet ScreenItem (Page 132)

Propriété FlashRateBorderColor

Description

Définit la fréquence de clignotement du trait de l'objet ou la restitue. Plage des valeurs de 0 à 2.

0 = lent

1 = moyen

2 = rapide

Voir aussi

Objet ScreenItem (Page 132)

Propriété FlashRateFlashPic

Description

Définit la fréquence de clignotement pour la vue clignotante ou la restitue. Plage des valeurs de 0 à 2.

0 = lent

1 = moyen

2 = rapide

Voir aussi

Indicateur d'état (Page 208)

Objet ScreenItem (Page 132)

Propriété FlashRateForeColor

Description

Définit la fréquence de clignotement pour le texte de l'objet ou la restitue. Plage des valeurs de 0 à 2.

0 = lent

1 = moyen

2 = rapide

Voir aussi

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété Flip

Retournement (Flip)

Définit le retournement de l'icône au Runtime.

Les paramétrages suivants sont possibles :

Valeur	Description	Signification
0	Aucune	L'icône ne subit pas de retournement.
1	Horizontal	L'icône est retournée sur l'axe central horizontal.
2	A la verticale	L'icône est retournée sur l'axe central vertical.
3	Les deux	L'icône est retournée sur l'axe central horizontal et l'axe central vertical.

L'attribut est dynamisable avec le nom **Flip**. Le type de données est LONG.

Propriété Flip

Description

Retourne le symbole selon l'axe vertical et/ou l'axe horizontal central du symbole.

- None - 0: Le symbole n'est pas retourné.
- Horizontal - 1: Le symbole est retourné selon l'axe vertical central.
- Vertical - 2: Le symbole est retourné selon l'axe horizontal central.
- Both - 3: Le symbole est retourné aussi bien selon l'axe central vertical qu'horizontal.

Voir aussi

HMI Symbol Library (Page 246)

Objet ScreenItem (Page 132)

Fo - Fr

Propriété FocusColor

Description

Lorsque le Focus se trouve sur le Control en Runtime, le marquage des axes et le texte de position sont identifiés par un cadre. FocusColor détermine la couleur du cadre.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété FocusRect

Description

TRUE, lorsque le bouton de commande est doté en Runtime d'un cadre de sélection dès qu'il a le focus. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété FocusWidth

Description

Lorsque le Focus se trouve sur le Control en Runtime, le marquage des axes et le texte de position sont identifiés par un cadre. FocusWidth détermine la largeur du cadre, plage des valeurs comprise entre 1 et 10 Pixel. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Slider Control (Page 275)
Objet ScreenItem (Page 132)

Propriété Font

Nom - Font

Détermine la police de caractères.

L'attribut n'est pas dynamisable.

Propriété Font (avant WinCC V7)

Description

Détermine la police de caractères ou la restitue. Accès en écriture et en lecture.

L'objet Font a les sous-propriétés

- Size (taille de police)
- Bold (gras oui/non)
- Name (police)
- Italic (italique oui/non)
- Underline (souligné oui/non)
- StrikeThrough (barré oui/non)

Si deux propriétés de l'objet Font sont directement assignées, seule la propriété par défaut "Name" est acceptée.

Exemple

```
'VBS74
Dim objControl1
Dim objControl2
Set objControl1 = ScreenItems("Control1")
Set objControl2 = ScreenItems("Control2")
objControl2.Font = objControl1.Font ' take over only the type of font
```

Voir aussi

WinCC Slider Control (Page 275)

WinCC Push Button Control (Page 268)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

WinCC Digital Analog Clock (Page 251)

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété FontBold

Description

TRUE, lorsque le texte reçoit l'attribut d'écriture "gras" dans l'objet. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Signalisation groupée (Page 203)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété FontItalic

Description

TRUE, lorsque le texte reçoit l'attribut d'écriture "italique" dans l'objet. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Signalisation groupée (Page 203)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété FontName

Description

Détermine la police de caractères du texte dans l'objet ou la restitue.
Vous disposez de toutes les polices installées sous Windows.

Voir aussi

WinCC Push Button Control (Page 268)

Signalisation groupée (Page 203)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété FontPosition

Description

Restitue la police de caractères pour l'affichage de la position du curseur dans la partie inférieure de l'objet. Vous disposez de toutes les polices installées sous Windows. Accès en lecture uniquement.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété FontSize

Description

Détermine en points la taille de police du texte dans l'objet ou la restitue.

Voir aussi

WinCC Push Button Control (Page 268)

Signalisation groupée (Page 203)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété FontStrikeThru

Description

TRUE, lorsque le texte reçoit l'attribut d'écriture "barré" dans l'objet. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété FontUnderline

Description

TRUE, lorsque le texte reçoit l'attribut d'écriture "souligné" dans l'objet. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Signalisation groupée (Page 203)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété ForeColor

Description

Détermine la couleur du texte dans l'objet ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Slider Control (Page 275)

WinCC Push Button Control (Page 268)

WinCC Digital Analog Clock (Page 251)

HMI Symbol Library (Page 246)

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété ForeColor

Couleur de premier plan (ForeColor)

Définit la couleur de premier plan de l'icône au moyen du dialogue "Sélection des couleurs". L'icône est affichée dans la couleur de premier plan en mode de premier plan "Ombré" et "Opaque".

L'attribut est dynamisable avec le nom **ForeColor**. Le type de données est LONG.

Propriété ForeFlashColorOff

Description

Définit la couleur du texte pour l'état clignotant "Arrêt" ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Liste de textes (Page 206)

Texte statique (Page 176)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété ForeFlashColorOn

Description

Définit la couleur du texte pour l'état clignotant "Marche" ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Texte statique (Page 176)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété FrameColor

Description

Détermine la couleur de la zone rectangulaire ou carrée sur laquelle se trouve le disque graduée, ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété FrameColorDown

Description

Détermine la couleur pour la partie inférieure droite du cadre tridimensionnel du bouton (bouton appuyé) ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété FrameColorUp

Description

Détermine la couleur pour la partie supérieure gauche du cadre tridimensionnel du bouton (bouton non appuyé) ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété FramePicture

Description

Restitue le nom d'image de la vue d'arrière-plan pour le disque gradué. Accès en lecture uniquement.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété FrameScale

Description

Détermine le diamètre du disque gradué par rapport à la plus petite valeur des attributs géométriques Largeur et Hauteur, ou le restitue. Accès en écriture et en lecture.

La plage des valeurs est (intervalle de graduation - largeur de graduation) jusqu'à 1.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété FrameWidth

Description

Détermine en Pixel la largeur de bordure du bouton de commande ou la restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété FreezeProviderConnections

Description

Permet de modifier les propriétés pour l'intégration des données ("ProviderType", "Source"...), sans que la modification ne soit immédiatement effective. Lorsque "SourceTagNameX" est par exemple modifié, il se peut que des combinaisons non admises se créent avec "SourceTagNameY".

C'est pourquoi vous devez positionner "FreezeProviderConnections" sur TRUE avant la modification d'un attribut pour l'intégration des données. Après la modification de toutes les propriétés pour l'intégration des données, "FreezeProviderConnection" est positionné sur FALSE, et les modifications deviennent actives.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

1.14.4.8 G

Propriété GlobalColorScheme

Description

Définit si les couleurs définies dans le schéma de couleurs global du design actuel sont utilisées pour cet objet.

TRUE, quand l'objet est représenté avec les couleurs issues du schéma de couleurs global défini pour ce type d'objet.

FALSE, quand l'objet est représenté avec les couleurs correspondant aux paramétrages dans l'objet.

BOOLEAN Accès en écriture et en lecture.

Propriété GlobalShadow

Description

Définit si l'objet est représenté avec l'ombrage déterminé dans le design actif.

TRUE, quand l'objet est représenté avec l'ombrage global défini pour ce type d'objet.

FALSE, quand aucune ombre n'est représentée.

BOOLEAN Accès en écriture et en lecture.

Propriété GraphDirection (avant WinCC V7)

Description

Détermine sur quel bord de la fenêtre de courbes les valeurs actuelles doivent s'afficher. Accès en écriture et en lecture.

0: Les valeurs positives sont affichées de haut en bas vers la droite.

-1: Les valeurs positives sont affichées de gauche à droite et vers le haut.

-2: Les valeurs positives sont affichées de haut en bas vers la droite.

-3: Les valeurs positives sont affichées de bas en haut vers la gauche.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété GraphDirection

Sens de tracé de l'enregistreur - GraphDirection

Détermine le sens selon lequel les valeurs sur l'axe sont mises à jour.

Valeur	Description	Explication
0	à partir de la droite	Les valeurs mises à jour sont représentées à droite dans le contrôle.
1	à partir de la gauche	Les valeurs mises à jour sont représentées à gauche dans le contrôle.
2	à partir du haut	Les valeurs mises à jour sont représentées en haut dans le contrôle.
3	à partir du bas	Les valeurs mises à jour sont représentées en bas dans le contrôle.

Si vous choisissez "à partir du haut" ou "à partir du bas" pour le sens de tracé de l'enregistreur, vous devez utiliser les polices True Type dans la fenêtre de courbes. C'est la seule façon de garantir que le texte de l'axe vertical est bien lisible.

L'attribut est dynamisable avec le nom **GraphDirection**. Le type de données est LONG.

Propriété GridLineColor

Couleur de la ligne de séparation / Contenu - GridLineColor

Indique la couleur des lignes de séparation dans le contenu du tableau. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **GridLineColor**. Le type de données est LONG.

Propriété GridLineHorz

Description

TRUE, lorsque les colonnes de la fenêtre d'alarmes sont séparées par des lignes horizontales. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété GridLines

Description

TRUE, lorsque la fenêtre de courbes est représentée avec des grilles parallèles à l'axe x. La distance entre deux lignes de la grille peut être modifiée via la propriété "GridLineValue". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété GridlinesValueX

Description

Détermine la distance entre deux lignes de la grille de l'axe X ou la restitue. L'indication sera ou non exploitée selon la valeur de la propriété "GridLinesX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété GridlinesValueY

Description

Détermine la distance entre deux lignes de la grille de l'axe Y ou la restitue. L'indication sera ou non exploitée selon la valeur de la propriété "GridLinesY".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété GridlinesX

Description

TRUE, lorsque la fenêtre de courbes est représentée avec des grilles parallèles à l'axe X. La distance entre deux lignes de la grille peut être modifiée via la propriété "GridLinesValueX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété GridlinesY

Description

TRUE, lorsque la fenêtre de courbes est représentée avec des grilles parallèles à l'axe Y. La distance entre deux lignes de la grille peut être modifiée via la propriété "GridLinesValueX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété GridLineValue

Description

Détermine la distance entre deux lignes de la grille. L'indication sera ou non exploitée selon la valeur de la propriété "GridLines".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété GridLineVert

Description

TRUE, lorsque les colonnes de la fenêtre d'alarmes sont séparées par des lignes verticales. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété GridLineWidth

Largeur des lignes de séparation - GridLineWidth

Détermine la largeur des lignes de séparation en pixels.

L'attribut est dynamisable avec le nom **GridLineWidth**. Le type de données est LONG.

1.14.4.9 h

Ha - Hi

Propriété HandFillColor

Description

Détermine la couleur de remplissage de toutes les aiguilles sur l'horloge analogique ou la restitue. Pour que les aiguilles soient représentées par la couleur de remplissage prévue, il vous faut paramétrer la propriété "Handtype" sur "0" (opaque). LONG Accès en écriture et en lecture.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété Handtype

Description

Détermine la représentation des aiguilles:

- 0: Lorsqu'elles sont remplies, les aiguilles sont dans la couleur de remplissage prévue à cet effet et lorsqu'elle ont un bord elles sont dans la couleur d'avant-plan.
- 1: Les aiguilles sont transparentes et représentées par un bord dans la couleur d'avant-plan.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété HeaderSort

Description

Détermine s'il est possible de trier les blocs d'alarmes par leurs titres de colonne.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété Height

Description

Détermine la hauteur de l'objet en Pixel ou la restitue.

LONG (Accès en écriture et en lecture)

Exemple

Dans l'exemple suivant la hauteur de tous les objets de la vue "NewPDL1" dont le nom commence par "Circle" est divisée par deux:

```
'VBS75
Dim objScreen
Dim objCircle
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
'
'Searching all circles
strName = objScreen.ScreenItems.Item(lngIndex).ObjectName
If "Circle" = Left(strName, 6) Then
'
'to halve the height of the circles
Set objCircle = objScreen.ScreenItems(strName)
objCircle.Height = objCircle.Height / 2
    End If
Next
```

Voir aussi

Propriété Width (Page 669)

Type d'objet de l'objet ScreenItem (Page 150)

Objet ScreenItem (Page 132)

Propriété HiddenInput

Description

TRUE, lorsque la valeur d'entrée n'est pas affichée pendant la saisie. Un * s'affiche pour chaque caractère. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

HideTagNames-Propriété

Description

TRUE quand le nom d'archive et de variable ne s'affiche pas dans la courbe via la touche droite de la souris, dans la barre d'état et dans le tableau représentant les coordonnées. BOOLEAN Accès en écriture et en lecture.

Propriété HitlistColumnAdd

HitlistColumnAdd

Accepte dans la liste des blocs d'alarme sélectionnés le bloc d'alarme sélectionné dans la liste des blocs d'alarme existants.

L'attribut est dynamisable avec le nom **HitlistColumnAdd** . Le type de données est STRING.

Propriété HitlistColumnCount

HitlistColumnCount

Cet attribut indique le nombre des blocs d'alarme affichés dans la liste d'occurrences en Runtime.

L'attribut est dynamisable avec le nom **HitlistColumnCount** . Le type de données est LONG.

Propriété HitlistColumnIndex

HitlistColumnIndex

Cet attribut fait référence à un bloc d'alarme sélectionné pour la liste d'occurrences. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un bloc d'alarme précis de la liste d'occurrences.

Les valeurs admissibles pour "HitlistColumnIndex" et "HitlistColumnCount" se situent entre 0 et -1. L'attribut "HitlistColumnCount" indique le nombre des blocs d'alarme sélectionnés pour la liste d'occurrences.

L'attribut "HitlistColumnIndex" est dynamisable via l'attribut **HitlistColumnRepos**. Le type de données est LONG.

Propriété HitlistColumnName

HitlistColumnName

Cet attribut affiche le nom du bloc d'alarme de la liste d'occurrences qui est référencé à l'aide de l'attribut "HitlistColumnIndex". Vous ne pouvez pas modifier le nom.

L'attribut est dynamisable avec le nom **HitlistColumnName** . Le type de données est STRING.

Propriété HitlistColumnRemove

HitlistColumnRemove

Supprime le bloc d'alarme marqué dans la liste des blocs d'alarme sélectionnés et l'insère dans la liste des blocs d'alarme existants.

L'attribut est dynamisable avec le nom **HitlistColumnRemove**. Le type de données est STRING.

HitlistColumnRepos

Haut/bas - MMessageColumnRepos/HitlistColumnRepos

Modifie l'ordre des blocs d'alarme. "Haut" et "Bas" déplacent le bloc d'alarme sélectionné dans la liste vers le haut ou le bas. Ce qui a pour conséquence que le bloc d'alarme est positionné plus en avant ou plus en arrière dans le contrôle en Runtime.

L'attribut pour la liste d'occurrences est dynamisable avec le nom **HitlistColumnRepos** .

L'attribut pour la liste d'alarmes est dynamisable avec le nom **MessageColumnRepos**.

Le type de données est LONG.

Propriété HitlistColumnSort

HitlistColumnSort

Cet attribut définit comment le bloc d'alarme référencé dans "HitlistColumnIndex" est trié pour la liste d'occurrences.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	aucun	Aucun tri
1	croissant	Tri croissant de la plus petite à la plus grande valeur
2	décroissant	Tri décroissant de la plus grande à la plus petite valeur

L'attribut est dynamisable avec le nom **HitlistColumnSort** . Le type de données est LONG.

Propriété HitlistColumnSortIndex

HitlistColumnSortIndex

Cet attribut indique l'ordre de tri du bloc d'alarme référencé dans "HitlistColumnIndex" pour la liste d'occurrences. Si vous activez la valeur sur "0", le critère de tri est supprimé dans "HitlistColumnSort".

L'attribut est dynamisable avec le nom **HitlistColumnSortIndex**. Le type de données est LONG.

HitlistColumnVisible

Blocs d'alarme sélectionnés - MessageColumnVisible/HitlistColumnVisible

Blocs d'alarme sélectionnés de la liste des alarmes ou de la liste d'occurrences à afficher en runtime :. Cet attribut définit comment s'affiche le bloc d'alarme référencé dans "MessageColumnIndex" ou "HitlistColumnIndex".

L'attribut pour la liste d'alarmes est dynamisable avec le nom **MessageColumnVisible**.

L'attribut pour la liste d'occurrences est dynamisable avec le nom **HitlistColumnVisible**.

Le type de données est BOOLEAN.

Propriété HitlistDefaultSort

HitlistDefaultSort

Cet attribut définit le tri standard dans les colonnes du tableau de la liste d'occurrences.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	croissant	La liste est triée par ordre croissant selon la fréquence.
1	décroissant	La liste est triée par ordre décroissant selon la fréquence.

L'attribut est dynamisable avec le nom **HitlistDefaultSort**. Le type de données est LONG.

Propriété HitListMaxSourceItems

Nombre maximal Enregistrements - HitListMaxSourceItems

Détermine le nombre maximal d'enregistrements pour les statistiques.

L'attribut est dynamisable avec le nom **HitListMaxSourceItems** . Le type de données est LONG.

Propriété HitListMaxSourceItemsWarn

Avertissement - HitListMaxSourceItemsWarn

Détermine si un avertissement s'affiche si le maximum d'enregistrements admis est atteint.

Valeur	Explication
TRUE	L'avertissement s'affiche si le maximum d'enregistrements admis est atteint.
FALSE	Aucun avertissement ne s'affiche si le maximum d'enregistrements admis est atteint.

L'attribut est dynamisable avec le nom **HitListMaxSourceItemsWarn**. Le type de données est BOOLEAN.

Propriété HitListRelTime

Plage de temps pour les statistiques - HitListRelTime

Détermine si une plage de temps est utilisée pour les statistiques.

Valeur	Explication
TRUE	Si aucune plage de temps n'est spécifiée dans la sélection, la plage de temps spécifiée est utilisée pour les statistiques.
FALSE	La plage de temps spécifiée n'est pas utilisée.

L'attribut est dynamisable avec le nom **HitListRelTime**. Le type de données est BOOLEAN.

Propriété HitListRelTimeFactor

Plaage de temps - HitListRelTimeFactor

Détermine le facteur pour définir la plage de temps. Seuls les facteurs entiers sont admis.

L'attribut est dynamisable avec le nom **HitListRelTimeFactor**. Le type de données est LONG.

Propriété HitListRelTimeFactorType

Plage de temps - HitListRelTimeFactorType

Détermine l'unité de temps pour définir la plage de temps.

Les unités de temps suivantes sont disponibles :

Valeur	Description
0	Minute
1	Heure
2	Tag
3	Semaine
4	Mois

L'attribut est dynamisable avec le nom **HitListMaxRelTimeFactorType**. Le type de données est LONG.

Ho - Hy

Propriété HorizontalGridLines

Horizontale - HorizontalGridLines

Détermine si les lignes de séparation horizontales seront affichées.

Valeur	Explication
TRUE	Les lignes de séparation horizontales seront affichées.
FALSE	Les lignes de séparation horizontales ne seront pas affichées.

L'attribut est dynamisable avec le nom **HorizontalGridLines**. Le type de données est BOOLEAN.

Propriété Hotkey

Description

Pour l'objet Button, restitue la touche de fonction pour la commande de la souris.

Accès en lecture uniquement.

Voir aussi

Bouton (Page 210)

Objet ScreenItem (Page 132)

Propriété HourNeedleHeight

Description

Détermine la longueur de l'aiguille des heures sur l'horloge analogique ou la restitue. La longueur est spécifiée en pour cent par rapport à la demi-longueur du côté le plus court du rectangle d'arrière-plan. Accès en écriture et en lecture.

Exemple:
Si le côté le plus court du rectangle d'arrière-plan était long de 100 pixels.
La longueur de l'aiguilles des heures serait 50.
D'où une aiguille des heures longue de (100 pixels / 2) * 0,5 = 25 pixels.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété HourNeedleWidth

Description

Détermine la largeur de l'aiguille des heures sur l'horloge analogique ou la restitue. La largeur est spécifiée en pour cent par rapport à la longueur de l'aiguille des heures multipliée par deux. Accès en écriture et en lecture.

Exemple:
La longueur de l'aiguille des heures est 25 pixels.
La largeur de l'aiguille des heures est 10.
D'où une aiguille des heures large de 25 pixels * 2 * 0,1 = 5 pixels.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété Hysteresis

Description

TRUE, lorsque l'affichage doit se faire avec hystérésis. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété HysteresisRange

Description

Détermine l'hystérésis en % de la valeur affichée ou la restitue.
La propriété Hysteresis doit avoir la valeur TRUE pour pouvoir calculer l'hystérésis.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

1.14.4.10 I

Propriété IconSpace

IconSpace

Cet attribut définit l'intervalle entre le symbole et le texte dans les cellules du tableau. La valeur s'applique si le symbole et le texte sont affichés.

L'attribut est dynamisable avec le nom **IconSpace**. Le type de données est LONG.

Propriété IndependentWindow

Description

Définit si la représentation de la fenêtre de vue en runtime dépend de la vue de process dans laquelle la fenêtre de vue a été configurée.

TRUE si la taille et la position de la fenêtre de vue sont indépendantes de la vue de process et uniquement définies par l'attribut "Mode fenêtre".

FALSE si la taille et la position de la fenêtre de vue varient avec le déplacement ou la mise à l'échelle de la vue de process.

Propriété Index

Description

Case à cocher, Case d'option

Définit les numéros (0 à 31) du champ dont vous voulez définir le texte, ou les retourne.

Champ de combinaison, champ de liste

Définit les numéros (0 à 31) de la ligne dont vous voulez définir le texte, ou les retourne.

Polygone, trait polygonal, polygone tuyau

Définit ou retourne le numéro du point d'angle dont vous voulez modifier ou faire afficher les coordonnées de position.

WinCC Online Trend Control, WinCC Online Table Control, WinCC Function Trend Control

La propriété "Index" est exploitée par d'autres propriétés pour pouvoir assigner leurs paramètres à une courbe ou un couple de colonnes particuliers . Les valeurs valables pour l'index sont comprises dans une plage entre 0 et (NumItems - 1). La propriété "NumItems" contient le nombre de courbes/couples de colonnes à afficher. L'indice doit toujours être activé avant de modifier les propriétés d'une courbe/colonne au Runtime.

Indicateur d'état

Détermine l'état (0 à 255) ou le restitue. Vous pouvez spécifier une vue de base et une vue clignotante pour chaque valeur d'état.

Voir aussi

Indicateur d'état (Page 208)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Trait polygonal (Page 168)

Polygone (Page 166)

Case d'option (Page 215)

Case à cocher (Page 213)

Objet ScreenItem (Page 132)

Propriété InnerBevelOffset

Description

Détermine la distance entre la bordure intérieure et la bordure extérieure.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété InnerBevelStyle

Description

Détermine l'effet tridimensionnel pour la bordure intérieure de l'objet.

- 0: Aucune bordure.
- 1: La bordure est représentée en creux.
- 2: La bordure est représentée en relief.
- 3: La bordure est représentée en une couleur, sans effet tridimensionnel. La couleur de la bordure est définie par la propriété "BevelColorDown" .

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété InnerBevelWidth

Description

Détermine la largeur de la bordure intérieure en Pixel.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété InputValue

Description

Définit la valeur indiquée par l'utilisateur dans le champ d'E/S. La valeur n'est pas affichée dans le champ d'E/S lors de la définition de la propriété.

Si vous voulez que la valeur saisie soit affichée dans le champ d'E/S après confirmation via la pression de la touche <Entrée>, configurez une liaison directe entre les propriétés "Valeur d'entrée" et "Valeur de sortie". La liaison directe n'est intéressante que si aucune liaison de variable n'a été configurée pour la valeur de sortie, mais que l'utilisateur désire néanmoins interroger la valeur saisie, par ex. via un script.

LONG Accès en écriture et en lecture.

Voir aussi

Exemple : Pour appeler les méthodes d'un ActiveX-Control (Page 807)

Propriété InsertData

Description

Insère des données pour la courbe actuelle.

TRUE: "DataIndex" est ignoré et les données sont ajoutées à la fin dans le tampon de données.

FALSE: Les données sont insérées à la position "DataIndex" du tampon de données.

La fenêtre des courbes est redessinée à chaque opération effectuée avec "Insert Data".

Remarque

La propriété est prise en charge uniquement pour les Controls avant WinCC V7.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété Instance

Description

Retourne une instance de l'objet d'alarme.

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété ItemBorderBackColor

Description

Détermine la couleur d'arrière-plan des traits de séparation dans la liste de sélection de l'objet "Liste de textes" . LONG Accès en écriture et en lecture. La couleur d'arrière-plan n'est visible que si la propriété ItemBorderStyle > 0.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété ItemBorderColor

Description

Détermine la couleur des traits de séparation dans la liste de sélection de l'objet "Liste de textes" ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété ItemBorderStyle

Description

Détermine le style des lignes de séparation dans la liste de sélection de l'objet "Liste des textes" ou la restitue. Plage des valeurs de 0 à 4.

0 = Trait plein

1 = Trait discontinu

2 = Trait en pointillés

3 = Trait mixte

4 = Trait point point

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété ItemBorderWidth

Description

Détermine en Pixel l'épaisseur des traits de séparation dans la liste de sélection de l'objet "Liste de textes" ou la restitue.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

ItemProviderClsid-Propriété

Description

"ItemProviderClsid" montre si la courbe référence dans Contrôle Courbes via "Index"est reliée à une variable d'archive ou à une variable en ligne.

Attention : Si vous affectez une valeur à la propriétés "ProviderClsid", vous écrasez la propriété "ItemProviderClsid" spécifique à la courbe.

- {416A09D2-8B5A-11D2-8B81-006097A45D48}: La courbe est associée à une variable d'archive.
- {A3F69593-8AB0-11D2-A440-00A0C9DBB64E}: La courbe est associée à une variable den ligne.

Si les courbes sont associées avec des variables d'archive et en ligne, la propriété "ProviderClsid" renvoie la valeur "{00000000-0000-0000-0000-000000000000}".

Propriété ItemVisible

Description

TRUE, lorsqu'une courbe référencée via la propriété "Index" ou un couple de colonnes référencé via "Index" sont visibles. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

1.14.4.11 L

Lab - Las

Propriété Label

Description

La propriété "Index" référence une courbe. Selon la valeur de la propriété "TimeAxis", la désignation de l'axe des temps ou de l'axe des valeurs sera déterminée via Label.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété LabelColor

Description

Détermine la couleur du marquage de la graduation.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété LabelX

Description

Détermine, selon la valeur de "TimeAxisX", la désignation de l'axe X par une courbe référencée via "Index" ou la restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété LabelY

Description

Détermine, selon la valeur de "TimeAxisY", la désignation de l'axe Y par une courbe référencée via "Index" ou la restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété LanguageSwitch

Description

Restitue la valeur qui définit où les textes d'assignation dépendant de la langue doivent être enregistrés. Accès en lecture uniquement.

TRUE, lorsque les textes sont gérés dans la bibliothèque de textes. Les traductions en d'autres langues s'effectuent dans la bibliothèque de textes.

FALSE, lorsque les textes sont directement gérés dans l'objet. Les traductions en d'autres langues peuvent se faire à l'aide de Text Distributor.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété Language

Description

Attribue la langue Runtime actuelle ou la lit.

Spécifiez la langue Runtime en VBS avec un indicatif de pays, p, ex. 1031 pour l'allemand - standard, 1033 pour l'anglais - USA, etc. Vous trouverez dans les principes de base de VBScript sous la rubrique "Diagramme du schéma région-ID (LCID)" un récapitulatif de tous les indicatifs de pays.

INTEGER (Accès en écriture et en lecture)

Exemple

L'exemple suivant attribue la langue allemande :

```
'VBS76
HMIRuntime.Language = 1031
```

Voir aussi

Objet HMIRuntime (Page 125)

Propriété LastError

Description

Restitue un code d'erreur signalant la réussite de la dernière opération, p. ex. des informations concernant la procédure d'écriture ou de lecture d'une variable. La propriété "QualityCode" permet de donner une appréciation sur la qualité de la valeur fournie. La propriété "ErrorDescription" permet de décrire l'erreur.

LONG (readonly)

Les codes d'erreur suivants sont définis :

Code écrit sous forme hexadécimale	Description
0x00000000	OK
0x80040001	Erreur d'exécution
0x80040002	Erreur de variable
0x80040003	Serveur non disponible
0x80040004	Multi Tag Error (Erreur pour une ou plusieurs variables)

Pour que LastError restitue une valeur, il faut tout d'abord exécuter un Read.

Le message "Multi Tag Error" est émis si une erreur se produit via l'objet TagSet en lisant ou en écrivant plusieurs variables. Il vous faut analyser la propriété LastError de chacune des variables pour déterminer la variable concernée par l'erreur ainsi que le type d'erreur.

Exemple

L'exemple suivant émet le code d'erreur pour la variable "Tag1" :

```
'VBS77
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
MsgBox objTag.LastError
```

Dans l'exemple suivant deux variables sont insérées dans la liste TagSet et la propriété LastError s'affiche en tant que Trace.

```
'VBS178
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
HMIRuntime.Trace "LastError: " & group.LastError & vbNewLine
```

Vous accédez comme suit à la propriété LastError d'une variable contenue dans la liste :

```
HMIRuntime.Trace "LastError: " & group("Motor1").LastError & vbNewLine
```

Voir aussi

Objet TagSet (Liste) (Page 149)

Propriété QualityCode (Page 518)

Propriété ErrorDescription (Page 386)

Objet Tag (Page 144)

Layer

Propriété Layer

Description

Restitue le niveau (Layer) dans la vue où se trouve un objet. 32 niveaux sont au total disponibles, le niveau "0" étant le plus bas et le niveau "31" le plus élevé.

Dans un niveau, ce sont les objets configurés en premier qui se trouvent au dernier plan.

LONG (readonly)

Remarque

La propriété Layer affiche le niveau où se trouve l'objet. Le niveau "0" s'affiche Niveau"0".

En cas d'accès, les niveaux sont comptés dans VBS à partir de 1. C'est pourquoi on accède au niveau "1" avec layers(2).

Exemple

L'exemple suivant affiche le nom et le niveau pour tous les objets de la vue "NewPDL1":

```
'VBS78
Dim objScreen
Dim objScrItem
Dim lngAnswer
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems.Item(lngIndex).ObjectName
Set objScrItem = objScreen.ScreenItems(strName)
lngAnswer = MsgBox(strName & " is in layer  " & objScrItem.Layer,vbOKCancel)
If vbCancel = lngAnswer Then Exit For
Next
```

Voir aussi

Objet ScreenItem (Page 132)

Propriété Layer00Checked

Description

TRUE, lorsque le seuil 0 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer00Value et Layer00Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer01Checked

Description

TRUE, lorsque le seuil 1 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer01Value et Layer01Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer02Checked

Description

TRUE, lorsque le seuil 2 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer02Value et Layer02Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer03Checked

Description

TRUE, lorsque le seuil 3 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer03Value et Layer03Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer04Checked

Description

TRUE, lorsque le seuil 4 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer04Value et Layer04Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer05Checked

Description

TRUE, lorsque le seuil 5 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer05Value et Layer05Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer06Checked

Description

TRUE, lorsque le seuil 6 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer06Value et Layer06Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer07Checked

Description

TRUE, lorsque le seuil 7 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer07Value et Layer07Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer08Checked

Description

TRUE, lorsque le seuil 8 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer08Value et Layer08Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer09Checked

Description

TRUE, lorsque le seuil 9 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer09Value et Layer09Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer10Checked

Description

TRUE, lorsque le seuil 10 doit être surveillé. BOOLEAN Accès en lecture et en écriture.
Le seuil et la représentation sont déterminés par les propriétés Layer10Value et Layer10Color.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer00Color

Description

Détermine la couleur du seuil 0 ou la restitue. LONG Accès en lecture et en écriture.
Lorsque la surveillance du seuil est activée (Propriété Layer00Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer01Color

Description

Détermine la couleur du seuil 1 ou la restitue. LONG Accès en lecture et en écriture.
Lorsque la surveillance du seuil est activée (Propriété Layer01Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer02Color

Description

Détermine la couleur du seuil 2 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer02Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer03Color

Description

Détermine la couleur du seuil 3 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer03Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer04Color

Description

Détermine la couleur du seuil 4 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer04Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer05Color

Description

Détermine la couleur du seuil 5 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer05Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer06Color

Description

Détermine la couleur du seuil 6 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer06Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer07Color

Description

Détermine la couleur du seuil 7 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer07Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer08Color

Description

Détermine la couleur du seuil 8 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer08Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer09Color

Description

Détermine la couleur du seuil 9 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer09Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer10Color

Description

Détermine la couleur du seuil 10 ou la restitue. LONG Accès en lecture et en écriture. Lorsque la surveillance du seuil est activée (Propriété Layer10Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer00FillColor

Couleur de remplissage du bargraphe 0 (Layer00FillColor)

L'attribut "Couleur de remplissage du bargraphe 0" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 0".

L'attribut "Couleur de remplissage du bargraphe 0" peut être dynamisé avec le nom "Layer00FillColor".

Propriété Layer01FillColor

Couleur de remplissage du bargraphe 1 (Layer01FillColor)

L'attribut "Couleur de remplissage du bargraphe 1" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 1".

L'attribut "Couleur de remplissage du bargraphe 1" peut être dynamisé avec le nom "Layer01FillColor".

Propriété Layer02FillColor

Couleur de remplissage du bargraphe 2 (Layer02FillColor)

L'attribut "Couleur de remplissage du bargraphe 2" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 2".

L'attribut "Couleur de remplissage du bargraphe 2" peut être dynamisé avec le nom "Layer02FillColor".

Propriété Layer03FillColor

Couleur de remplissage du bargraphe 3 (Layer03FillColor)

L'attribut "Couleur de remplissage du bargraphe 3" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 3".

L'attribut "Couleur de remplissage du bargraphe 3" peut être dynamisé avec le nom "Layer03FillColor".

Propriété Layer04FillColor

Couleur de remplissage du bargraphe 4 (Layer04FillColor)

L'attribut "Couleur de remplissage du bargraphe 4" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 4".

L'attribut "Couleur de remplissage du bargraphe 4" peut être dynamisé avec le nom "Layer04FillColor".

Propriété Layer05FillColor

Couleur de remplissage du bargraphe 5 (Layer05FillColor)

L'attribut "Couleur de remplissage du bargraphe 5" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 5".

L'attribut "Couleur de remplissage du bargraphe 5" peut être dynamisé avec le nom "Layer05FillColor".

Propriété Layer06FillColor

Couleur de remplissage du bargraphe 6 (Layer06FillColor)

L'attribut "Couleur de remplissage du bargraphe 6" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 6".

L'attribut "Couleur de remplissage du bargraphe 6" peut être dynamisé avec le nom "Layer06FillColor".

Propriété Layer07FillColor

Couleur du motif de remplissage du bargraphe 7 (Layer07FillColor)

L'attribut "Couleur de remplissage du bargraphe 7" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 7".

L'attribut "Couleur de remplissage du bargraphe 7" peut être dynamisé avec le nom "Layer07FillColor".

Propriété Layer08FillColor

Couleur du motif de remplissage du bargraphe 8 (Layer08FillColor)

L'attribut "Couleur de remplissage du bargraphe 8" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 8".

L'attribut "Couleur de remplissage du bargraphe 8" peut être dynamisé avec le nom "Layer08FillColor".

Propriété Layer09FillColor

Couleur du motif de remplissage du bargraphe 9 (Layer09FillColor)

L'attribut "Couleur de remplissage du bargraphe 9" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 9".

L'attribut "Couleur de remplissage du bargraphe 9" peut être dynamisé avec le nom "Layer09FillColor".

Propriété Layer10FillColor

Couleur du motif de remplissage du bargraphe 10 (Layer10FillColor)

L'attribut "Couleur de remplissage du bargraphe 10" définit la couleur avec laquelle le bargraphe est rempli par rapport au "Seuil 10".

L'attribut "Couleur de remplissage du bargraphe 10" peut être dynamisé avec le nom "Layer10FillColor".

Propriété Layer00FillStyle

Motif de remplissage du bargraphe 0 (Layer00FillStyle)

L'attribut "Motif de remplissage du bargraphe 0" définit le motif du bargraphe par rapport au "Seuil 0". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 0" doit être différente de la "Couleur du bargraphe 0".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 0" peut être dynamisé avec le nom "Layer00FillStyle".

Propriété Layer01FillStyle

Motif de remplissage du bargraphe 1 (Layer01FillStyle)

L'attribut "Motif de remplissage du bargraphe 1" définit le motif du bargraphe par rapport au "Seuil 1". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 1" doit être différente de la "Couleur du bargraphe 1".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 1" peut être dynamisé avec le nom "Layer01FillStyle".

Propriété Layer02FillStyle

Motif de remplissage du bargraphe 2 (Layer02FillStyle)

L'attribut "Motif de remplissage du bargraphe 2" définit le motif du bargraphe par rapport au "Seuil 2". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 2" doit être différente de la "Couleur du bargraphe 2".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 2" peut être dynamisé avec le nom "Layer02FillStyle".

Propriété Layer03FillStyle

Motif de remplissage du bargraphe 3 (Layer03FillStyle)

L'attribut "Motif de remplissage du bargraphe 3" définit le motif du bargraphe par rapport au "Seuil 3". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 3" doit être différente de la "Couleur du bargraphe 3".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 3" peut être dynamisé avec le nom "Layer03FillStyle".

Propriété Layer04FillStyle

Motif de remplissage du bargraphe 4 (Layer04FillStyle)

L'attribut "Motif de remplissage du bargraphe 4" définit le motif du bargraphe par rapport au "Seuil 4". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 4" doit être différente de la "Couleur du bargraphe 4".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 4" peut être dynamisé avec le nom "Layer04FillStyle".

Propriété Layer05FillStyle

Motif de remplissage du bargraphe 5 (Layer05FillStyle)

L'attribut "Motif de remplissage du bargraphe 5" définit le motif du bargraphe par rapport au "Seuil 5". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 5" doit être différente de la "Couleur du bargraphe 5".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 5" peut être dynamisé avec le nom "Layer05FillStyle".

Propriété Layer06FillStyle

Motif de remplissage du bargraphe 6 (Layer06FillStyle)

L'attribut "Motif de remplissage du bargraphe 6" définit le motif du bargraphe par rapport au "Seuil 6". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 6" doit être différente de la "Couleur du bargraphe 6".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 6" peut être dynamisé avec le nom "Layer06FillStyle".

Propriété Layer07FillStyle

Motif de remplissage du bargraphe 7 (Layer07FillStyle)

L'attribut "Motif de remplissage du bargraphe 7" définit le motif du bargraphe par rapport au "Seuil 7". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 7" doit être différente de la "Couleur du bargraphe 7".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 7" peut être dynamisé avec le nom "Layer07FillStyle".

Propriété Layer08FillStyle

Motif de remplissage du bargraphe 8 (Layer08FillStyle)

L'attribut "Motif de remplissage du bargraphe 8" définit le motif du bargraphe par rapport au "Seuil 8". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 8" doit être différente de la "Couleur du bargraphe 8".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 8" peut être dynamisé avec le nom "Layer08FillStyle".

Propriété Layer09FillStyle

Motif de remplissage du bargraphe 9 (Layer09FillStyle)

L'attribut "Motif de remplissage du bargraphe 9" définit le motif du bargraphe par rapport au "Seuil 9". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 9" doit être différente de la "Couleur du bargraphe 9".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 9" peut être dynamisé avec le nom "Layer09FillStyle".

Propriété Layer10FillStyle

Motif de remplissage du bargraphe 10 (Layer10FillStyle)

L'attribut "Motif de remplissage du bargraphe 10" définit le motif du bargraphe par rapport au "Seuil 10". Pour que le motif de remplissage soit visible, la "Couleur de remplissage du bargraphe 10" doit être différente de la "Couleur du bargraphe 10".

Il existe 50 motifs de remplissage. Le motif de remplissage 0 "Massif" remplit l'objet avec la couleur d'arrière-plan prédéfinie. Le motif de remplissage 1 "Transparent" a pour effet que ne sont visibles ni arrière-plan ni motif de remplissage.

L'attribut "Motif de remplissage du bargraphe 10" peut être dynamisé avec le nom "Layer10FillStyle".

Propriété Layer00Value

Description

Détermine la valeur pour "Seuil 0" ou la restitue.
La surveillance n'est active que si la propriété Layer00Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer01Value

Description

Détermine la valeur pour "Seuil 1" ou la restitue.
La surveillance n'est active que si la propriété Layer01Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer02Value

Description

Détermine la valeur pour "Seuil 2" ou la restitue.
La surveillance n'est active que si la propriété Layer02Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer03Value

Description

Détermine la valeur pour "Seuil 3" ou la restitue.
La surveillance n'est active que si la propriété Layer03Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer04Value

Description

Détermine la valeur pour "Seuil 4" ou la restitue.
La surveillance n'est active que si la propriété Layer04Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer05Value

Description

Détermine la valeur pour "Seuil 5" ou la restitue.
La surveillance n'est active que si la propriété Layer05Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer06Value

Description

Détermine la valeur pour "Seuil 6" ou la restitue.
La surveillance n'est active que si la propriété Layer06Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer07Value

Description

Détermine la valeur pour "Seuil 7" ou la restitue.
La surveillance n'est active que si la propriété Layer07Checked a la valeur TRUE.

Voir aussi

Objet ScreenItem (Page 132)

Bargraphes tridimensionnels (Page 179)

Propriété Layer08Value

Description

Détermine la valeur pour "Seuil 8" ou la restitue.
La surveillance n'est active que si la propriété Layer08Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer09Value

Description

Détermine la valeur pour "Seuil 9" ou la restitue.
La surveillance n'est active que si la propriété Layer09Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Layer10Value

Description

Détermine la valeur pour "Seuil 10" ou la restitue.
La surveillance n'est active que si la propriété Layer10Checked a la valeur TRUE.

Voir aussi

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété LayerDeclutteringEnable

Description

Restitue la propriété LayerDecluttering d'une vue.

LayerDecluttering permet d'afficher et de masquer des plans en fonction du zoom minimal ou maximal réglé.

BOOLEAN Accès uniquement en lecture.

Exemple :

Cet exemple affiche la propriété LayerDecluttering de la vue NewPDL1 en tant que Trace.

```
'VBS156
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
HMIRuntime.Trace "Enable: " & objScreen.LayerDeclutteringEnable & vbNewLine
```

Voir aussi

Objet Screen (Page 138)

Propriété Layers

Description

Renvoie un objet de type "Layers".

Layers (read-only)

Voir aussi

Objet Layers (Liste) (Page 128)

Objet HMIRuntime (Page 125)

Le - Li

Propriété Left

Description

Détermine en Pixel la coordonnée X d'un objet (mesurée du bord gauche supérieur de la vue) ou la restitue. La coordonnée X se réfère à l'angle gauche supérieur du rectangle entourant l'objet.

LONG (Accès en écriture et en lecture)

Exemple

L'exemple suivant déplace tous les objets de la vue "NewPDL1" de 5 Pixel vers la gauche:

```
'VBS79
Dim objScreen
Dim objScrItem
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems.Item(lngIndex).ObjectName
Set objScrItem = objScreen.ScreenItems(strName)
objScrItem.Left = objScrItem.Left - 5
Next
```

Voir aussi

Propriété Top (Page 614)
Objet ScreenItem (Page 132)

Propriété LeftComma

Description

Détermine le nombre de chiffres avant la virgule (0 à 20) ou les restitue.

Voir aussi

Bargraphe (Page 184)
Objet ScreenItem (Page 132)

Propriété LightEffect

Description

TRUE, lorsque l'effet de lumière est activé. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphes tridimensionnels (Page 179)
Objet ScreenItem (Page 132)

Propriété LimitHigh4

Description

Détermine le seuil supérieur pour "Reserve 4" ou le restitue.
La propriété CheckLimitHigh4 doit être réglée sur TRUE pour que le seuil "Reserve 4" puisse être surveillé.
Déterminez avec la propriété "TypeLimitHigh4" le type d'exploitation (en pour cent ou absolu).

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LimitHigh5

Description

Détermine le seuil supérieur pour "Reserve 5" ou le restitue.
La propriété CheckLimitHigh5 doit être réglée sur TRUE pour que le seuil "Reserve 5" puisse être surveillé.
Déterminez avec la propriété "TypeLimitHigh5" le type d'exploitation (en pour cent ou absolu).

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LimitLow4

Description

Détermine le seuil inférieur pour "Reserve 4" ou le restitue.
La propriété CheckLimitLow4 doit être réglée sur TRUE pour que le seuil "Reserve 4" puisse être surveillé.
Déterminez avec la propriété "TypeLimitLow4" le type d'exploitation (en pour cent ou absolu).

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LimitLow5

Description

Détermine le seuil inférieur pour "Reserve 5" ou le restitue.
La propriété CheckLimitLow5 doit être réglée sur TRUE pour que le seuil "Reserve 5" puisse être surveillé.
Déterminez avec la propriété "TypeLimitLow5" le type d'exploitation (en pour cent ou absolu).

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LimitMax

Description

Détermine le seuil supérieur en tant que valeur absolue selon le format de données ou le restitue.
Si la valeur à afficher dépasse le seuil supérieur, elle est identifiée comme non représentable par une série de ***.

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété LimitMin

Description

Détermine le seuil inférieur en tant que valeur absolue selon le format de données ou le restitue.
Si la valeur à afficher dépasse le seuil supérieur, elle est identifiée comme non représentable par une série de ***.

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété LineColor

Couleur Lignes de séparation-Fenêtres - LineColor

Indique la couleur pour les lignes de séparation des fenêtres. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **LineColor**. Le type de données est LONG.

Propriété LineFont

Description

TRUE, lorsque la taille de police s'adapte automatiquement à la hauteur de ligne. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété LineHeight

Description

TRUE, lorsque la hauteur de ligne peut être modifiée. BOOLEAN Accès en écriture et en lecture.

La propriété "LineHeight" n'est désactivée que si "FALSE" est paramétré pour les deux propriétés "LineHeight" et "LineFont".

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété LineJoinStyle

Description

Définit de quelle manière les angles d'un polygone tuyau sont représentés.

carré	Les tuyaux sont reliés entre eux sans arrondi au niveau des angles
rond	Les tuyaux sont arrondis au niveau des angles, côté extérieur.

Propriété LineTitle

Description

TRUE, lorsque la fenêtre d'alarmes contient une colonne où sont numérotées consécutivement les alarmes présentes. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété LineWidth (avant WinCC V7)

Description

Détermine l'épaisseur de trait de la courbe référencée par "Index". Plage des valeurs de 0 à 10.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété LineWidth

Largeur Lignes de séparation Fenêtre - LineWidth

Détermine la largeur des lignes de séparation des fenêtres en pixels.

L'attribut est dynamisable avec le nom **LineWidth**. Le type de données est LONG.

Propriété ListType

Description

Restitue pour l'objet "Liste de textes" le type de données représenté. Accès en lecture uniquement.

Plage des valeurs de 0 à 2.

0 = Décimal

1 = Binaire

2 = Bit

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Lo

Propriété LoadDataImmediately

Charger données de l'archive - LoadDataImmediately

Détermine si les valeurs de variable des archives sont chargées lors de l'ouverture d'une vue, pour la plage de temps à représenter.

Valeur	Explication
TRUE	Les valeurs archivées sont chargée lors de l'ouverture d'une vue.
FALSE	Seules les valeurs actuelles sont chargées lors de l'ouverture d'une vue.

L'attribut est dynamisable avec le nom **LoadDataImmediately**. Le type de données est BOOLEAN.

Propriété LoadDataImmediately (avant WinCC V7)

Description

TRUE, lorsqu'en cas d'ouverture de vue les valeurs variable pour la période à afficher sont chargées à partir des archives. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété LocaleID

Description

Règle la langue dans laquelle le Control s'affiche, 1031 par exemple pour l'allemand. Accès en écriture et en lecture.

Vous trouverez dans la documentation WinCC la liste des codes de langue (Index > Language Code).

Voir aussi

WinCC Slider Control (Page 275)
WinCC Gauge Control (Page 258)
WinCC Digital Analog Clock (Page 251)
Objet ScreenItem (Page 132)

LocaleSpecificSettings-propriété

Description

TRUE si, pour une langue Runtime, une police doit être affectée et formatée. BOOLEAN Accès en écriture et en lecture.

Propriété LockBackColor

Description

Détermine la couleur d'arrière-plan du bouton pour un point de mesure bloqué, ou la restitue. LONG Accès en lecture et en écriture.
La propriété LockStatus doit avoir la valeur TRUE pour que la couleur d'arrière-plan puisse s'afficher.

Voir aussi

Signalisation groupée (Page 203)
Objet ScreenItem (Page 132)

Propriété LockStatus

Description

TRUE, lorsqu'un point de mesure bloqué doit s'afficher. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)
Objet ScreenItem (Page 132)

Propriété LockText

Description

Détermine le texte des boutons pour un point de mesure verrouillé.
La propriété LockStatus doit avoir la valeur TRUE pour que le texte puisse s'afficher.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété LockTextColor

Description

Détermine la couleur du texte du bouton pour un point de mesure bloqué, ou la restitue. LONG Accès en lecture et en écriture.
La propriété LockStatus doit avoir la valeur TRUE pour que la couleur d'arrière-plan puisse s'afficher.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété Logging

Description

Renvoie un objet de type "Logging".

Logging (read-only)

Voir aussi

Objet HMIRuntime (Page 125)

Objet Logging (Page 129)

Propriété LongStrokesBold

Description

TRUE, lorsqu'en cas de représentation de l'échelle les graduations longues doivent s'afficher en gras. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LongStrokesOnly

Description

TRUE, lorsqu'en cas de représentation de l'échelle seules les graduations longues doivent s'afficher. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LongStrokesSize

Description

Détermine en Pixel la longueur des graduations d'axe ou la restitue.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LongStrokesTextEach

Description

Restitue la valeur qui définit quelles graduations doivent être marquées (1 = toutes les graduations, 2 = toutes les deux graduations, etc.) en cas de représentation de l'échelle. Accès en lecture uniquement.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété LongTimeArchiveConsistency

LongTimeArchiveConsistency

Si "LongTimeArchiveConsistency" est activé sur "Non", 1000 alarmes sont affichées dans la liste d'archives longue durée sur le poste individuel, sur le serveur ou sur le client par serveur ou par paire de serveurs redondants.

Si vous activez "LongTimeArchiveConsistency" sur "Oui", les 1000 alarmes les plus récentes sur le client de tous les serveurs ou paires de serveurs redondants sont affichés dans la liste d'archives longue durée.

L'attribut est dynamisable avec le nom **LongTimeArchiveConsistency** . Le type de données est BOOLEAN.

Propriété LongTimeArchiveConsistency (avant WinCC V7)

Description

Si "LongTimeArchiveConsistency" est mis sur "non", 1000 messages sont affichés dans la liste d'archives longue durée sur le "poste individuel", sur le serveur ou sur le client par serveur ou par paire de serveurs redondants.

Si vous réglez "LongTimeArchiveConsistency" sur "oui", les 1000 messages les plus récents sur le client de tous les serveurs ou paires de serveurs redondants sont affichés dans la liste d'archives longue durée.

Accès en écriture et en lecture.

Propriété LowerLimit

Description

WinCC Online Trend Control/WinCC Function Trend Control

TRUE, lorsque la spécification "LowerLimitColor" est utilisée pour identifier les valeurs variable (d'une courbe référencée via "Index") qui sont inférieures à la valeur "LowerLimitValue". BOOLEAN Accès en écriture et en lecture.

WinCC Online Table Control

La valeur de cet attribut ne peut être modifiée. Accès en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété LowerLimitColor

Description

WinCC Online Trend Control/WinCC Function Trend Control

Détermine la couleur qui est utilisée pour identifier les valeurs variable (d'une courbe référencée via "Index") qui sont inférieures à la valeur "LowerLimitValue". L'indication sera ou non exploitée selon la propriété "LowerLimit". La couleur doit être spécifiée sous la forme RVB. LONG Accès en écriture et en lecture.

Control Online Table

La valeur de cet attribut ne peut être modifiée. Accès en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété LowerLimitTagName

Description

Définit la limite inférieure de la zone de courbe reprise automatiquement des propriétés de variables configurées dans PCS 7. Accès en écriture et en lecture.

Propriété LowerLimitValue

Description

WinCC Online Trend Control/WinCC Function Trend Control

Les valeurs variable (d'une courbe référencée via "Index") qui sont inférieures à la valeur de "LowerLimitValue" sont identifiées par la couleur spécifiée dans "LowerLimitColor". L'indication sera ou non exploitée selon l'attribut "LowerLimit".

Control Online Table

La valeur de cet attribut ne peut être modifiée. Accès en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

1.14.4.12 M

Ma - Mc

Propriété Marker

Description

TRUE, lorsque les seuils doivent s'afficher comme valeur de graduation. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété Max

Description

Détermine la valeur absolue en cas d'affichage maximal des valeurs ou la restitue. Cette valeur s'affiche si la graduation est active.

Voir aussi

Bargraphe (Page 184)

Slider (Page 219)

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété MaximizeButton

Description

TRUE, lorsque l'objet peut être agrandi dans Runtime. Accès en lecture uniquement.

Voir aussi

Fenêtre de vue (Page 188)

Fenêtre d'application (Page 183)

Objet ScreenItem (Page 132)

Propriété MCGUBackColorOff

Description

Détermine pour l'état "Disparu Non acquitté" la couleur de l'arrière-plan pour l'état de clignotement "Arrêt", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCGUBackColorOn

Description

Détermine pour l'état "Disparu Non acquitté" la couleur de l'arrière-plan pour l'état de clignotement "Marche", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCGUBackFlash

Description

TRUE, lorsque l'arrière-plan doit clignoter lorsqu'une alarme disparaît non acquittée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCGUTextColorOff

Description

Détermine pour l'état "Disparu Non acquitté" la couleur du texte pour l'état de clignotement "Arrêt", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCGUTextColorOn

Description

Détermine pour l'état "Disparu Non acquitté" la couleur du texte pour l'état de clignotement "Marche", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCGUTextFlash

Description

TRUE, lorsque le texte doit clignoter lorsqu'une alarme disparaît non acquittée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKOBackColorOff

Description

Détermine pour l'état "Apparu" la couleur de l'arrière-plan pour l'état de clignotement "Arrêt", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKOBackColorOn

Description

Détermine pour l'état "Apparu" la couleur de l'arrière-plan pour l'état de clignotement "Marche", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKOBackFlash

Description

TRUE, lorsque l'arrière-plan doit clignoter lorsqu'une alarme apparaît. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKOTextColorOff

Description

Détermine pour l'état "Apparu" la couleur du texte pour l'état de clignotement "Arrêt", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKOTextColorOn

Description

Détermine pour l'état "Apparu" la couleur d'arrière-plan du texte pour l'état de clignotement "Marche", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKOTextFlash

Description

TRUE, lorsque le texte doit clignoter lorsqu'une alarme apparaît. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKQBackColorOff

Description

Détermine pour l'état "Disparu Acquitté" la couleur de l'arrière-plan pour l'état de clignotement "Arrêt", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKQBackColorOn

Description

Détermine pour l'état "Disparu Acquitté" la couleur de l'arrière-plan pour l'état de clignotement "Marche", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKQBackFlash

Description

TRUE, lorsque l'arrière-plan doit clignoter lorsqu'une alarme disparaît acquittée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKQTextColorOff

Description

Détermine pour l'état "Disparu Acquitté" la couleur du texte pour l'état de clignotement "Arrêt", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKQTextColorOn

Description

Détermine pour l'état "Disparu Acquitté" la couleur d'arrière-plan du texte pour l'état de clignotement "Marche", ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCKQTextFlash

Description

TRUE, lorsque le texte doit clignoter lorsqu'une alarme disparaît acquittée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MCText

Description

Détermine le texte pour la classe d'alarmes respective ou le restitue.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Me

Propriété MeasurePoints

Description

La propriété "Index" référence une courbe. "MeasurePoints" détermine le nombre de points de mesure à représenter. L'indication n'est exploitée que si la propriété "TimeAxis" porte la valeur "-1".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété-MenuToolBarConfig

Description

Charge le fichier de configuration indiqué avec un menu personnalisé et des barres d'outilsou rend le nom du fichier de configuration. STRING-Accès en écriture et en lecture.

Voir aussi

Fenêtre de vue (Page 188)

Objet HMIRuntime (Page 125)

Propriété MessageBlockAlign

Alignement Blocs d'alarme - MessageBlockAlign

Détermine comment les contenus du bloc d'alarme sélectionné sont alignés dans le tableau.

Pour pouvoir modifier l'alignement, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	à gauche	Les contenus du bloc d'alarme sélectionné sont alignés à gauche.
1	centré	Les contenus du bloc d'alarme sélectionné sont centrés.
2	vers la droite	Les contenus du bloc d'alarme sélectionné sont alignés à droite.

L'attribut est dynamisable avec le nom **MessageBlockAlign**. Le type de données est LONG.

Propriété MessageBlockAutoPrecisions

Chiffres après la virgule Automatique - MessageBlockAutoPrecisions

Détermine si le nombre de chiffres après la virgule est défini automatiquement.

Valeur	Explication
TRUE	Le nombre de chiffres après la virgule est défini automatiquement. La valeur du champ "Chiffres après la virgule" reste sans effet.
FALSE	La valeur du champ "Chiffres après la virgule" est effective.

L'attribut est dynamisable avec le nom **MessageBlockAutoPrecisions**. Le type de données est BOOLEAN.

Propriété MessageBlockCaption

Désignation - MessageBlockCaption

Permet de désigner dans la fenêtre d'alarmes le titre de colonne pour le bloc d'alarme sélectionné. La désignation entrée est active dans toutes les langues Runtime.

Pour pouvoir modifier la désignation, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

L'attribut est dynamisable avec le nom **MessageBlockCaption**. Le type de données est STRING.

Propriété MessageBlockCount

MessageBlockCount

Cet attribut définit le nombre des blocs d'alarme existants et disponibles pour la liste d'alarmes et la liste d'occurrences.

L'attribut est dynamisable avec le nom **MessageBlockCount** . Le type de données est LONG.

Propriété MessageBlockDateFormat

Format de la date - MessageBlockDateFormat

Détermine quel format de date sera utilisé pour l'affichage des alarmes.

Pour pouvoir modifier le format de date, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

Les formats de date suivants sont disponibles :

Valeur	Explication
Automatique	Le format de la date est automatiquement défini.
dd.MM.yy	Jour.Mois.Année, par exemple 24.12.07.
dd.MM.yyyy	Jour.Mois.Année, par exemple 24.12.2007.
dd/MM/yy	Jour/Mois/Année, par exemple 24/12/07.
dd/MM/yyyy	Jour/Mois/Année, par exemple 24/12/2007.

L'attribut est dynamisable avec le nom **MessageBlockDateFormat**. Le type de données est STRING.

Propriété MessageBlockExponentialFormat

Représentation exponentielle - MessageBlockExponentialFormat

Détermine si les valeurs du bloc d'alarme sélectionné s'affichent en représentation exponentielle.

Valeur	Explication
TRUE	Les valeurs sont affichées en représentation exponentielle.
FALSE	Les valeurs sont affichées en représentation décimale.

L'attribut est dynamisable avec le nom **MessageBlockExponentialFormat**. Le type de données est BOOLEAN.

Propriété MessageBlockFlashMode

Mode de clignotement - MessageBlockFlashMode

Détermine si le contenu du bloc d'alarme sélectionné clignote à l'arrivée d'une alarme en Runtime. L'option "Clignotement" doit être activée.

Pour pouvoir modifier le réglage, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

Valeur	Description	Explication
0	Standard	En cas de clignotement, la couleur du texte alterne entre la couleur standard et la couleur de clignotement.
1	Alternance couleur de l'arrière-plan/ couleur du texte	En cas de clignotement, il y a une alternance entre la couleur de l'arrière-plan et celle du texte de l'alarme. Vous configurez les couleurs d'alarmes dans l'éditeur AlarmLogging, dans le type d'alarme.
2	Alternance des couleurs d'alarmes/ couleurs des tableaux	En cas de clignotement, il y a une alternance des couleurs d'alarmes et des couleurs qui ont été configurées pour les tableaux. Vous configurez les couleurs d'alarmes dans l'éditeur AlarmLogging, dans le type d'alarme. Vous paramétrez les couleurs des tableaux dans l'onglet "Représentation" de l'AlarmControl.

L'attribut est dynamisable avec le nom **MessageBlockFlashMode**. Le type de données est LONG.

Propriété MessageBlockFlashOn

Clignotement - MessageBlockFlashOn

Détermine si le contenu du bloc d'alarme sélectionné clignote à l'arrivée d'une alarme en Runtime.

Pour pouvoir modifier le réglage, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

Valeur	Explication
TRUE	Le contenu du bloc d'alarmes clignote.
FALSE	Le contenu du bloc d'alarmes ne clignote pas.

L'attribut est dynamisable avec le nom **MessageBlockFlashOn**. Le type de données est BOOLEAN.

Propriété MessageBlockHideText

Contenu comme texte - MessageBlockHideText

Détermine si le contenu du bloc d'alarme sélectionné doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le contenu ne s'affiche pas sous forme de texte. L'option n'est pas activée.
FALSE	Le contenu s'affiche sous forme de texte. L'option est activée.

L'attribut est dynamisable avec le nom **MessageBlockHideText**. Le type de données est BOOLEAN.

Propriété MessageBlockHideTitleText

Titre comme texte - MessageBlockHideTitleText

Détermine si le titre du bloc d'alarme sélectionné doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le titre ne s'affiche pas sous forme de texte. L'option n'est pas activée.
FALSE	Le titre s'affiche sous forme de texte. L'option est activée.

L'attribut est dynamisable avec le nom **MessageBlockHideTitleText**. Le type de données est BOOLEAN.

Propriété MessageBlockId

MessageBlockId

Affectation définie du numéro d'identification et bloc d'alarme dans WinCC AlarmControl

L'attribut est dynamisable avec le nom **MessageBlockID**. Le type de données est LONG.

Propriété MessageBlockInvertUseMessageColor

MessageBlockInvertUseMessageColor

Cet attribut détermine pour le bloc d'alarme, si les couleurs d'alarmes sont affichées, ou non, contrairement au paramètre central de l'AlarmControl. Vous avez par exemple paramétré la propriété "UseMessageColor" de l'AlarmControl sur "FALSE". Vous avez paramétré la propriété "MessageBlockInvertUseMessageColor" sur "TRUE" pour un bloc d'alarmes. Les couleurs d'alarmes sont donc affichées en mode Runtime pour ce bloc d'alarmes.

Valeur	Explication
TRUE	Les couleurs d'alarmes sont affichées, ou non, contrairement au paramètre central, dans "UseMessageColor" pour le bloc d'alarmes.
FALSE	Les couleurs d'alarmes sont affichées, ou non, comme le paramètre central, dans "UseMessageColor" pour le bloc d'alarmes.

L'attribut est dynamisable avec le nom **MessageBlockInvertUseMessageColor**. Le type de données est BOOLEAN.

Propriété MessageBlockIndex

MessageBlockIndex

Cet attribut fait référence à un bloc d'alarme. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un bloc d'alarme précis.

Les valeurs admissibles pour "MessageBlockIndex" et "MessageBlockCount" se situent entre 0 et -1. L'attribut "MessageBlockCount" indique le nombre des blocs d'alarme existants.

L'attribut est dynamisable avec le nom **MessageBlockIndex**. Le type de données est LONG.

Propriété MessageBlockLeadingZeros

Nombre de positions - MessageBlockLeadingZeros

Détermine le nombre de zéros non significatifs pour le contenu du bloc d'alarme. Le nombre maximal s'élève à "11". La valeur "0" a pour effet que l'option "Avec zéros non significatifs" est désactivée.

Pour pouvoir modifier le réglage, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

L'attribut est dynamisable avec le nom **MessageBlockLeadingZeros**. Le type de données est LONG.

Propriété MessageBlockLength

Longueur en caractères - MessageBlockLength

Détermine la longueur en caractères pour le contenu du bloc d'alarme sélectionné.

Pour pouvoir modifier la longueur, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

L'attribut est dynamisable avec le nom **MessageBlockLength**. Le type de données est LONG.

Propriété MessageBlockName

Nom d'objet - MessageBlockName

Affiche le nom pour le bloc d'alarme sélectionné. Vous ne pouvez pas modifier le nom.

Le type de données est STRING.

Propriété MessageBlockPrecisions

Chiffres après la virgule - MessageBlockPrecisions

Définit le nombre de chiffres après la virgule des valeurs du bloc d'alarme sélectionné. Vous ne pouvez entrer la valeur que si l'option "Automatique" est désactivée.

L'attribut est dynamisable avec le nom **MessageBlockPrecisions**. Le type de données est SHORT.

Propriété MessageBlockSelected

Blocs d'alarme existants - MessageBlockSelected

Les blocs d'alarme existants sont des blocs qui sont disponibles en Runtime pour la liste d'alarmes ou la liste d'occurrences.

Sur l'onglet "Blocs d'alarme", activez les blocs des blocs d'alarme existants dont vous avez besion dans le contrôle. Sur les onglets "Liste d'occurrences" et "Liste d'alarmes", vous configurez la liste d'alarmes et la liste d'occurrences à partir des blocs existants.

Pour pouvoir modifier le réglage, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

L'attribut est dynamisable avec le nom **MessageBlockSelected**. Le type de données est BOOLEAN.

Propriété MessageBlockShowDate

Affichage de la date - MessageBlockShowDate

Détermine si la date est également affichée à côté de l'heure dans le bloc d'alarme "Heure".

Valeur	Explication
TRUE	La date et l'heure sont affichées.
FALSE	L'heure est affichée.

L'attribut est dynamisable avec le nom **MessageBlockShowDate**. Le type de données est BOOLEAN.

Propriété MessageBlockShowIcon

Contenu comme symbole - MessageBlockShowIcon

Détermine si le contenu du bloc d'alarme sélectionné doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le contenu s'affiche sous forme de symbole.
FALSE	Le contenu ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **MessageBlockShowIcon**. Le type de données est BOOLEAN.

Propriété MessageBlockShowTitleIcon

Titre comme symbole - MessageBlockShowTitleIcon

Détermine si le titre du bloc d'alarme sélectionné doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le titre s'affiche sous forme de symbole.
FALSE	Le titre ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **MessageBlockShowTitleIcon**. Le type de données est BOOLEAN.

Propriété MessageBlockTextId

ID de texte - MessageBlockTextId

Permet de désigner le bloc d'alarme sélectionné à l'aide d'un numéro d'identification de texte provenant de la Text Library. La désignation s'adapte automatiquement en cas de changement de langue Runtime.

Pour pouvoir modifier le réglage, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

L'attribut est dynamisable avec le nom **MessageBlockTextId**. Le type de données est LONG.

Propriété MessageBlockTimeFormat

MessageBlockTimeFormat

Détermine quel format de l'heure ou format de durée sera utilisé pour l'affichage des alarmes.

Pour pouvoir modifier le réglage, l'option "Adopter les paramétrages de projet" doit être désactivée ou "ApplyProjectSettings" doit être mise sur "FALSE.

Les formats d'heure suivants sont disponibles :

Valeur	Explication
Automatique	Le format de l'heure est automatiquement défini.
HH:mm:ss	Heures:Minutes:Secondes, par exemple 15:35:44
HH:mm:ss.ms	Heures:Minutes:Secondes.Millisecondes, par exemple 15:35:44.240.
hh:mm:ss tt	Heures:Minutes:Secondes AM/PM, par exemple 03:35:44 PM.
hh:mm:ss.ms tt	Heures:Minutes:Secondes.Millisecondes AM/PM, par exemple 03:35:44.240 PM.

Les formats de durée suivants sont disponibles :

Valeur	Explication
Automatique	Le format de la durée est automatiquement défini.
d H:mm:ss	Jour Heures:Minutes:Secondes, par exemple 1 2:03:55.
H:mm:ss.	Heures:Minutes:Secondes, par exemple 26:03:55.
m:ss	Minutes:Secondes, exemple : 1563:55.
s	Secondes, par exemple 93835.

L'attribut est dynamisable avec le nom **MessageBlockTimeFormat**. Le type de données est STRING.

Propriété MessageBlockType

MessageBlockType

Cet attribut affiche l'appartenance du bloc d'alarme.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Bloc système	Le bloc d'alarme fait partie des blocs système.
1	Bloc de texte	Le bloc d'alarme fait partie des blocs de texte utilisateur.
2	Bloc de valeurs process	Le bloc d'alarme fait partie des blocs de valeur de process.
3	Bloc de liste d'occurrences	Le bloc d'alarme fait partie des blocs d'alarme de la liste d'occurrences.

L'attribut est dynamisable avec le nom **MessageBlockType** . Le type de données est LONG.

Propriété MessageClass

Description

Détermine le type d'alarme (Alarm High, Alarm Low, Warning High, Warning Low, ...) pour lequel les paramètres "Texte affiché", "Apparu", "Apparu Acquitté" et "Disparu Non acquitté -" doivent être configurés.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété MessageColumnAdd

MessageColumnAdd

Accepte dans la liste des blocs d'alarme sélectionnés le bloc d'alarme marqué issu de la liste des blocs d'alarme existants.

L'attribut est dynamisable avec le nom **MessageColumnAdd** . Le type de données est STRING.

Propriété MessageColumnCount

MessageColumnCount

Cet attribut indique le nombre des blocs d'alarme affichés dans la liste d'alarmes en Runtime.

L'attribut est dynamisable avec le nom **MessageColumnCount** . Le type de données est LONG.

Propriété MessageColumnIndex

MessageColumnIndex

Cet attribut fait référence à un bloc d'alarme sélectionné pour la liste d'alarmes. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un bloc d'alarme précis de la liste d'alarmes.

Les valeurs admissibles pour "MessageColumnIndex" et "MessageColumnCount" se situent entre 0 et -1. L'attribut "MessageColumnCount" indique le nombre des blocs d'alarme sélectionnés pour la liste d'alarmes.

L'attribut "MessageColumnIndex" est dynamisable via l'attribut **MessageColumnRepos**. Le type de données est LONG.

Propriété MessageColumnName

MessageColumnName

Cet attribut affiche le nom du bloc d'alarme de la liste d'alarmes qui est référencé à l'aide de l'attribut "MessageColumnIndex". Vous ne pouvez pas modifier le nom.

L'attribut est dynamisable avec le nom **MessageColumnName**. Le type de données est STRING.

Propriété MessageColumnRemove

MessageColumnRemove

Supprime le bloc d'alarme marqué dans la liste des blocs d'alarme sélectionnés et l'insère dans la liste des blocs d'alarme existants.

L'attribut est dynamisable avec le nom **MessageColumnRemove** . Le type de données est STRING.

Propriété MessageColumnRepos

Haut/bas - MMessageColumnRepos/HitlistColumnRepos

Modifie l'ordre des blocs d'alarme. "Haut" et "Bas" déplacent le bloc d'alarme sélectionné dans la liste vers le haut ou le bas. Ce qui a pour conséquence que le bloc d'alarme est positionné plus en avant ou plus en arrière dans le contrôle en Runtime.

L'attribut pour la liste d'occurrences est dynamisable avec le nom **HitlistColumnRepos** .

L'attribut pour la liste d'alarmes est dynamisable avec le nom **MessageColumnRepos**.

Le type de données est LONG.

Propriété MessageColumnSort

MessageColumnSort

Cet attribut définit comment le bloc d'alarme référencé dans "MessageColumnIndex" est trié.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	non	Aucun tri
1	croissant	Tri croissant de la plus petite à la plus grande valeur
2	décroissant	Tri décroissant de la plus grande à la plus petite valeur

L'attribut est dynamisable avec le nom **MessageColumnSort** . Le type de données est LONG.

Propriété MessageColumnSortIndex

MessageColumnSortIndex

Cet attribut indique l'ordre de tri du bloc d'alarme référencé dans "MessageColumnIndex". Si vous activez la valeur sur "0", le critère de tri est supprimé dans "MessageColumnSort".

L'attribut est dynamisable avec le nom **MessageColumnSortIndex**. Le type de données est LONG.

Propriété MessageColumnVisible

Blocs d'alarme sélectionnés - MessageColumnVisible/HitlistColumnVisible

Blocs d'alarme sélectionnés de la liste des alarmes ou de la liste d'occurrences à afficher en runtime :. Cet attribut définit comment s'affiche le bloc d'alarme référencé dans "MessageColumnIndex" ou "HitlistColumnIndex".

L'attribut pour la liste d'alarmes est dynamisable avec le nom **MessageColumnVisible**.

L'attribut pour la liste d'occurrences est dynamisable avec le nom **HitlistColumnVisible**.

Le type de données est BOOLEAN.

Propriété MessageListType

Liste active à l'ouverture de vue - MessageListType

Champ de sélection pour déterminer la liste active à l'ouverture d'une vue.

Valeur	Description	Explication
0	Liste d'alarmes	Les alarmes actuellement présentes sont représentées à l'ouverture d'une vue.
1	Liste d'archives courtes	A l'ouverture d'une vue, une liste d'archives courte durée contenant les alarmes archivées est représentée. L'affichage est immédiatement rafraîchi lorsqu'une alarme arrive.
2	Liste d'archives longues	A l'ouverture d'une vue, une liste d'archives courte durée contenant les alarmes archivées est représentée.
3	Liste de verrouillages	Seules les alarmes actuellement verrouillées sont représentées à l'ouverture d'une vue.
4	Liste d'occurences	Les informations statistiques configurées sont représentées à l'ouverture d'une vue.
5	Liste des alarmes à masquer	Les alarmes à masquer sont représentées à l'ouverture d'une vue.

L'attribut est dynamisable avec le nom **MessageListType**. Le type de données est LONG.

Mi - Ms

Propriété Min

Description

Détermine la valeur absolue en cas d'affichage minimal des valeurs ou la restitue. Cette valeur s'affiche si la graduation est active.

Voir aussi

Slider (Page 219)

Bargraphe (Page 184)

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété MinuteNeedleHeight

Description

Détermine la longueur de l'aiguille des minutes sur l'horloge analogique ou la restitue. La longueur est spécifiée en pour cent par rapport à la demi-longueur du côté le plus court du rectangle d'arrière-plan. Accès en écriture et en lecture.

Exemple:
Si le côté le plus court du rectangle d'arrière-plan était long de 100 pixels.
La longueur de l'aiguilles des minutes serait 80.
D'où une aiguille des minutes longue de (100 pixels / 2) * 0,8 = 40 pixels.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété MinuteNeedleWidth

Description

Détermine la largeur de l'aiguille des minutes sur l'horloge analogique ou la restitue. La largeur est spécifiée en pour cent par rapport à la longueur de l'aiguille des minutes multipliée par deux.

Exemple:
La longueur de l'aiguille des minutes est 40 pixels.
La largeur de l'aiguille des minutes est 8.
D'où une aiguille des minutes large de 40 pixels * 2 * 0,08 = 6 pixels (arrondi).

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété Moveable

Description

TRUE, lorsque l'objet peut être déplacé dans Runtime. Accès en lecture uniquement.

Voir aussi

Fenêtre de vue (Page 188)

Fenêtre d'application (Page 183)

Objet ScreenItem (Page 132)

Propriété Moveable

Déplaçable - Moveable

Détermine si le contrôle est déplaçable en Runtime.

Valeur	Explication
TRUE	Le contrôle peut être déplacé en Runtime.
FALSE	Le contrôle ne peut pas être déplacé en Runtime.

L'attribut est dynamisable avec le nom **Moveable**. Le type de données est BOOLEAN.

Propriété MsgCtrlFlags

Description

Détermine l'ordre du tri dans Alarm Control. Accès en écriture et en lecture.

- 0: Tri par ordre ascendant dans la plage de temps définie dans la colonne de temps : les alarmes les plus anciennes sont affichées en haut de la fenêtre d'alarmes.
- 1: Tri par ordre descendant dans la plage de temps définie dans la colonne de temps : les alarmes les plus anciennes sont affichées en bas de la fenêtre d'alarmes et les plus récentes en haut. Avec ce paramètre, la propriété "AutoScroll" se désactive automatiquement étant donné que l'alarme la plus récente peut sinon disparaître de la zone d'affichage de la fenêtre d'alarmes.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété MsgFilterSQL (avant WinCC V7)

Description

Détermine une instruction SQL pour la sélection des alarmes affichées dans la fenêtre d'alarmes. Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété MsgFilterSQL

MsgFilterSQL

Cet attribut définit une sélection personnalisée des alarmes pour une ou plusieurs instructions SQL. Plusieurs sélections personnalisées sont reliées par "OR". Si vous avez défini une sélection fixe "DefaultMsgFilterSQL", les instructions SQL de "DefaultMsgFilterSQL" et "MsgFilterSQL" sont reliées par "AND".

L'attribut est dynamisable avec le nom **MsgFilterSQL**. Le type de données est STRING.

1.14.4.13 N

Propriété Name

Description des objets Layer et Tag

Restitue le nom de l'objet. STRING (readonly)

- Pour les variables, il s'agit du nom des variables sans ServerPrefix ni VariablePrefix
- Pour les niveaux, il s'agit du nom des niveaux

Variables

La propriété "Name" d'une variable permet d'accéder à la variable via la liste des Tags. Le nom d'une variable peut contenir un ServerPrefix. Les noms des variables dans WinCC sont établis selon le schéma suivant :

<Préfixe de serveur>::<Préfixe de variable><Nom de la variable>

Si c'est simplement le nom de la variable qui est spécifié, Serverprefix et Variableprefix sont sortis du contexte de la fenêtre de vue.

Si l'on spécifie pour la variable un Serverprefix dans le nom de variable, Variable et Serverprefix du contexte sont ignorés et c'est le Serverprefix joint qui est utilisé.

Description de WinCC Function Trend Control

La propriété "Index" référence une courbe. "Name" détermine le nom de cette courbe.

Description de l'objet Project

Renvoie le nom du projet Runtime actuel. STRING (readonly)

Exemple

L'exemple suivant affiche le nom du projet Runtime actuel en tant que Trace :

```
'VBS160
HMIRuntime.Trace "Name: " & HMIRuntime.ActiveProject.Name & vbNewLine
```

Description de l'objet DataItem

Renvoie le nom de l'objet DataItem.

Voir aussi

Propriété ActiveProject (Page 294)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet Tag (Page 144)

Segment d'ellipse (Page 156)

Objet Layer (Page 127)

Objet DataItem (Page 120)

Propriété NeedleColor

Description

Définit la couleur de l'aiguille ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété NormalColor

Description

Détermine la couleur de la plage normale sur l'échelle. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété NumberLines

Description

Liste de textes

Définit pour l'objet Liste de texte combien de lignes la liste de sélection doit contenir, ou retourne la valeur. Si la quantité de texte configurée est supérieure à cette valeur; la liste de sélection est complétée par une barre de défilement verticale.

Champ de combinaison et champ de liste

Définit pour les objets Champ de combinaison et Champ de liste combien d'entrées l'objet doit contenir, ou retourne la valeur. Vous pouvez paramétrer au maximum 100 000 lignes.

La valeur de l'attribut "Nombre de lignes" indique en même temps le seuil supérieur de l'attribut "Index" dans le groupe de propriétés "Police". La modification de la valeur entraîne les conséquences suivantes :

- Augmentation du nombre : de nouvelles lignes sont ajoutées en bas. Vous changez l'inscription standard du nouveau champ à l'aide de l'attribut "Texte" dans le groupe de propriétés "Police".
- Diminution du nombre : Toutes les lignes pour lesquelles la valeur de l'attribut "Index" est supérieure au nouveau nombre sont supprimées.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété NumItems

Description

Restitue le nombre de couples de colonnes ou de courbes configurées (visibles et invisibles) dans la fenêtre. Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

1.14.4.14 O

Ob - On

Propriété Object

Description

Si vous utilisez un Control n'ayant pas été mis à disposition par WinCC, il se peut que les propriétés apportées dans ce Control portent les mêmes noms que les propriétés ScreenItem générales. Dans ce cas, ce sont les propriétés ScreenItem qui ont priorité. Vous pouvez accéder aux propriétés "cachées" d'un Control d'une autre marque par le biais de la propriété supplémentaire "object".

Exemple

Vous accédez par exemple aux propriétés d'un Control d'une autre marque sous la forme suivante:

Control.object.type

Si vous utilisez uniquement la forme

Control.type

ce sont les propriétés de l'objet ScreenItem qui seront utilisées si les noms sont identiques.

Voir aussi

Contrôles (Page 224)

Objet ScreenItem (Page 132)

Propriété ObjectName

Description

Restitue le nom de l'objet.

- Pour les objets graphiques, il s'agit du nom d'objet
- Pour les vues, il s'agit du nom de vue

STRING (readonly)

Exemple

L'exemple suivant affiche le nom de tous les objets contenus dans la vue "NewPDL1":

```
'VBS80
Dim objScreen
Dim lngIndex
Dim lngAnswer
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems.Item(lngIndex).ObjectName
lngAnswer = MsgBox("Name of object " & lngIndex & ": " & strName, vbOKCancel)
If vbCancel = lngAnswer Then Exit For
Next
```

Vues

Déterminer le nom de la vue directement via la propriété "ObjectName":

```
'VBS81
MsgBox "Screenname: " & HMIRuntime.ActiveScreen.ObjectName
```

Voir aussi

Objet Screen (Page 138)

Objet ScreenItem (Page 132)

Propriété ObjectSizeDeclutteringEnable

Description

Restitue la propriété ObjectSizeDecluttering d'une vue.

Si la propriété ObjectSizeDecluttering est activée, seuls sont affichés les objets se trouvant dans une plage déterminée.

Vous déterminez la limite supérieure/inférieure pour la plage à afficher dans le Graphics Designer, sous "Outils > Paramétrages > Afficher/Masquer".

BOOLEAN Accès uniquement en lecture.

Exemple :

Cet exemple affiche la propriété Decluttering de la vue NewPDL1 en tant que Trace.

```
'VBS157
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
HMIRuntime.Trace "Min: " & objScreen.ObjectSizeDeclutteringMin & vbNewLine
HMIRuntime.Trace "Max: " & objScreen.ObjectSizeDeclutteringMax & vbNewLine
HMIRuntime.Trace "Enable: " & objScreen.LayerDeclutteringEnable & vbNewLine
```

Voir aussi

Objet Screen (Page 138)

Propriété ObjectSizeDeclutteringMax

Description

ObjectSizeDeclutteringMax permet de lire la plage supérieure d'une vue.

Les objets qui sont supérieurs à la taille spécifiée en pixels ne s'affichent plus si la propriété ObjectSizeDecluttering est activée.

LONG Accès uniquement en lecture.

Exemple :

Cet exemple affiche la propriété Decluttering de la vue NewPDL1 en tant que Trace.

```
'VBS157
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
HMIRuntime.Trace "Min: " & objScreen.ObjectSizeDeclutteringMin & vbNewLine
HMIRuntime.Trace "Max: " & objScreen.ObjectSizeDeclutteringMax & vbNewLine
HMIRuntime.Trace "Enable: " & objScreen.LayerDeclutteringEnable & vbNewLine
```

Voir aussi

Objet Screen (Page 138)

Propriété ObjectSizeDeclutteringMin

Description

ObjectSizeDeclutteringMin permet de lire la plage inférieure d'une vue.

Les objets qui sont inférieurs à la taille spécifiée en pixels ne s'affichent plus si la propriété ObjectSizeDecluttering est activée.

LONG Accès uniquement en lecture.

Exemple :

Cet exemple affiche la propriété Decluttering de la vue NewPDL1 en tant que Trace.

```
'VBS157
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
HMIRuntime.Trace "Min: " & objScreen.ObjectSizeDeclutteringMin & vbNewLine
HMIRuntime.Trace "Max: " & objScreen.ObjectSizeDeclutteringMax & vbNewLine
HMIRuntime.Trace "Enable: " & objScreen.LayerDeclutteringEnable & vbNewLine
```

Voir aussi

Objet Screen (Page 138)

Propriété OffsetLeft

Description

Détermine la distance entre la vue et le bord gauche de la fenêtre de vue, ou la restitue.

La vue est représentée comme découpée de la fenêtre. Les barres de défilement se trouvent sur les bords gauche et supérieur de la vue. Si vous voulez afficher la vue dans la fenêtre de vue par déplacement horizontal et vertical des barres de défilement, utilisez les propriétés "ScrollPositionX" et "ScrollPositionY" pour le déplacement.

Voir aussi

Propriété ScrollPositionY (Page 535)

Propriété ScrollPositionX (Page 534)

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété OffsetTop

Description

Détermine la distance entre la vue et le bord supérieur de la fenêtre de vue, ou la restitue.

La vue est représentée comme découpée de la fenêtre. Les barres de défilement se trouvent sur les bords gauche et supérieur de la vue. Si vous voulez afficher la vue dans la fenêtre de vue par déplacement horizontal et vertical des barres de défilement, utilisez les propriétés "ScrollPositionX" et "ScrollPositionY" pour le déplacement.

Voir aussi

Propriété ScrollPositionY (Page 535)

Propriété ScrollPositionX (Page 534)

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété OneY

Description

TRUE quand seul l'axe Y de la courbe est affiché au premier-plan au lieu de tous les axes Y des courbes représentées. BOOLEAN Accès en écriture et en lecture.

Propriété Online (avant WinCC V7)

Description

Sert à démarrer ou à arrêter le cycle de mise à jour.

- 0: La représentation mise à jour est arrêtée. Les valeurs sont sauvegardées en mémoire intermédiaire et représentées lorsque le bouton est à nouveau activé.
- -1: La représentation mise à jour est reprise.

Voir aussi

Objet ScreenItem (Page 132)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Propriété Online

Lancer la mise à jour - En ligne

Détermine si la représentation des valeurs est mise à jour à l'ouverture de la vue en Runtime.

Valeur	Description
TRUE	Les valeurs sont mises à jour à l'ouverture de la vue.
FALSE	Les valeurs ne sont pas mises à jour à l'ouverture de la vue.

L'attribut est dynamisable avec le nom **Online**. Le type de données est BOOLEAN.

Propriété OnTop

Description

TRUE, lorsque l'objet reste toujours au premier plan en Runtime. Accès en lecture uniquement.

Voir aussi

Fenêtre de vue (Page 188)

Fenêtre d'application (Page 183)

Objet ScreenItem (Page 132)

Op

Propriété OperationMessage

Description

TRUE, lorsqu'une alarme doit être émise après qu'une commande ait été exécutée. BOOLEAN Accès en lecture et en écriture.
La commande est envoyée au système d'alarme et archivée. Une alarme peut p. ex. être affichée sur une ligne d'alarme par le système d'alarme.

Particularité pour Champ E/S, Liste des textes et Slider

La raison pour la commande ne peut être entrée que si la propriété "OperationReport" est positionnée sur TRUE.

Voir aussi

Slider (Page 219)

Liste de textes (Page 206)

Case d'option (Page 215)

Case à cocher (Page 213)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété OperatorMessageID

OperatorMessageID

Affectation définie du numéro d'identification et de l'événement de déclenchement dans WinCC OnlineTableControl :

Valeur	Description	Explication
5	EditValue	Evénement de déclenchement "Modifier valeur d'archive"
6	InsertValue	Evénement de déclenchement "Générer valeur d'archive"

L'attribut est dynamisable avec le nom **OperatorMessageID**. Le type de données est LONG.

Propriété OperatorMessageIndex

OperatorMessageIndex

Référence l'événement d'une modification de valeur d'archive pour une alarme de conduite. En cas d'utilisation de l'attribut, vous pouvez affecter les valeurs d'autres attributs à une alarme de conduite précise.

Les valeurs suivantes sont disponibles :

Valeur	Explication
0	Evénement de déclenchement "Modifier valeur d'archive"
1	Evénement de déclenchement "Générer valeur d'archive"

L'attribut est dynamisable avec le nom **OperatorMessageIndex**. Le type de données est LONG.

Propriété OperatorMessageName

Nom d'objet - OperatorMessageName

Cet attribut affiche le nom référencié avec l'attribut "OperatorMessageIndex" en cas d'événements d'alarme pour les alarmes de conduite Vous ne pouvez pas modifier le nom.

Les noms suivants sont disponibles pour les événements d'alarme :

Valeur	Explication
Lock	L'événement d'alarme "Verrouiller"
Unlock	L'événement d'alarme "Déverrouiller"
Hide	L'événement d'alarme "Masquer"
Unhide	L'événement d'alarme "Afficher"
Quit	L'événement d'alarme "Acquitter"

L'attribut est dynamisable avec le nom **OperatorMessageName** . Le type de données est STRING.

Voir aussi

Comment configurer les alarmes de conduite

Propriété OperatorMessageNumber

Numéro d'alarme - OperatorMessageNumber

Définissez un numéro d'alarme pour l'alarme de conduite de l'événement d'alarme sélectionné si vous ne souhaitez pas utiliser l'alarme de conduite de WinCC.

L'attribut est dynamisable avec le nom **OperatorMessageNumber**. Le type de données est LONG.

Propriété OperatorMessageSelected

Alarmes de conduite pour - OperatorMessageSelected

Activez dans la liste les événements d'alarme pour lesquels des alarmes de conduite sont déclenchées.

L'attribut est dynamisable avec le nom **OperatorMessageSelected**. Le type de données est BOOLEAN.

Propriété OperatorMessageSource1

Source - OperatorMessageSource1

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 1" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 1" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "1" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource1**. Le type de données est STRING.

Propriété OperatorMessageSource2

Source - OperatorMessageSource2

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 2" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 2" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "2" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource2**. Le type de données est STRING.

Propriété OperatorMessageSource3

Source - OperatorMessageSource3

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 3" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 3" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "3" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource3**. Le type de données est STRING.

Propriété OperatorMessageSource4

Source - OperatorMessageSource4

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 4" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 4" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "4" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource4**. Le type de données est STRING.

Propriété OperatorMessageSource5

Source - OperatorMessageSource5

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 5" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 5" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "5" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource5**. Le type de données est STRING.

Propriété OperatorMessageSource6

Source - OperatorMessageSource6

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 6" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 6" de l'alarme de conduite. Sélectionnez à

cet effet sous la valeur de process "6" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource6**. Le type de données est STRING.

Propriété OperatorMessageSource7

Source - OperatorMessageSource7

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 7" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 7" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "7" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource7**. Le type de données est STRING.

Propriété OperatorMessageSource8

Source - OperatorMessageSource8

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 8" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 8" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "8" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource8**. Le type de données est STRING.

Propriété OperatorMessageSource9

Source - OperatorMessageSource9

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 9" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 9" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "9" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource9**. Le type de données est STRING.

Propriété OperatorMessageSource10

Source - OperatorMessageSource10

Définissez un bloc d'alarme de l'alarme utilisée qui est ajoutée au "Bloc de valeur de process 10" de l'alarme de conduite configurée ici.

Exemple : vous voulez générer une alarme de conduite lors du verrouillage d'une alarme. Le contenu du "Bloc de texte utilisateur 1" de l'alarme verrouillée, par ex. "Moteur défectueux", doit s'afficher dans le "Bloc de valeur de process 10" de l'alarme de conduite. Sélectionnez à cet effet sous la valeur de process "10" "Bloc de texte utilisateur 1" en tant que bloc d'alarme de l'alarme utilisée.

L'attribut est dynamisable avec le nom **OperatorMessageSource10**. Le type de données est STRING.

Propriété OperatorMessageSourceType1

Transfert sous forme de - OperatorMessageSourceType1

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType1**. Le type de données est LONG.

Propriété OperatorMessageSourceType2

Transfert sous forme de - OperatorMessageSourceType2

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType2**. Le type de données est LONG.

Propriété OperatorMessageSourceType3

Transfert sous forme de - OperatorMessageSourceType3

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType3**. Le type de données est LONG.

Propriété OperatorMessageSourceType4

Transfert sous forme de - OperatorMessageSourceType4

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType4**. Le type de données est LONG.

Propriété OperatorMessageSourceType5

Transfert sous forme de - OperatorMessageSourceType5

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType5**. Le type de données est LONG.

Propriété OperatorMessageSourceType6

Transfert sous forme de - OperatorMessageSourceType6

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType6**. Le type de données est LONG.

Propriété OperatorMessageSourceType7

Transfert sous forme de - OperatorMessageSourceType7

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType7**. Le type de données est LONG.

Propriété OperatorMessageSourceType8

Transfert sous forme de - OperatorMessageSourceType8

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType8**. Le type de données est LONG.

Propriété OperatorMessageSourceType9

Transfert sous forme de - OperatorMessageSourceType9

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Transfert de la source sous forme de texte.
1	Valeur	Transfert de la source sous forme de valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType9**. Le type de données est LONG.

Propriété OperatorMessageSourceType10

Transfert sous forme de - OperatorMessageSourceType10

Détermine dans quel format s'effectue le transfert de la source.

Les formats suivants sont disponibles :

Valeur	Description	Explication
0	Texte	Le contenu de la source est transféré comme texte.
1	Valeur	Le contenu de la source est transféré comme valeur.

L'attribut est dynamisable avec le nom **OperatorMessageSourceType10**. Le type de données est LONG.

Propriété OperationReport

Description

TRUE, lorsque la raison pour une commande doit être également listée. BOOLEAN Accès en lecture et en écriture.
Lors de la commande de l'objet dans Runtime, un dialogue apparaît dans lequel l'utilisateur peut entrer le motif de la commande sous forme de texte. La commande est envoyée vers le système d'alarme et archivée.

Voir aussi

Slider (Page 219)

Liste de textes (Page 206)

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Or - Ou

Propriété Orientation

Description

TRUE, lorsque le texte est représenté horizontalement dans l'objet. BOOLEAN Accès en écriture et en lecture.

Description du type d'objet "Connecteur"

Modifie l'orientation du connecteur. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Connecteur (Page 178)
Texte statique (Page 176)
Liste de textes (Page 206)
Case d'option (Page 215)
Case à cocher (Page 213)
Bouton (Page 210)
Champ d'E/S (Page 193)
Objet ScreenItem (Page 132)

Propriété OuterBevelStyle

Description

Détermine l'effet tridimensionnel pour la bordure extérieure de l'objet.

- 0: Aucune bordure.
- 1: La bordure est représentée en creux.
- 2: La bordure est représentée en relief.
- 3: La bordure est représentée en une couleur, sans effet tridimensionnel. La couleur de la bordure est définie par la propriété "BevelColorUp" .

Voir aussi

WinCC Slider Control (Page 275)
Objet ScreenItem (Page 132)

Propriété OuterBevelWidth

Description

Détermine la largeur de la bordure extérieure en Pixel.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété Outline

Description

TRUE, lorsqu'en plus du cadre tridimensionnel, le bouton est bordé de noir. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété OutputFormat

Description

Fournit la valeur en retour pour la représentation de la valeur de sortie et la définit. La représentation dépend du format de données.

Voir aussi

Champ d'E/S (Page 193)

Objet ScreenItem (Page 132)

Propriété OutputValue

Description

Détermine le paramétrage par défaut pour la valeur à afficher ou le restitue.
Dans Runtime, cette valeur est utilisée si au démarrage de la vue la variable en question n'est pas reliée ou n'est pas mise à jour.

Voir aussi

Liste de textes (Page 206)
Champ d'E/S (Page 193)
Objet ScreenItem (Page 132)

1.14.4.15 P

Pa - Pe

Propriété PageMode

Activer le défilement - PageMode

Détermine si le défilement est possible dans la liste d'archives longue durée. Vous pouvez ainsi afficher par page toutes les alarmes de l'archive cyclique dans la liste des archives longue durée La propriété "Alarmes par page" ou "PageModeMessageNumber" permet de définir le nombre d'alarmes affichées par page.

Les boutons pour le défilement dans la barre d'outils peuvent être utilisés si la fonction de défilement est activée.

Valeur	Explication
TRUE	Le défilement dans la liste d'archives longue durée est possible.
FALSE	Le défilement dans la liste d'archives longue durée n'est pas possible.

L'attribut est dynamisable avec le nom **PageMode**. Le type de données est BOOLEAN.

Propriété PageModeMessageNumber

Alarmes par page - PageModeMessageNumber

Détermine le nombre d'alarmes devant s'afficher par page lors du défilement dans la liste d'archives longue durée.

L'attribut est dynamisable avec le nom **PageModeMessageNumber**. Le type de données est LONG.

Propriété Parent

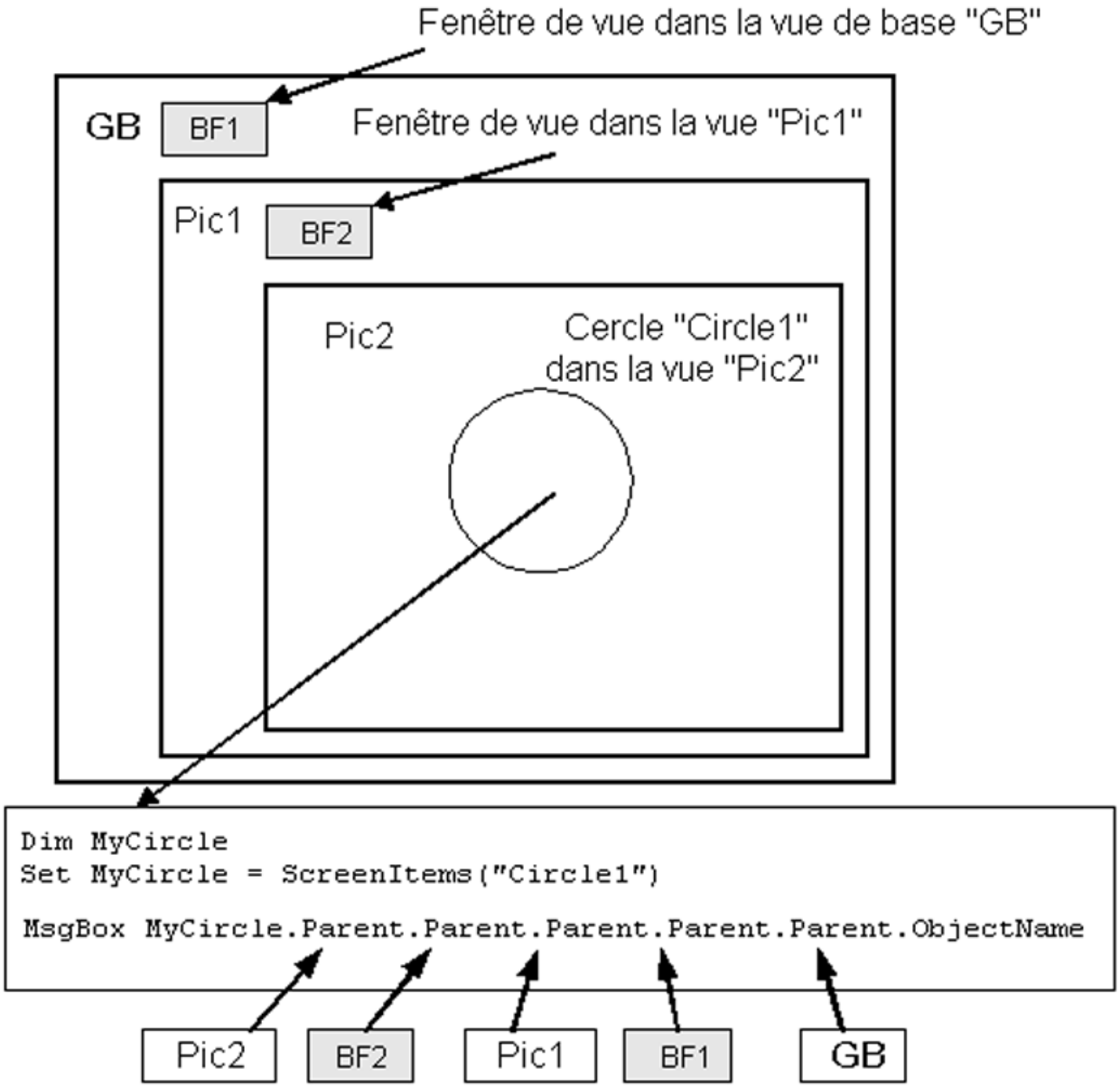
Description

Fournit une référence sur l'objet prioritaire.

L'accès aux objets se trouvant dans le modèle d'objets VBS se fait dans l'ordre hiérarchique. Screen et Screenitems permet de reculer dans la hiérarchie des vues. La propriété Parent permet d'avancer dans la hiérarchie.

Utilisation

Vous pouvez utiliser la propriété Parent aussi souvent que vous le voulez dans une hiérarchie d'objets. Vous apprenez schématiquement ci-dessous comment accéder à tous les éléments d'une hiérarchie :



La commande

```
MsgBox MyCircle.Parent.Objectname
```

renvoie le nom de "Vue2" qui se trouve dans la hiérarchie des objets à un niveau supérieur juste au dessus de l'objet d'origine ScreenItem "Circle1".

Si vous utilisez par exemple "Parent" trois fois, vous grimpez de trois niveaux dans la hiérarchie des objets :

```
MsgBox MyCircle.Parent.Parent.Parent.Objectname
```

fournit le nom de Vue1.

Explication :

- ScreenItem "Circle1" est à l'origine référencé
- "Circle1" se trouve dans "Vue2" (1er niveau)
- "Vue2" se trouve dans FenêtreVue2 "FV2" (2ème niveau)
- "FV2" se trouve dans "Vue 1" (3ème niveau)

Exemple

L'exemple suivant affiche le nom de l'objet Parent :

```
'VBS120
Dim objCircle
Set objCircle = HMIRuntime.Screens("ScreenWindow1").ScreenItems("Circle1")
MsgBox objCircle.Parent.ObjectName
```

Voir aussi

Fenêtre de vue (Page 188)

Objet Screen (Page 138)

Objets et listes (Page 114)

Propriété PasswordLevel

Description

Détermine l'autorisation pour la commande (p. ex. aucune saisie ou aucun déclenchement d'action) pour l'objet.

Voir aussi

Objet ScreenItem (Page 132)

Propriété Path

Description

Renvoie le chemin du projet actuel (sans le nom de fichier). Sur un client WinCC n'ayant pas son propre projet le chemin est renvoyé en format UNC, sinon c'est le chemin local.

STRING (accès en lecture uniquement)

Exemple

L'exemple suivant affiche le chemin de projet en tant que Trace :

```
'VBS161
HMIRuntime.Trace "Path: " & HMIRuntime.ActiveProject.Path & vbNewLine
```

Voir aussi

Objet Project (Page 131)

Propriété PercentageAxis

PercentageAxis

Cet attribut détermine si un axe à la mise à l'échelle en pourcentage est également affiché dans la fenêtre de courbes relative aux axes de valeurs.

Valeur	Explication
TRUE	Un axe à la mise à l'échelle en pourcentage est affiché.
FALSE	Aucun axe à la mise à l'échelle en pourcentage n'est affiché.

L'attribut est dynamisable avec le nom **PercentageAxis**. Le type de données est BOOLEAN.

Propriété PercentageAxisAlign

PercentageAxisAlign

Cet attribut détermine l'alignement de l'axe à la mise à l'échelle en pourcentage dans la fenêtre de courbes.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	à gauche	L'axe à la mise à l'échelle en pourcentage est aligné à gauche.
1	à droite	L'axe à la mise à l'échelle en pourcentage est aligné à droite.

L'attribut est dynamisable avec le nom **PercentageAxisAlign**. Le type de données est LONG.

Propriété PercentageAxisColor

PercentageAxisColor

Indique la couleur de l'axe à la mise à l'échelle en pourcentage. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **PercentageAxisColor**. Le type de données est LONG.

Propriété PersistentRT

Description

TRUE, lorsque des paramètres de fenêtre modifiés sont encore là après un changement de vue. L'indication sera ou non exploitée selon la valeur de la propriété "AllowPersistance".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété PersistentRTCS

Description

TRUE, lorsque des paramètres modifiés restent maintenus après un changement de vue et sont validés dans le système de configuration. L'indication sera ou non exploitée selon la valeur de la propriété "AllowPersistance". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété PersistentRTCSPermission

Description

Détermine l'autorisation de conduite qui est nécessaire pour pouvoir modifier les paramètres en terme de persistance. La valeur à entrer correspond au numéro de l'autorisation concernée dans User Administrator. L'indication sera ou non analysée en fonction de la valeur de la propriété "AllowPersistance" (n'est pas valable pour WinCC Alarm Control).

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété PersistentRTPermission

Description

Détermine l'autorisation de conduite qui est nécessaire pour pouvoir modifier en Runtime les paramètres en terme de persistance. La valeur à entrer correspond au numéro de l'autorisation concernée dans User Administrator. L'indication sera ou non analysée en fonction de la valeur de la propriété "AllowPersistance" (n'est pas valable pour WinCC Alarm Control).

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Pi

Propriété PicDeactReferenced

Description

TRUE, lorsque la vue assignée à l'état "Désactivé" est enregistrée dans l'objet RoundButton. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. Accès en lecture uniquement.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicDeactTransparent

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à l'état "Désactivé" doit être configurée sur "transparent", ou restitue la couleur. LONG Accès en lecture et en écriture. La couleur n'est configurée sur "transparent" que si la propriété "PicDeactUseTransColor" est positionnée sur la valeur TRUE.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicDeactUseTransColor

Description

TRUE, lorsque la couleur transparente déterminée avec la propriété "PicDeactTransparent" doit être utilisée pour l'état "Désactivé". BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicDownReferenced

Description

TRUE, lorsque la vue assignée à l'état "Marche" est enregistrée. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. Accès en lecture uniquement.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicDownTransparent

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à l'état "Marche" doit être configurée sur "transparent", ou restitue la couleur. LONG Accès en lecture et en écriture. La couleur n'est configurée sur "transparent" que si la propriété "PicDownUseTransColor" est positionnée sur la valeur TRUE.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicDownUseTransColor

Description

TRUE, lorsque la couleur transparente déterminée avec la propriété "PicDownTransparent" doit être utilisée pour l'état "Marche". BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicReferenced

Description

TRUE, lorsque la vue assignée est référencée et pas enregistrée dans l'objet. Accès en lecture uniquement.

Voir aussi

Objet graphique (Page 196)

Objet ScreenItem (Page 132)

Propriété PictAlignment

Description

Définit ou restitue l'orientation de la vue sur le bouton ou le bouton rond.

LONG Accès en écriture et en lecture.

Propriété PicTransColor

Description

Détermine quelle couleur de l'objet Bitmap assigné (.bmp, .dib) doit être configurée sur "transparent" ou restitue la couleur. LONG Accès en lecture et en écriture.
La couleur n'est configurée sur "transparent" que si la propriété "PicUseTransColor" est positionnée sur la valeur TRUE.

Voir aussi

Objet graphique (Page 196)

Objet ScreenItem (Page 132)

Propriété Picture

Description

Restitue aussi bien à l'horloge analogique qu'à l'horloge numérique le nom de vue d'arrière-plan pour l'arrière-plan rectangulaire. Accès en lecture uniquement.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété PictureBack

Description

Restitue le nom de vue pour l'arrière-plan de l'objet. Accès en lecture uniquement.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété PictureDeactivated

Description

Détermine la vue qui s'affiche à l'état "Désactivé" ou restitue le nom de la vue.
La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PictureDown

Description

Détermine la vue qui s'affiche à l'état "Activé" ou restitue le nom de la vue.
La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Voir aussi

Bouton (Page 210)

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PictureName

Description

Détermine la vue qui s'affiche dans l'objet graphique en Runtime ou restitue le nom de la vue.

La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Voir aussi

Objet graphique (Page 196)

Objet ScreenItem (Page 132)

Propriété PictureSelected

Description

Restitue le nom de la vue qui s'affiche à l'état "Marche". "AutoSize" commande l'adaptation de la taille de l'image au bouton. Accès en lecture uniquement.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété PictureSizeMode

PictureSizeMode

Définit le mode d'adaptation de la taille entre l'image et le contrôle.

Valeur	Désignation	Signification
0	Fit size to content	Le contrôle est adapté à la taille de l'image.
1	Fit content to size	L'image est adaptée au contrôle ou mise à l'échelle.

L'attribut est dynamisable avec le nom **PictureSizeMode**. Le type de données est LONG.

Propriété PictureThumb

Description

Restitue le nom de la vue d'arrière-plan pour le curseur. Accès en lecture uniquement.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété PictureUnselected

Description

Restitue le nom de la vue qui s'affiche à l'état "Arrêt". "AutoSize" commande l'adaptation de la taille de l'image au bouton. Accès en lecture uniquement.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété PictureUp

Description

Détermine la vue qui s'affiche à l'état "Arrêt" ou restitue le nom de la vue.
La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Voir aussi

Bouton rond (Page 217)

Bouton (Page 210)

Objet ScreenItem (Page 132)

Propriété PicUpReferenced

Description

TRUE, lorsque la vue assignée à l'état "Arrêt" est enregistrée dans l'objet. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. Accès en lecture uniquement.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicUpTransparent

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à l'état "Arrêt" doit être configurée sur "transparent", ou restitue la couleur. LONG Accès en lecture et en écriture. La couleur n'est configurée sur "transparent" que si la propriété "PicUpUseTransColor" est positionnée sur la valeur TRUE.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicUpUseTransColor

Description

TRUE, lorsque la couleur transparente déterminée avec la propriété "PicUpTransparent" doit être utilisée pour l'état "Arrêt". BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété PicUseTransColor

Description

TRUE, lorsque la couleur transparente déterminée avec la propriété "PicDeactTransparent" doit être utilisée pour l'état "Désactivé". BOOLEAN Accès en écriture et en lecture.

Voir aussi

Objet graphique (Page 196)

Objet ScreenItem (Page 132)

Pl - Pr

Propriété PlayEndless

PlayEndless

Définit si les films peuvent être lus en boucle sans fin dans le contrôle.

L'attribut est dynamisable avec le nom **PlayEndless**. Le type de données est BOOLEAN.

Propriété PointCount

Description

Détermine le nombre de points d'angle ou le restitue. Chaque point d'angle a des coordonnées de position et est identifié par un indice.

Voir aussi

Trait polygonal (Page 168)

Polygone (Page 166)

Objet ScreenItem (Page 132)

Propriété Position

Description

Détermine le préréglage pour la position du curseur.

Cette valeur est prise comme valeur initiale dans runtime.

Pour la commande de la valeur processus liée à cet attribut, il est impératif de lier également la valeur processus et l'événement "Position". Vous trouverez l'événement "Position" dans

l'onglet "Evénement", dans l'arborescence des rubriques sous SliderCtrl\Propertythemen \Control Propriétés\Valeur.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété Precisions

Description

WinCC Online Table Control

La propriété "Index" référence un couple de colonnes. "Precision" détermine le nombre de chiffres après la virgule qui doivent s'afficher dans cette colonne de valeurs. 16 chiffres peuvent au maximum s'afficher après la virgule.

WinCC Online Trend Control

Détermine le nombre de chiffres après la virgule que l'on utilise pour spécifier les valeurs graduées.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété PrecisionX

Description

Détermine le nombre de chiffres après la virgule pour spécifier les valeurs graduées pour l'axe X, ou le restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété PrecisionY

Description

Détermine le nombre de chiffres après la virgule pour spécifier les valeurs graduées pour l'axe Y, ou le restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété PredefinedAngles

Description

Détermine la représentation de la profondeur pour l'objet tridimensionnel ou la restitue. Plage des valeurs de 0 à 3.

0 = Cavalière

1 = Isométrique

2 = Axonométrique

3 = Librement définie

Voir aussi

Objet ScreenItem (Page 132)

Bargraphes tridimensionnels (Page 179)

Propriété Pressed

Description

TRUE, lorsque l'objet Button ou RoundButton est enfoncé. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

PrintBackgroundColor-propriété

Description

TRUE quand lors de l'impression du Contrôle la couleur d'arrière-plan déterminée est imprimée avec. BOOLEAN Accès en écriture et en lecture.

Propriété PrintJob

Description

Détermine ou lit quel layout d'impression est utilisé pour la sortie sur papier.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété PrintJobName

Travail d'impression vue actuelle - PrintJobName

Détermine le travail d'impression que déclenche la fonction d'impression du bouton "Imprimer" de la barre d'outils. Le travail d'impression conseillé est prédéfini pour le contrôle.

Le bouton de sélection permet d'afficher la boîte de dialogue "Sélectionner le travail d'impression" pour sélectionner le travail d'impression.

L'attribut est dynamisable avec le nom **PrintJobName**. Le type de données est STRING.

Propriété Process

Description

Définit le préréglage pour la valeur à afficher ou le restitue .
En Runtime, cette valeur est utilisée lorsqu'au moment de l'activation de la vue la variable correspondante n'est ni raccordée, ni mise à jour.

Voir aussi

Slider (Page 219)
Case d'option (Page 215)
Case à cocher (Page 213)
Bargraphe (Page 184)
Bargraphes tridimensionnels (Page 179)
Objet ScreenItem (Page 132)

Propriété ProcessValue

Description

Retourne un objet de type "ProcessValue".

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété ProjectPath

Description

Contient le chemin et le nom du projet correspondant.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)
Objet ScreenItem (Page 132)

Propriété ProviderClsid

Description

La propriété "Index" référence une courbe. "ProviderClsid" détermine si dans cette courbe une variable d'archive ou bien une variable soit interne, soit externe doit être s'afficher.

- {416A09D2-8B5A-11D2-8B81-006097A45D48}: La courbe est associée à une variable d'archive.
- {A3F69593-8AB0-11D2-A440-00A0C9DBB64E}: La courbe est associée à une variable interne ou une variable externe.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété ProviderType

Description

Détermine quel type de valeur doit être représenté dans une courbe référencée via "Index". En cas de modification du "ProviderType", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "ProviderType", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

0: La distribution des valeurs se fait via l'interface API.

-1: Représentation de valeurs en ligne ou de variables d'archives

-2: Représentation de valeurs dans une archive utilisateur

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

1.14.4.16 Q

Propriété QualityCode

Description

Représente une mesure de qualité pour la valeur de variable après la lecture. Le QualityCode est proposé comme valeur 16 Bit pour une exploitation autonome. Après l'écriture d'une variable, la valeur n'est plus valable.

SHORT (readonly)

Remarque

Vous trouverez un récapitulatif des QualityCodes possibles en consultant WinCC Information System sous le mot repère "Communication" > "Diagnostic" ou "Communication" > "Codes qualité".

Exemple

L'exemple suivant affiche la qualité de la valeur lue, lorsqu'aucune erreur n'est survenue lors de la lecture :

```
'VBS83
Dim objTag
Dim lngLastErr
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
lngLastErr = objTag.LastError
If 0 = lngLastErr Then
MsgBox objTag.QualityCode
    End If
```

Voir aussi

Propriété LastError (Page 432)

Propriété ErrorDescription (Page 386)

Objet Tag (Page 144)

1.14.4.17 R

Ra - Ri

Propriété Radius

Description

Détermine le rayon en Pixel ou le restitue.

Voir aussi

Segment de cercle (Page 162)

Arc de cercle (Page 160)

Cercle (Page 158)

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété RadiusHeight

Description

Détermine le rayon vertical en Pixel (0 à 5000) ou le restitue.

Voir aussi

Segment d'ellipse (Page 156)
Arc d'ellipse (Page 154)
Ellipse (Page 151)
Objet ScreenItem (Page 132)

Propriété RadiusWidth

Description

Détermine le rayon horizontal en Pixel (0 à 5000) ou le restitue.

Voir aussi

Segment d'ellipse (Page 156)
Arc d'ellipse (Page 154)
Ellipse (Page 151)
Objet ScreenItem (Page 132)

Propriété RangeMax

Description

Détermine la valeur absolue maximale pour l'affichage des valeurs.

Si la propriété "WithLabels" porte la valeur -1 (oui), cette valeur s'affiche sur l'échelle.

Voir aussi

WinCC Slider Control (Page 275)
Objet ScreenItem (Page 132)

Propriété RangeMin

Description

Détermine la valeur absolue minimale pour l'affichage des valeurs.

Si la propriété "WithLabels" porte la valeur -1 (oui), cette valeur s'affiche sur l'échelle.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété Rectangular

Description

Détermine le rapport de la hauteur à la largeur pour l'arrière-plan rectangulaire de la jauge, ou le restitue. BOOLEAN Accès en écriture et en lecture.

FALSE: La taille de la jauge peut être réglée avec la souris en faisant glisser les points de repère sous tous les rapports de hauteur/largeur que vous voulez.

TRUE: La taille de la jauge ne peut être réglée avec la souris qu'en faisant glisser les points de repère et uniquement sur ces points. Le rapport hauteur/largeur de l'arrière plan reste alors toujours 1:1.

Voir aussi

Objet ScreenItem (Page 132)

WinCC Gauge Control (Page 258)

Propriété ReferenceRotationLeft

Description

Détermine la coordonnée X du point de référence autour duquel l'objet doit pivoter en Runtime ou la restitue.
La valeur de la coordonnée X est relative à la largeur de l'objet. Spécifiez en pour cent la valeur en partant du bord gauche du rectangle entourant l'objet.

Voir aussi

Trait (Page 164)

Trait polygonal (Page 168)

Polygone (Page 166)

Objet ScreenItem (Page 132)

Propriété ReferenceRotationTop

Description

Détermine la coordonnée Y du point de référence autour duquel l'objet doit pivoter en Runtime ou la restitue.

La valeur de la coordonnée Y est relative à la hauteur de l'objet. Spécifiez en pour cent la valeur en partant du bord supérieur du rectangle entourant l'objet.

Voir aussi

Objet ScreenItem (Page 132)

Trait (Page 164)

Trait polygonal (Page 168)

Polygone (Page 166)

Propriété RelayCurves

Description

TRUE, lorsque les courbes doivent être échelonnées. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété Relevant

Description

TRUE, lorsque l'objet est pris en compte pour la constitution de la signalisation groupée. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété Replacement

Description

La propriété "Index" référence une courbe. Les valeurs dont la valeur initiale n'est pas connue après activation du mode Runtime ou pour lesquelles une valeur de substitution est utilisée ont un état non stable. "Replacement" détermine si ce genre de valeur doit être identifié par la couleur définie dans "ReplacementColor". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ReplacementColor

Description

La propriété "Index" référence une courbe. Les valeurs dont la valeur initiale n'est pas connue après activation du mode Runtime ou pour lesquelles une valeur de substitution est utilisée ont un état non stable. "ReplacementColor" détermine la couleur qui est utilisée pour identifier ces valeurs. La couleur doit être spécifiée sous la forme RVB. L'indication sera ou non exploitée selon la propriété "Replacement".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété RightComma

Description

Détermine le nombre de chiffres après la virgule (0 à 20) ou les restitue.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Ro - Ru

Propriété Rotation

Rotation (Rotation)

Définit la rotation antihoraire autour du centre de l'icône.

Les paramétrages suivants sont possibles :

Valeur	Signification
0	L'icône ne subit pas de rotation.
90	L'icône subit une rotation de 90 degrés.
180	L'icône subit une rotation de 180 degrés.
270	L'icône subit une rotation de 270 degrés.

L'attribut est dynamisable avec le nom **Rotation**. Le type de données est LONG.

Propriété RotationAngle

Description

Objets standard

Détermine en degrés l'angle de rotation ou le restitue.

L'objet est représenté en runtime (à partir de la position initiale configurée) pivoté autour du point de référence dans le sens des aiguilles d'une montre à hauteur de la valeur entrée. L'orientation modifiée de l'objet n'est visible qu'en runtime.

Les coordonnées du point de référence sont définies à l'aide des attributs "Référence de rotation X" et "Référence de rotation Y".

Raccord en T

Définit l'orientation d'un raccord en T en degrés ou la retourne.

L'attribut peut accepter quatre valeurs. Si vous entrez une autre valeur, celle-ci est automatiquement convertie en module 360 et arrondie à la valeur autorisée supérieure ou inférieure la plus proche.

L'orientation résulte d'une rotation du raccord en T autour de son centre dans le sens des aiguilles d'une montre à hauteur du nombre de degrés entrés.

0 La position par défaut du raccord en T représentée par la lettre "T"

90 La "jambe" du "T" est dirigée vers la gauche

180 La "jambe" du "T" est dirigée vers le haut
270 La "jambe" du "T" est dirigée vers la droite

Voir aussi

Trait (Page 164)
Trait polygonal (Page 168)
Polygone (Page 166)
Objet ScreenItem (Page 132)

Propriété RoundCornerHeight

Description

Détermine le rayon d'angle ou le restitue.
Tapez la valeur en pourcentage de la demi-hauteur de l'objet.

Voir aussi

Rectangle arrondi (Page 173)
Objet ScreenItem (Page 132)

Propriété RoundCornerWidth

Description

Détermine le rayon d'angle ou le restitue.
Tapez la valeur en pourcentage de la demi-largeur de l'objet.

Voir aussi

Objet ScreenItem (Page 132)

Propriété RowCellCount

RowCellCount

Indique le nombre de cellules de l'objet Row dans un Control table. Le nombre de cellules correspond au nombre de colonnes.

Propriété RowCellText

RowCellText

Restitue le contenu d'une cellule. La cellule est déterminée en fonction du numéro de colonne de l'objet Row. La numérotation va de "1" à "CellCount".

Propriété RowCount

RowCount

Indique le nombre de lignes de l'objet Row dans un Control table.

Propriété RowNumber

RowNumber

Indique le numéro de ligne de l'objet Row dans un Control table.

Propriété RowScrollbar

Barre de défilement des lignes - RowScrollbar

Détermine si les barres de défilement des lignes s'affichent ou non.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	non	Les barres de défilement des lignes ne s'affichent pas.
1	si besoin est	Les barres de défilement des lignes s'affichent lorsque l'encombrement du contrôle en sens horizontal est plus grand que la plage d'affichage disponible.
2	toujours	Les barres de défilement des lignes s'affichent toujours.

L'attribut est dynamisable avec le nom **RowScrollbar**. Le type de données est LONG.

Propriété RowTitleAlign

Alignement de l'étiquette de ligne - RowTitleAlign

Détermine comment les titres de lignes sont alignés.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	à gauche	Les titres de lignes sont alignés à gauche.
1	centré	Les titres de lignes sont centrés.
2	à droite	Les titres de lignes sont alignés à droite.

L'attribut est dynamisable avec le nom **RowTitleAlign**. Le type de données est LONG.

Propriété RowTitles

Affichage de l'étiquette de ligne - RowTitles

Détermine si les étiquettes des lignes sont affichées.

Valeur	Explication
TRUE	Les étiquettes des lignes sont affichées.
FALSE	Les étiquettes des lignes ne sont pas affichées.

L'attribut est dynamisable avec le nom **RowTitles**. Le type de données est BOOLEAN.

Propriété RTPersistence

Configuration en ligne lors du prochain changement de vue - RTPersistence

Détermine si les configurations en ligne du contrôle sont conservées après un changement de vue.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	rejeter	Les configurations en ligne actuelles sont rejetées au prochain changement de vue.
1	conserver	Les configurations en ligne actuelles sont conservées au prochain changement de vue.
2	remise à zéro	Toutes les configurations en ligne encore non effectuées sont perdues. La vue est définie dans le contenu disponible dans le système de configuration.

L'attribut est dynamisable avec le nom **RTPersistence**. Le type de données est LONG.

Propriété RTPersistencePasswordLevel

Autorisation de conduite pour la configuration en ligne - RTPersistencePasswordLevel

Indique l'autorisation pour la configuration en ligne. Vous pouvez modifier l'autorisation par le biais du bouton de sélection. Les autorisations sont configurées dans l'éditeur "User Administrator".

L'attribut est dynamisable avec le nom **RTPersistencePasswordLevel** . Le type de données est LONG.

Propriété RTPersistenceType

Configuration en ligne - RTPersistenceType

Détermine comment conserver des configurations en ligne de WinCC.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	ne pas conserver	Les configurations en ligne ne sont pas conservées. Elles sont perdues au prochain changement de vue.
1	Conserver pendant le mode Runtime	Les configurations en ligne sont conservées pendant le mode Runtime. Elle sont perdues en quittant le mode.
2	Conserver en permanence	Les configurations en ligne sont conservées en permanence. Elles sont également disponible après un redémarrage.

L'attribut n'est pas dynamisable.

Propriété RulerFont

Description

Cet attribut définit la police de la table des valeurs de variable qui s'affiche via la fonction de touche "Afficher valeur à cet endroit" / "Règle". Accès en écriture et en lecture.

Propriété RulerPrecisions

Description

Détermine le nombre de chiffres après la virgule pour l'affichage d'une valeur mesurée lorsqu'elle est relevée via la fonction "Afficher valeur à cet endroit".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété RulerPrecisionX

Description

Détermine le nombre de chiffres après la virgule qu'utilise la fonction "Afficher valeur à cet endroit" pour afficher la coordonnée X d'une valeur mesurée. L'indication sera ou non exploitée selon la valeur de l'attribut "TimeAxisX".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété RulerPrecisionY

Description

Détermine le nombre de chiffres après la virgule qu'utilise la fonction "Afficher valeur à cet endroit" pour afficher la coordonnée Y d'une valeur mesurée.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété RulerType

Fenêtre - RulerType

Détermine quelle fenêtre s'affiche en runtime. Selon le type de fenêtre, seuls certains blocs sont utilisables comme colonnes du RulerControl WinCC.

On dispose des types de fenêtre suivantes :

Valeur	Description	Explication
0	Fenêtre "Règle"	La fenêtre Règle indique les valeurs des coordonnées des courbes sur la règle ou les valeurs d'une ligne sélectionnée dans le tableau.
1	Fenêtre "Plage de statistiques"	La fenêtre Plage de statistiques indique les valeurs des seuils inférieur et supérieur des courbes entre deux règles ou de la plage sélectionnée dans le tableau.
2	Fenêtre "Statistiques"	La fenêtre Statistiques présente l'interprétation statistique des courbes entre deux règles ou des valeurs sélectionnées dans le tableau.

L'attribut est dynamisable avec le nom **RulerType**. Le type de données est LONG.

1.14.4.18 S

Sa - Sc

Propriété SameSize

Description

TRUE, lorsque les quatre boutons de l'objet GroupDisplay sont de même taille. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

SavedTrend-propriété

Description

Affiche le nom des dernières courbes enregistrées qui ont été exportées dans WinCC Online Trend Control avec le bouton "Enregistrer le rapport". Accès en lecture uniquement.

Propriété ScaleColor

Description

Détermine la couleur de l'échelle ou la restitue. LONG Accès en écriture et en lecture.
Il faut que la propriété "Scaling" soit positionnée sur "True" pour que la couleur s'affiche.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ScaleTicks

Description

Détermine le nombre de segments dans lequel est divisée l'échelle par le biais de grandes divisions :
0-100: L'objet est divisé en 100 segments au plus
= 0: Le nombre optimal de segments est déterminé automatiquement.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété Scaling

Description

TRUE, lorsqu'une échelle est également utilisée pour représenter les valeurs. BOOLEAN
Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ScalingType

Description Mise à l'échelle du bargraphe

Détermine le type d'échelle du bargraphe ou le restitue. Plage des valeurs de 0 à 6.

0 = linéaire

1 = logarithmique

2 = négative logarithmique

3 = automatique (linéaire)

4 = tangentielle

5 = carrée

6 = cubique

Il faut que la propriété "Scaling" soit positionnée sur TRUE pour que la couleur s'affiche.

Description Online Trend Control

Détermine ou restitue le type de mise à l'échelle de la courbe référencée via "Index". Plage des valeurs de 0 à 2.

0 = linéaire

1 = logarithmique

2 = négative logarithmique

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ScalingTypeX

Description

Détermine le type d'échelle de l'axe X pour une courbe référencée via "Index". L'indication sera ou non exploitée selon la valeur de l'attribut "TimeAxisX".

0: Linéaire

-1: Logarithmique Aucune valeur négative ne peut être affichée pour ce paramètre.

-2: Logarithmique inverse. Aucune valeur positive ne peut être affichée pour ce paramètre.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ScalingTypeY

Description

Détermine le type d'échelle de l'axe Y pour une courbe référencée via "Index".

0: Linéaire

-1: Logarithmique Aucune valeur négative ne peut être affichée pour ce paramètre.

-2: Logarithmique inverse. Aucune valeur positive ne peut être affichée pour ce paramètre.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ScreenName

Description

Détermine la vue qui s'affiche dans la fenêtre de vue en Runtime ou restitue le nom de la vue.

Remarque

Pour des raisons de compatibilité avec les versions ultérieures, il est toujours préférable d'écrire les noms de vue sans l'extension de fichier ".PDL" .

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété Screens

Description

Renvoie un objet de type "Screens".

Screens (read-only)

Exemple

L'exemple suivant accède à la vue "NewPDL1" :

```
'VBS84
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
```

Voir aussi

Objet Screens (Liste) (Page 141)

Objet Screen (Page 138)

Objet HMIRuntime (Page 125)

Propriété ScreenItems

Description

Renvoie un objet de type "ScreenItems".

ScreenItems (readonly)

Exemple

L'exemple suivant affiche le nom des objets contenus dans la vue "NewPDL1" :

```
'VBS85
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
Msgbox objScreen.ScreenItems.Count
```

Voir aussi

Objet ScreenItems (Liste) (Page 136)

Objet HMIRuntime (Page 125)

Propriété ScrollBars

Description

TRUE, lorsque l'objet a en Runtime des barres de défilement. Accès en lecture uniquement.

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété ScrollPositionX

Description

Définit le déplacement horizontal de la barre de défilement dans une fenêtre de vue ou restitue la valeur.

La vue est affichée dans la fenêtre de vue par le déplacement horizontal ou vertical des barres de défilement. Si vous voulez afficher une coupe de la vue dans laquelle les barres de défilement se trouvent sur les bords gauche et supérieur de la vue, utilisez les propriétés "OffsetLeft" et "OffsetTop" pour l'origine de cette coupe.

Voir aussi

Objet ScreenItem (Page 132)

Propriété OffsetTop (Page 488)

Propriété OffsetLeft (Page 487)

Fenêtre de vue (Page 188)

Propriété ScrollPositionY

Description

Définit le déplacement vertical de la barre de défilement dans une fenêtre de vue ou restitue la valeur.

La vue est affichée dans la fenêtre de vue par le déplacement horizontal ou vertical des barres de défilement. Si vous voulez afficher une coupe de la vue dans laquelle les barres de défilement se trouvent sur les bords gauche et supérieur de la vue, utilisez les propriétés "OffsetLeft" et "OffsetTop" pour l'origine de cette coupe.

Voir aussi

Propriété OffsetTop (Page 488)

Propriété OffsetLeft (Page 487)

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Se

Propriété SecondNeedleHeight

Description

Détermine la longueur de l'aiguille des secondes sur l'horloge analogique ou la restitue. La longueur est spécifiée en pour cent par rapport à la demi-longueur du côté le plus court du rectangle d'arrière-plan. Accès en écriture et en lecture.

Exemple:
Si le côté le plus court du rectangle d'arrière-plan était long de 100 pixels,
La longueur de l'aiguilles des secondes serait 80.
D'où une aiguille des minutes secondes de (100 pixels / 2) * 0,8 = 40 pixels.

Voir aussi

Objet ScreenItem (Page 132)

WinCC Digital Analog Clock (Page 251)

Propriété SecondNeedleWidth

Description

Détermine la largeur de l'aiguille des secondes sur l'horloge analogique ou la restitue. La largeur est spécifiée en pour cent par rapport à la longueur de l'aiguille des secondes multipliée par deux. Accès en écriture et en lecture.

Exemple:
La longueur de l'aiguille des secondes est 40 pixels.
La largeur de l'aiguille des secondes est 2.
D'où une aiguille des secondes large de 40 pixels * 2 * 0,02 = 2 pixels (arrondi).

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété SelBGColor

Description

Détermine la couleur d'arrière-plan pour l'inscription sélectionnée pour l'objet Liste des textes, ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété SelectArchiveName

SelectArchiveName

Ouvre la boîte de dialogue pour la sélection de l'archive utilisateur.

Les programmeurs peuvent utiliser l'attribut pour permettre par ex. à l'utilisateur de sélectionner une archive utilisateur via un bouton.

L'attribut est dynamisable avec le nom **SelectArchiveName**. Le type de données est BOOLEAN.

Propriété SelectedCellColor

Couleur d'arrière-plan Cellule marquée - SelectedCellColor

Indique la couleur d'arrière-plan de la cellule marquée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **SelectedCellColor**. Le type de données est LONG.

Propriété SelectedCellForeColor

Couleur de texte Cellule marquée - SelectedCellForeColor

Indique la couleur du texte de la cellule marquée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **SelectedCellForeColor**. Le type de données est LONG.

Propriété SelectedRowColor

Couleur d'arrière-plan Ligne marquée - SelectedRowColor

Indique la couleur d'arrière-plan de la ligne marquée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **SelectedRowColor**. Le type de données est LONG.

Propriété SelectedRowForeColor

Couleur de texte Ligne marquée - SelectedRowForeColor

Indique la couleur du texte de la ligne marquée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **SelectedRowForeColor**. Le type de données est LONG.

Propriété SelectedTitleColor

Arrière-plan Couleur de marquage - SelectedTitleColor

Indique la couleur d'arrière-plan du titre de tableau marqué. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif en Runtime que si l'option "Couleur de marquage" ou "UseSelectedTitleColor" est activée.

L'attribut est dynamisable avec le nom **SelectedTitleColor**. Le type de données est LONG.

Propriété SelectedTitleForeColor

Texte Couleur de marquage - SelectedTitleForeColor

Indique la couleur de texte du titre de tableau marqué. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif en Runtime que si l'option "Couleur de marquage" ou "UseSelectedTitleColor" est activée.

L'attribut est dynamisable avec le nom **SelectedTitleForeColor**. Le type de données est LONG.

SelectedTrend-propriété

Description

Cette propriété permet d'afficher une courbe au premier plan à l'aide de son nom. Accès en écriture et en lecture.

Propriété SelectionColoring

Couleurs de marquage pour - SelectionColoring

Détermine si les couleurs de marquage utilisées sont pour cellule ou pour ligne.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Aucun	Aucune couleur de marquage pour cellule et ligne.
1	Cellule	Couleur de marquage pour cellule.
2	Ligne	Couleur de marquage pour ligne.
3	Cellule et ligne	Couleurs de marquage pour cellule et ligne.

L'attribut est dynamisable avec le nom **SelectionColoring**. Le type de données est LONG.

Propriété SelectionMode

Description

Détermine si et comment une ligne d'alarme peut être sélectionnée.

- 0 - NoSelection: Empêche la sélection d'une alarme. Un acquittement agit toujours sur la plus ancienne des alarmes présentes.
- 1 - Cell: Permet la sélection de champs sur la ligne d'alarme. Un acquittement agit toujours sur l'alarme sélectionnée.
- 2 - Line: Permet la sélection d'une ligne d'alarme. Un acquittement agit toujours sur l'alarme sélectionnée.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété SelectionRect

Cadre de sélection - SelectionRect

Détermine si un cadre de sélection est utilisé pour les cellules ou les lignes de tableau marquées.

Les paramétrages suivants sont possibles :

Valeur	Description	Explication
0	Aucune	Aucun cadre de sélection n'est utilisé pour les cellules et les lignes marquées.
1	Cellule	Un cadre de sélection est utilisé pour la cellule marquée.
2	Ligne	Un cadre de sélection est utilisé pour la ligne marquée.

L'attribut est dynamisable avec le nom **SelectionRect**. Le type de données est LONG.

Propriété SelectionRectColor (avant WinCC V7)

Description

Détermine la couleur du rectangle de sélection dans la fenêtre d'alarmes lorsque le type de sélection est "1".

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété SelectionRectColor

Couleur Cadre de sélection - SelectionRectColor

Indique la couleur du trait du cadre de sélection. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **SelectionRectColor**. Le type de données est LONG.

Propriété SelectionRectWidth (avant WinCC V7)

Description

Détermine la largeur de ligne du rectangle de sélection dans la fenêtre d'alarmes lorsque le type de sélection est "1".

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété SelectionRectWidth

Largeur Cadre de sélection - SelectionRectWidth

Détermine la largeur du trait du cadre de sélection en pixels.

L'attribut est dynamisable avec le nom **SelectionRectWidth**. Le type de données est LONG.

Propriété SelectionType (avant WinCC V7)

Description

Détermine si l'alarme sélectionnée dans la fenêtre d'alarmes est optiquement mise en relief par un changement de couleur ou par un rectangle de sélection.

- 0 - Changement de couleur : l'alarme sélectionnée est optiquement mise en relief par un changement de couleur
- 1 - Rectangle de sélection : l'alarme sélectionnée est optiquement mise en relief par un rectangle de sélection

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété SelectionType

Lignes pouvant être marquées - SelectionType

Détermine combien de lignes peuvent être marquées. Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Aucun	Aucune ligne ne sera marquée.
1	Sélection individuelle	Une ligne peut être marquée.
2	Sélection multiple	Plusieurs lignes peuvent être marquées.

L'attribut est dynamisable avec le nom **SelectionType**. Le type de données est LONG.

Propriété SelIndex

Description

Définit et retourne l'index dont le texte assigné est affiché en surmarquage dans le champ de combinaison ou le champ de liste.

La valeur maximale est le nombre de lignes (NumberLines) de l'objet.

Propriété SelText

Description

Montre le texte défini avec l'attribut "Champ sélectionné" (SelIndex) qui est représenté en surmarquage dans le champ de combinaison ou le champ de liste.

Propriété SelTextColor

Description

Détermine la couleur du texte de l'inscription sélectionnée pour l'objet Liste de textes, ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété ServerData

Description

L'attribut ne peut être modifié que via le dialogue "Propriétés de WinCC Online Trend Control". Accès en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

ServerDataX

ServerDataX

Accède dans WinCC FunctionTrendControl à la connexion de données configurée pour l'axe X.

L'attribut est dynamisable avec le nom **ServerDataX**. Le type de données est LONG.

Exemple : modifier l'ID de début

L'attribut **ServerDataX** permet de modifier l'ID de début de l'axe X.

Il faut, pour ce faire, qu'une courbe existe, qu'une fenêtre de courbes ait été créée, que les axes X et Y soient configurés et qu'une connexion de données vers l'archive utilisateur soit établie.

Dans l'exemple suivant, vous établissez avec la méthode GetTrend une référence à l'objet dans une première étape et une référence à la courbe utilisée dans une deuxième étape. A la troisième étape, vous déterminez les paramètres de la connexion de données et, à la quatrième étape, l'ID de début reçoit la valeur 4. Le (3) correspond au type de liste "Archive utilisateur" pour la transmission de données. La modification des paramètres de connexion de données est exécutée à l'étape cinq :

```
Sub OnCklick(ByVal Item)
```

1. Etape :
```
Dim fx_ctrl
Set fx_ctrl  ScreenItems.Item("Control1")
```
2. Etape :
```
Dim fx_trend
Set fx_trend = fx_ctrl.Getrend("myTrend1")
```
3. Etape :
```
Dim vServerDataX, vServerDataY
vServerDataX = fx_trend.ServerDataX
vServerDataY = fx_trend.ServerDataY
```

4. Etape :
```
Dim startId
startId = CLng(4)
vServerDataX(3) = startId
vServerDataY(3) = startId
```
5. Etape :
```
fx_trend.ServerDataX = ServerDataX
fx_trend.ServerDataY = ServerDataY
```

```
End Sub
```

ServerDataY

ServerDataY

Accède dans WinCC FunctionTrendControl à la liaison de données configurée pour l'axe Y.

L'attribut est dynamisable avec le nom **ServerDataY**. Le type de données est LONG.

Exemple : modifier l'ID de début

L'attribut **ServerDataY** permet de modifier l'ID de début de l'axe Y.

Il faut, pour ce faire, qu'une courbe existe, qu'une fenêtre de courbes ait été créée, que les axes X et Y soient configurés et qu'une connexion de données vers l'archive utilisateur soit établie.

Dans l'exemple suivant, vous établissez avec la méthode GetTrend une référence à l'objet dans une première étape et une référence à la courbe utilisée dans une deuxième étape. A la troisième étape, vous déterminez les paramètres de la connexion de données et, à la quatrième étape, l'ID de début reçoit la valeur 4. Le (3) correspond au type de liste "Archive utilisateur" pour la transmission de données. La modification des paramètres de connexion de données est exécutée à l'étape cinq :

```
Sub OnCklick(ByVal Item)
```

1. Etape :
```
Dim fx_ctrlSet fx_ctrl  ScreenItems.Item("Control1")
```
2. Etape :
```
Dim fx_trendSet fx_trend = fx_ctrl.Getrend("myTrend1")
```
3. Etape :
```
Dim vServerDataX, vServerDataYvServerDataX =
fx_trend.ServerDataXvServerDataY = fx_trend.ServerDataY
```
4. Etape :
```
Dim startIdstartId = CLng(4)vServerDataX(3) =
startIdvServerDataY(3) = startId
```
5. Etape :
```
fx_trend.ServerDataX = ServerDataXfx_trend.ServerDataY =
ServerDataY
```

```
End Sub
```

Propriété ServerNames

Sélection de serveur - ServerNames

Dans un système réparti, cet attribut détermine à partir de quels serveurs la fenêtre d'alarmes obtient les données à représenter.

L'attribut est dynamisable avec le nom **ServerNames**. Le type de données est STRING.

Propriété ServerNames (avant WinCC V7)

Description

Détermine les serveurs d'un système réparti desquels la fenêtre d'alarmes doit recevoir des données. L'indication s'effectue sous la forme : NameServer1;NameServer2;NameServer3. Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété ServerPrefix

Description

Détermine sur quel serveur se trouve la vue qui s'affiche en Runtime dans la fenêtre de vue, ou restitue le nom de serveur.
Spécifiez le nom de serveur suivi de deux deux-points: "<Nom de serveur>::". Le système ne vérifie pas si le serveur existe effectivement.

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Sh - Sk

Propriété ShareSpaceWithSourceControl

ShareSpaceWithSourceControl

Cet attribut détermine si la taille du contrôle source est adaptée dans la fenêtre de vue afin que le RulerControl WinCC soit également affiché dans une petite fenêtre de vue.

Valeur	Explication
TRUE	Le contrôle source est adapté dans la fenêtre de vue.
FALSE	Le contrôle source n'est pas adapté dans la fenêtre de vue.

L'attribut est dynamisable avec le nom **ShareSpaceWithSourceControl**. Le type de données est BOOLEAN.

Propriété ShowBar

Description

TRUE, lorsque le bargraphe s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété ShowDanger

Description

Contrôle l'affichage de la "zone de danger" sur l'échelle. BOOLEAN Accès en écriture et en lecture.

TRUE: La zone est identifiée par la couleur déterminée via "DangerColor".

FALSE: L'identification colorée de la zone est désactivée.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété ShowDecimalPoint

Description

TRUE, lorsque la graduation est marquée par des chiffres décimaux (virgule décimale et chiffre décimal).

FALSE, lorsque la graduation est marquée par des chiffres entiers.

BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété ShowNormal

Description

Contrôle l'affichage de la "zone normale" sur l'échelle. BOOLEAN Accès en écriture et en lecture.

TRUE: La zone est identifiée par la couleur déterminée via "NormalColor".

FALSE: L'identification colorée de la zone est désactivée.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété ShowPeak

Description

Détermine l'affichage d'un index mobile pour afficher la valeur maximale. BOOLEAN Accès en écriture et en lecture.

TRUE: L'index mobile est affiché.

FALSE: L'index mobile n'est plus affiché.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété ShowPosition

Description

TRUE, lorsque la position du curseur s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété ShowRuler

Afficher règle - ShowRuler

Détermine si lors de l'ouverture d'une vue la règle permettant d'interroger les points des coordonnées s'affiche.

Valeur	Explication
TRUE	La règle permettant d'interroger les points des coordonnées s'affiche.
FALSE	La règle permettant d'interroger les points des coordonnées ne s'affiche pas.

L'attribut est dynamisable avec le nom **ShowRuler**. Le type de données est BOOLEAN.

Propriété ShowRulerImmediately

Description

TRUE, lors de l'ouverture d'une vue, lorsque la règle permettant de définir les valeurs des coordonnées doit s'afficher. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ShowRulerInAxis

ShowRulerInAxis

Détermine si les règles sont également affichées dans les axes des temps.

Valeur	Explication
TRUE	Les règles sont également affichées dans les axes des temps.
FALSE	Les règles ne sont pas affichées dans les axes des temps.

L'attribut est dynamisable avec le nom **ShowRulerInAxis**. Le type de données est BOOLEAN.

Propriété ShowScrollbars

Barre de défilement - ShowScrollbars

Détermine si les barres de défilement sont affichées.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	non	Les barres de défilement ne sont pas affichées.
1	si besoin est	Les barres de défilement s'affichent lorsque l'encombrement du contrôle est plus grand que la plage d'affichage actuelle.
2	toujours	Les barres de défilement sont toujours affichées.

L'attribut est dynamisable avec le nom **ShowScrollbars**. Le type de données est LONG.

Propriété ShowSlider

ShowSlider

Définit si une barre d'avancement s'affiche dans le contrôle.

L'attribut est dynamisable avec le nom **ShowSlider**. Le type de données est BOOLEAN.

Propriété ShowSortButton

Utiliser la touche de tri - ShowSortButton

Détermine si la touche de tri au dessus de la barre de défilement verticale est affichée. La touche de tri vous permet de trier par un clic de souris la colonne sélectionnée dans l'ordre de

tri configuré. Si aucune barre de défilement verticale n'existe dans le tableau, la touche de tri n'est pas affichée.

Valeur	Explication
TRUE	Vous pouvez trier la colonne sélectionnée via la touche de tri.
FALSE	Le touche de tri n'est pas affichée.

L'attribut est dynamisable avec le nom **ShowSortButton** . Le type de données est BOOLEAN.

Propriété ShowSortIcon

Affichage du symbole de tri - ShowSortIcon

Détermine si le symbole de tri est affiché.

Valeur	Explication
TRUE	Le symbole de tri est affiché.
FALSE	Le symbole de tri n'est pas affiché.

L'attribut est dynamisable avec le nom **ShowSortIcon**. Le type de données est BOOLEAN.

Propriété ShowSortIndex

Affichage de l'index de tri - ShowSortIndex

Spécifie si un index de tri s'affiche.

Valeur	Explication
TRUE	Un index de tri est affiché.
FALSE	Un index de tri n'est pas affiché.

L'attribut est dynamisable avec le nom **ShowSortIndex**. Le type de données est BOOLEAN.

ShowSpanNames-propriété

Description

TRUE quand dans la colonne "Valeur" de la fenêtre de règle dans le Contrôle de courbe, un nom de domaine est affiché en supplément à la valeur mesurée et à l'affichage de statut "i" et "u". BOOLEAN Accès en écriture et en lecture.

Propriété ShowStatisticRuler

ShowStatisticRuler

Cet attribut détermine si les règles s'affichent pour la plage de statistiques à l'ouverture de la vue.

Valeur	Explication
TRUE	Les règles s'affichent pour la plage de statistiques.
FALSE	Les règles ne s'affichent pas pour la plage de statistiques.

L'attribut est dynamisable avec le nom **ShowStatisticRuler**. Le type de données est BOOLEAN.

Propriété ShowThumb

Description

TRUE, lorsque le curseur s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété ShowTitle

Titre de fenêtre - ShowTitle

Détermine si le titre de fenêtre du contrôle est ou non représenté.

Valeur	Désignation	Explication
0	non	Pas de titre de fenêtre.
1	Normal	Le titre de fenêtre se compose d'un symbole WinCC et d'un texte. Le texte est saisi dans le champ "´Texte".
2	étroit	Le titre de fenêtre ne se compose que d'un texte. Le texte est saisi dans le champ "´Texte".

L'attribut est dynamisable avec le nom **ShowTitle**. Le type de données est LONG.

Propriété ShowToolbar

ShowToolbar

Définit si une barre d'outils s'affiche dans le contrôle.

L'attribut est dynamisable avec le nom **ShowToolbar**. Le type de données est BOOLEAN.

Propriété ShowTrendIcon

ShowTrendIcon

Détermine si un symbole est affiché sous les axes de valeurs. Le symbole indique quelle courbe est affichée au premier plan.

L'attribut est dynamisable avec le nom **ShowTrendIcon**. Le type de données est BOOLEAN.

Propriété ShowValuesExponentialX

Description

TRUE, lorsque la coordonnée X définie par la fonction "Afficher valeur à cet endroit", coordonnée d'une valeur mesurée pour une courbe référencée via "Index", est représentée sous une forme de notation exponentielle. L'indication sera ou non exploitée selon la valeur de la propriété "TimeAxisX". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ShowValuesExponentialY

Description

TRUE, lorsque la coordonnée Y définie par la fonction "Afficher valeur à cet endroit", coordonnée d'une valeur mesurée pour une courbe référencée via "Index", est représentée sous une forme de notation exponentielle. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ShowWarning

Description

Contrôle l'affichage de la "zone d'avertissement" sur l'échelle. BOOLEAN Accès en écriture et en lecture.

TRUE: La zone est identifiée par la couleur déterminée via "WarningColor".

FALSE: L'identification colorée de la zone est désactivée.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété SignificantMask

Description

Nécessaire en Runtime pour représenter la classe d'alarmes active ayant la priorité la plus élevée.
La valeur de la propriété SignificantMask représente une valeur de sortie interne au système qui ne requiert aucune configuration spécifique de la part de l'utilisateur. La mise à jour s'effectue en Runtime en cliquant sur l'objet.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété Sizeable

Redimensionnable - Sizeable

Détermine si la taille du contrôle est modifiable en Runtime.

Valeur	Explication
TRUE	La taille du contrôle est modifiable en Runtime.
FALSE	La taille du contrôle n'est pas modifiable en Runtime.

L'attribut est dynamisable avec le nom **Sizeable**. Le type de données est BOOLEAN.

Propriété SkinName

Style - SkinName

Ce champ de sélection permet de définir le style du contrôle.

Les paramétrages suivants sont disponibles :

Valeur	Désignation	Explication
	Paramètre de projet	Le style correspond aux paramètres de projet dans l'Explorateur WinCC.
0	Simple	Style WinCC "classique"
1	Standard	Nouveau style WinCC V7
	Basic Process Control	Le style est réservé pour l'utilisation interne pour le Basic Process Control.

L'attribut est dynamisable avec le nom **SkinName**. Le type de données est STRING.

Sm - Sq

Propriété SmallChange

Description

Détermine de combien de pas le régulateur doit être déplacé avec un clic de souris, ou restitue cette valeur.

Voir aussi

Slider (Page 219)

Objet ScreenItem (Page 132)

Propriété SmartTag

Description

Retourne un objet de type "SmartTag".

Voir aussi

Objet SmartTags (Page 143)

SortOrder-propriété

Description

Définit l'ordre de tri des blocs d'alarme dans la fenêtre d'alarme.

Propriété SortSequence

Ordre de tri par clic de souris - SortSequence

Détermine comment modifier l'ordre de tri par un clic de souris.

Vous disposez des ordre de tri suivants :

Valeur	Description	Explication
0	croissant/ décroissant/aucun	Vous pouvez permuter par un clic de souris entre le tri croissant, décroissant et aucun tri.
1	croissant/ décroissant	Vous pouvez permuter par un clic de souris entre le tri croissant et le tri décroissant.

L'attribut est dynamisable avec le nom **SortSequence**. Le type de données est LONG.

Propriété SourceBeginTime

Description

Détermine pour les variables en ligne et les variables d'archive ("ProviderType" = -1) le moment de départ de la plage de temps à représenter dans la fenêtre de courbes pour une courbe référencée via "Index". En cas de modification du "SourceBeginTime", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceBeginTime", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceControl

Source - SourceControl

Définit à quel contrôle le RulerControl WinCC est relié.

L'attribut est dynamisable avec le nom **SourceControl**. Le type de données est STRING.

Propriété SouceControlType

Type - SouceControlType

Indique le type de contrôle auquel est lié RulerControl WinCC dans le champ "Source".

Valeur	Désignation	Explication
0	Aucun	Le RulerControl n'est relié à aucune source.
1	OnlineTrend Control	Le RulerControl est relié à un OnlineTrendControl.
2	OnlineTable Control	Le RulerControl est relié à un OnlineTableControl.
3	FunctionTrend Control	Le RulerControl est relié à un FunctionTrendControl.

L'attribut est dynamisable avec le nom **SourceControlType**. Le type de données est LONG.

Propriété SourceEndTime

Description

Détermine pour les variables en ligne et les variables d'archive ("ProviderType" = -1) la fin de la plage de temps à représenter dans la fenêtre de courbes pour une courbe référencée via "Index". En cas de modification du "SourceEndTime", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceEndTime", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceNumberOfUAValues

Description

Détermine pour des valeurs des archives utilisateur ("ProviderType" = -2) pour une courbe référencée via "Index", le nombre de valeurs qui doivent être chargées depuis l'archive utilisateur. En cas de modification des "SourceNumberOfUAValues", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceNumberOfUAValues", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceNumberOfValues

Description

La propriété "Index" référence une courbe. Pour les variables en ligne et les variables d'archive ("ProviderType" = -1), "SourceNumberOfValues" détermine le nombre de valeurs qui doivent être représentées dans la fenêtre des courbes. L'indication sera ou non exploitée selon la valeur de la propriété "SourceTimeRange".

En cas de modification du "SourceNumberOfValues", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier

"SourceNumberOfValues", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceTagNameX

Description

La propriété "Index" référence une courbe. Pour les variables en ligne et les variables d'archive ("ProviderType" = -1), "SourceTagNameX" détermine les variables qui doivent être représentées le long de l'axe X. En cas de modification du "SourceTagNameX", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceTagNameX", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceTagNameY

Description

La propriété "Index" référence une courbe. Pour les variables en ligne et les variables d'archive ("ProviderType" = -1), "SourceTagNameY" détermine les variables qui doivent être représentées le long de l'axe Y. En cas de modification du "SourceTagNameY", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceTagNameY", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceTagProviderDataX

Description

Ne peut être modifié que via le dialogue "Propriétés de WinCC Function Trend Control".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceTagProviderDataY

Description

Ne peut être modifié que via le dialogue "Propriétés de WinCC Function Trend Control".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceTimeRange

Description

La propriété "Index" référence une courbe. Pour les variables en ligne et les variables d'archive ("ProviderType" = -1), "SourceTimeRange" détermine comment est définie la plage de temps à représenter dans la fenêtre des courbes. En cas de modification du "SourceTimeRange", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceTimeRange", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

0: La plage de temps à représenter est déterminée par le moment de départ (SourceBeginTime) et le nombre de couples de valeur (SourceNumberOfValues).

-1: La plage de temps à représenter est déterminée par le moment de départ (SourceBeginTime) et la fin (SourceEndTime)

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceUAArchive

Description

La propriété "Index" référence une courbe. Pour les valeurs des archives utilisateur ("ProviderType" = -2), "SourceUAArchive" détermine l'archive utilisateur depuis laquelle les valeurs doivent être chargées. En cas de modification du "SourceUAArchive", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des

données. Avant de modifier "SourceUAArchive", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceUAArchiveStartID

Description

La propriété "Index" référence une courbe. Pour les valeurs des archives utilisateur ("ProviderType" = -2), "SourceUAArchiveStartID" détermine l'enregistrement à partir duquel les valeurs de l'archive utilisateur doivent être chargées. En cas de modification du "SourceUAArchiveStartID", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceUAArchiveStartID", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceUAColumnX

Description

La propriété "Index" référence une courbe. Pour les valeurs des archives utilisateur ("ProviderType" = -2), "SourceUAColumnX" détermine la colonne dans l'archive utilisateur depuis laquelle doivent être chargées les valeurs pour l'axe X. En cas de modification du "SourceUAColumnX", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceUAColumnX", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SourceUAColumnY

Description

La propriété "Index" référence une courbe. Pour les valeurs des archives utilisateur ("ProviderType" = -2), "SourceUAColumnY" détermine la colonne dans l'archive utilisateur depuis laquelle doivent être chargées les valeurs pour l'axe Y. En cas de modification du "SourceUAColumnY", il se peut que des combinaisons non admises se créent avec d'autres attributs pour l'intégration des données. Avant de modifier "SourceUAColumnY", il est pour cette raison impératif d'empêcher la validation immédiate de la modification avec "FreezeProviderConnections".

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété SquareExtent

Description

TRUE, lorsque la taille de l'horloge peut être réglée avec la souris en faisant glisser les points de repère sous tous les rapports de hauteur/largeur que vous voulez. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

St - Sy

Propriété StartAngle

Description

Détermine le début de l'objet ou le restitue. L'indication se fait dans le sens horaire, en degrés, en commençant à 12:00 h.

Voir aussi

Segment de cercle (Page 162)

Arc de cercle (Page 160)

Segment d'ellipse (Page 156)

Arc d'ellipse (Page 154)

Objet ScreenItem (Page 132)

Propriété State

Description

Édite le statut d'une alarme (propriété State)

Le tableau suivant montre les statuts possibles d'une alarme :

State	Statut de l'alarme
1	Arrivée
2	Partie
5	Arrivée et commentaire
6	Partie et commentaire

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété Statusbar

Description

TRUE, lorsqu'une ligne d'état s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété StatusbarBackColor

Couleur d'arrière-plan - StatusbarBackColor

Indique la couleur d'arrière-plan pour la barre d'état. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Afin que le paramétrage soit effectif, vous devez activer l'option "Afficher" ou "StatusbarUseBackColor".

L'attribut est dynamisable avec le nom StatusbarBackColor. Le type de données est LONG.

Propriété StatusbarElementAdd

Nouveau - StatusbarElementAdd

Détermine un nouvel élément de barre d'état défini par l'utilisateur. Vous pouvez modifier le nom attribué par WinCC dans le champ "Nom d'objet".

L'attribut est dynamisable avec le nom StatusbarElementAdd . Le type de données est STRING.

Propriété StatusbarElementAutoSize

Automatique - StatusbarElementAutoSize

Détermine si la largeur de l'élément de barre d'état sélectionné est automatiquement paramétrée.

Valeur	Explication
TRUE	La largeur de l'élément sélectionné est automatiquement paramétrée.
FALSE	La largeur de l'élément sélectionné n'est pas automatiquement paramétrée.

L'attribut est dynamisable avec le nom StatusbarElementAutoSize. Le type de données est BOOLEAN.

Propriété StatusbarElementCount

StatusbarElementCount

Cet attribut indique le nombre des éléments configurables de la barre d'état.

L'attribut est dynamisable avec le nom StatusbarElementCount. Le type de données est LONG.

Propriété StatusbarElementIconId

StatusbarElementIconId

Affectation définie du numéro d'identification et symbole d'un élément de la barre d'état

L'attribut est dynamisable avec le nom **StatusbarElementIconId** pour les éléments personnalisés de la barre d'état. Le type de données est LONG.

Propriété StatusbarElementID

ID objets - StatusbarElementID

Numéro d'identification univoque pour l'élément sélectionné de la barre d'état. Le numéro d'identification est attribué par WinCC et ne peut être modifié.

L'attribut est dynamisable avec le nom **StatusbarElementID**. Le type de données est LONG.

Propriété StatusbarElementIndex

StatusbarElementIndex

Cet attribut fait référence à un élément de la barre d'état. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un élément précis de la barre d'état.

Les valeurs admissibles pour "StatusbarElementIndex" et "StatusbarElementCount" se situent entre 0 et -1. L'attribut "StatusbarElementCount" indique le nombre des éléments configurables de la barre d'état.

L'attribut "StatusbarElementIndex" est dynamisable via l'attribut **StatusbarElementIndex**. Le type de données est LONG.

Propriété StatusbarElementName

Nom d'objet - StatusbarElementName

Affiche le nom d'objet pour l'élément sélectionné de la barre d'état. Vous pouvez modifier le nom d'objet des éléments personnalisés de la barre d'état.

L'attribut "StatusbarElementName" pour les éléments personnalisés est dynamisable via l'attribut **StatusbarElementRename**. Le type de données est STRING.

Propriété StatusbarElementRemove

Supprimer - StatusbarElementRemove

Supprime l'élément de barre d'état sélectionné. Vous ne pouvez supprimer de la liste que des éléments personnalisés de la barre d'état.

L'attribut est dynamisable avec le nom **StatusbarElementRemove**. Le type de données est STRING.

Propriété StatusbarElementRename

StatusbarElementRename

Cet attribut modifie le nom de l'élément configurable de la barre d'état qui est référencé via l'attribut "StatusbarElementIndex".

L'attribut est dynamisable avec le nom **StatusbarElementRename** pour les éléments personnalisés. "StatusbarElementRename" permet également de dynamiser l'attribut "StatusbarElementName". Le type de données est STRING.

Propriété StatusbarElementRepos

Haut/bas - StatusbarElementRepos

Modifie l'ordre des fonctions de touche. "Haut" et "Bas" déplacent vers le haut ou vers le bas de la liste l'élément sélectionné de la barre d'état. Ce qui a pour conséquence que l'élément dans la barre d'état du contrôle est positionné plus en avant ou plus en arrière.

L'attribut est dynamisable avec le nom **StatusbarElementRepos** . Le type de données est LONG.

Propriété StatusbarElementText

StatusbarElementText

Cet attribut définit le texte qui est affiché pour l'élément dans la barre d'état. Vous pouvez modifier l'attribut "StatusbarElementText" pour les éléments personnalisés.

L'attribut est dynamisable avec le nom **StatusbarElementText** pour les éléments personnalisés. Le type de données est STRING.

Propriété StatusbarElementTooltipText

StatusbarElementTooltipText

Cet attribut définit le texte pour l'info-bulle de l'élément personnalisé de la barre d'état.

L'attribut est dynamisable avec le nom **StatusbarElementTooltipText**. Le type de données est STRING.

Propriété StatusbarElementVisible

Eléments de la barre d'état - StatusbarElementVisible

Activez dans la liste les éléments de la barre d'état que vous souhaitez afficher en Runtime.

Cliquez sur une entrée dans la liste afin d'adapter les propriétés ou pour modifier la position dans la barre d'état du contrôle avec les touches "Haut" et "Bas".

Valeur	Explication
TRUE	L'élément de la barre d'état s'affiche.
FALSE	L'élément de la barre d'état ne s'affiche pas.

L'attribut est dynamisable avec le nom **StatusbarElementVisible**. Le type de données est BOOLEAN.

Propriété StatusbarElementUserDefined

StatusbarElementUserDefined

Cet attribut affiche si l'élément de la barre d'état a été ajouté par l'auteur de la configuration en tant que nouvel élément personnalisé.

Valeur	Explication
TRUE	L'élément de la barre d'état est personnalisé.
FALSE	L'élément de la barre d'état est spécifié par le système.

L'attribut est dynamisable avec le nom **StatusbarElementUserDefined**. Le type de données est BOOLEAN.

Propriété StatusbarElementWidth

Largeur en pixels - StatusbarElementWidth

Indique en pixels la largeur de l'élément sélectionné de la barre d'état. Si l'option "Automatique" n'est pas activée, vous pouvez définir la largeur.

L'attribut est dynamisable avec le nom **StatusbarElementWidth**. Le type de données est LONG.

Propriété StatusbarFontColor

StatusbarFontColor

Spécifie la couleur de police de caractères des textes dans la barre d'état.

L'attribut est dynamisable avec le nom **StatusbarFontColor**. Le type de données est LONG.

Propriété StatusbarPanes

Description

Détermine les éléments à représenter sur la ligne d'état. Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété StatusbarShowTooltips

Info-bulle - StatusbarShowTooltips

Détermine si les infos-bulles relatives aux éléments de la barre d'état s'affichent en Runtime.

Valeur	Explication
TRUE	Les infos-bulles s'affichent.
FALSE	Les infos-bulles ne s'affichent pas.

L'attribut est dynamisable avec le nom StatusbarShowTooltips. Le type de données est BOOLEAN.

L'attribut permettant de déterminer le texte d'info-bulle est "StatusbarElementTooltipText".

Propriété StatusbarText

StatusbarText

Texte défini dans la barre d'état

L'attribut est dynamisable avec le nom StatusbarText. Le type de données est STRING.

Propriété StatusbarUseBackColor

Afficher Couleur d'arrière-plan - StatusbarUseBackColor

Spécifie si la couleur d'arrière-plan de la barre d'état s'affiche.

Valeur	Explication
TRUE	La couleur d'arrière-plan de la barre d'état s'affiche.
FALSE	La couleur d'arrière-plan de la barre d'état ne s'affiche pas.

L'attribut est dynamisable avec le nom StatusbarUseBackColor. Le type de données est BOOLEAN.

Propriété StatusbarVisible

Afficher barre d'état - StatusbarVisible

Spécifie si la barre d'état du contrôle s'affiche.

Valeur	Explication
TRUE	La barre d'état s'affiche.
FALSE	La barre d'état ne s'affiche pas.

L'attribut est dynamisable avec le nom **StatusbarVisible** . Le type de données est BOOLEAN.

Propriété StepSeconds

StepSeconds

Définit l'intervalle de temps de saut avant ou arrière pour les films.

L'attribut est dynamisable avec le nom **StepSeconds**. Le type de données est LONG.

Propriété Stretch

Description

Détermine, en cas de modification de la taille de symbole, si le rapport hauteur/largeur reste le même ou peut être modifié. BOOLEAN Accès en écriture et en lecture.

- FALSE: En cas de modification de la taille de symbole, le rapport hauteur/largeur du symbole reste le même.
- TRUE: En cas de modification de la taille de symbole, il est possible de modifier simultanément le rapport hauteur/largeur du symbole.

Voir aussi

HMI Symbol Library (Page 246)

Objet ScreenItem (Page 132)

Propriété SymbolAppearance

Mode de premier plan (SymbolAppearance)

Définit l'aspect de l'icône.

Les paramétrages suivants sont possibles :

Valeur	Description	Signification
0	Original	L'aspect de l'icône correspond à la représentation multicolore dans la sélection de l'onglet "Icônes".
1	Ombré	Des lignes en noir sont conservées en tant que lignes de contour. Les éléments de l'icône qui sont de couleur différente sont représentés en tant que degrés de luminosité de la couleur de premier plan actuelle.
2	Opaque	Des lignes en noir sont conservées en tant que lignes de contour. La teinte de la couleur de premier plan actuelle est attribuée à tous les éléments de l'icône qui sont de couleur différente.
3	Contour	Des lignes en noir sont conservées en tant que lignes de contour. La teinte de l'arrière-plan est attribuée à tous les éléments de l'icône qui sont de couleur différente.

L'attribut est dynamisable avec le nom **SymbolAppearance**. Le type de données est LONG.

1.14.4.19 T

Ta -Tic

Propriété TableColor

Arrière-plan Couleur de ligne 1 - TableColor

Indique la couleur d'arrière-plan pour les lignes. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TableColor**. Le type de données est LONG.

Propriété TableColor2

Arrière-plan Couleur de ligne 2 - TableColor2

Indique la couleur d'arrière-plan de la "Couleur de ligne 2". Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif en Runtime que si l'option "Couleur de ligne 2" ou "UseTableColor2" est activée. Les couleurs d'arrière-plan de la "Couleur de ligne 1" et de la "Couleur de ligne 2" sont alors utilisées en alternance.

L'attribut est dynamisable avec le nom **TableColor2**. Le type de données est LONG.

TableFocusOnButtonCommand-propriété

Description

Détermine si en Runtime le focus sera paramétré sur le tableau du contrôle en cliquant sur un bouton dans un Script.

Propriété TableForeColor

Texte Couleur de ligne 1 - TableForeColor

Indique la couleur de texte des lignes. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TableForeColor**. Le type de données est LONG.

Propriété TableForeColor2

Texte Couleur de ligne 2 - TableForeColor2

Indique la couleur de ligne de la "Couleur de ligne 2". Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif en Runtime que si l'option "Couleur de ligne 2" ou "UseTableColor2" est activée. Les couleurs de ligne de la "Couleur de ligne 1" et de la "Couleur de ligne 2" sont alors utilisées en alternance.

L'attribut est dynamisable avec le nom **TableForeColor2**. Le type de données est LONG.

Propriété TagName

Description

La propriété "Index" référence une courbe. "TagName" détermine la variable qui est associée à cette courbe. L'indication se fait sous la forme "Nom d'archive\de variable" pour représenter les variables d'une archive de valeurs de processus ou "Nom de variable" pour représenter une variable interne ou externe qui n'est pas enregistrée dans une archive.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété TagPrefix

Description

Détermine le préfixe de variable précédant toutes les variables qui se trouvent dans l'objet de fenêtre de vue, ou le restitue. De cette manière, une vue incorporée dans une fenêtre de vue conserve l'accès à ses variables pendant qu'une autre accède à d'autres variables.

La modification d'un TagPrefix n'est appliquée que lors du chargement d'image suivant. En cas de changement de vue, cette action est automatique, sinon, il convient d'assigner une nouvelle fois le nom de la vue.

TagPrefix peut être défini au choix, mais il doit correspondre au nom des variables structurées.

Remarque

La propriété TagPrefix n'est pas disponible pour les contrôles.

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété Tags

Description

Renvoie un objet de type "Tags".

Tags (read-only)

Exemple

L'exemple suivant accède à la variable "Tag1" :

```
'VBS86
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
```

Voir aussi

Objet Tags (Liste) (Page 147)

Objet HMIRuntime (Page 125)

Propriété TagProviderClsid

Description

La propriété Index référence une courbe. "TagProviderClsid" détermine si une variable en ligne ou une valeur archivée doivent être représentées dans cette courbe. L'indication n'est exploitée que pour les variables en ligne et les variables d'archives ("ProviderType" = -1).

{A3F69593-8AB0-11D2-A440-00A0C9DBB64E}: Variables en ligne.

{416A09D2-8B5A-11D2-8B81-006097A45D48}: Les valeurs sont lues dans une archive de valeurs processus ou dans une archive utilisateur.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété Template

Description

Restitue le modèle pour l'affichage du contenu de fenêtre de l'objet "Fenêtre d'application". Accès en lecture uniquement.

Selon la valeur de la propriété, les modèles suivants sont possibles :

Contenu de la fenêtre = Global Script

"GSC-Diagnostic"
La fenêtre d'application provient des applications de Global Script. S'y affichent les résultats du système de diagnostic.

"GSC-Runtime"
La fenêtre d'application provient des applications de Global Script. S'y affichent les résultats d'analyse du comportement en exploitation.

Contenu de la fenêtre = Print Jobs

"All Jobs" :
La fenêtre d'application provient du système de journalisation. Les journaux disponibles sont représentés sous forme de liste.

"All Jobs - Context Menu" :
La fenêtre d'application provient du système de journalisation. Les journaux disponibles sont représentés sous forme de liste. Le menu contextuel permet de sélectionner les options d'impression, la représentation d'un aperçu avant impression ainsi que l'impression d'un journal.

"Job Detail View" :
La fenêtre d'application provient du système de journalisation. Les journaux disponibles sont représentés dans un menu de sélection. Pour le journal sélectionné, affichage des informations de détail.

"Selected Jobs - Context Menu" :
La fenêtre d'application provient du système de journalisation. Les journaux disponibles sont représentés sous forme de liste. Cette liste ne contient que les protocoles pour lesquels vous avez activé l'options "Sélection pour liste de travaux d'impression" dans le dialogue "Propriétés de travail d'impression". Le menu contextuel permet de sélectionner les options d'impression, la représentation d'un aperçu avant impression ainsi que l'impression d'un journal.

Voir aussi

Objet ScreenItem (Page 132)

Fenêtre d'application (Page 183)

Propriété Text

Description

Détermine le texte pour un objet ou le restitue.

Voir aussi

Case d'option (Page 215)

Case à cocher (Page 213)

Bouton (Page 210)

Texte statique (Page 176)

Objet ScreenItem (Page 132)

Propriété ThumbBackColor

Description

Détermine la couleur du curseur.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété TicColor

Description

Détermine la couleur de la graduation. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété TicFont

Description

Commande la représentation du marquage de la graduation. Accès en lecture uniquement.

Les propriétés suivantes sont paramétrables :

- Police de caractères
- Style de police
- Taille de police
- Effet "barré"
- Effet "souligné"

Voir aussi

WinCC Gauge Control (Page 258)
Objet ScreenItem (Page 132)

Propriété TicOffset

Description

Détermine le diamètre du cercle imaginaire sur lequel se trouve la graduation. Cette valeur se réfère à la plus petite valeur des propriétés géométriques Largeur et Hauteur.

Les traits principaux de la graduation se trouvent alors sur ce cercle, leur extrémité étant dirigée vers l'extérieur.

Plage des valeurs de 0 à 1.

0: La graduation se trouve au centre du disque.

1: Le diamètre du cercle imaginaire pour la graduation est la valeur la plus petite des propriétés géométriques Largeur et Hauteur.

Voir aussi

WinCC Gauge Control (Page 258)
Objet ScreenItem (Page 132)

Propriété TicTextColor

Description

Détermine la couleur du marquage de la graduation.

Voir aussi

WinCC Gauge Control (Page 258)
Objet ScreenItem (Page 132)

Propriété TicTextOffset

Description

Détermine le diamètre du cercle imaginaire sur lequel se trouve le marquage de la graduation. Cette valeur se réfère à la plus petite valeur des propriétés géométriques Largeur et Hauteur.

Plage des valeurs de 0 à 1.

0: Le marquage se trouve au centre du disque.

1: Le diamètre du cercle imaginaire pour le marquage est la valeur la plus petite des propriétés géométriques Largeur et Hauteur. Une partie du marquage peut ainsi se trouver à l'extérieur des limites de l'objet pour être invisible.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété TicWidth

Description

Détermine la longueur des traits principaux de la graduation. La valeur se réfère à la moitié de la plus petite valeur des propriétés géométriques Largeur et Hauteur.

La longueur des graduations fines est 0,5*Largeur de graduation.

Plage des valeurs de 0 jusqu'à l'intervalle.

0: Aucune graduation n'existe. La répartition de l'échelle en zones n'est pas non plus visible.

Intervalles: La graduation s'étend du centre du disque jusqu'à la valeur qui est déterminée par l'intervalle.

Voir aussi

Objet ScreenItem (Page 132)

WinCC Gauge Control (Page 258)

Propriété Ticks

Description

TRUE, lorsque le cadran s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété TicksColor

Description

Détermine la couleur du marquage des heures sur le cadran de l'horloge analogique, ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Digital Analog Clock (Page 251)

Objet ScreenItem (Page 132)

Propriété TickStyle

Description

Cet attribut vous permet de définir la présentation de l'échelle. Plage des valeurs : 0 à 3.

La graduation automatique peut éventuellement placer deux graduations de l'échelle directement l'une à côté de l'autre (graduation semblant plus épaisse). Cet effet peut être corrigé en raccourcissant ou en rallongeant légèrement l'objet Slider.

Vous pouvez également entièrement supprimer la représentation de la graduation ("WithAxes").

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

TimeAxis - TimeBase

Propriété TimeAxis

Description

Détermine dans la fenêtre des courbes si un axe de temps commun doit être utilisé pour toutes les courbes.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)
Objet ScreenItem (Page 132)

Propriété TimeAxisActualize

Rafraîchissement - TimeAxisActualize

Définit si les valeurs de l'axe de temps sélectionné sont mises à jour.

Valeur	Explication
TRUE	La fenêtre de courbes affectée à l'axe de temps est mise à jour.
FALSE	La fenêtre de courbes affectée à l'axe de temps n'est pas mise à jour. Ce paramétrage est judicieux lorsqu'une courbe archivée doit être comparée à une courbe actuelle.

L'attribut est dynamisable avec le nom **TimeAxisActualize**. Le type de données est BOOLEAN.

Propriété TimeAxisAdd

Nouveau - TimeAxisAdd

Crée un nouvel axe de temps.

L'attribut est dynamisable avec le nom **TimeAxisAdd**. Le type de données est STRING.

Propriété TimeAxisAlign

Alignement - TimeAxisAlign

Spécifie la façon dont s'aligne l'axe de temps sélectionné.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	en bas	L'axe de temps sélectionné s'affiche en dessous de la courbe.
1	en haut	L'axe de temps sélectionné s'affiche au dessus de la courbe.

L'attribut est dynamisable avec le nom **TimeAxisAlign**. Le type de données est LONG.

Propriété TimeAxisBeginTime

Heure de début - TimeAxisBeginTime

Détermine l'heure de début de la plage de temps pour l'axe de temps sélectionné.

L'attribut est dynamisable avec le nom **TimeAxisBeginTime**. Le type de données est Date.

Si vous dynamisez l'intervalle de temps, utilisez le format "aaaa-mm-jj hh:mm:ss".

Propriété TimeAxisColor

Couleur Axe de temps - TimeAxisColor

Indique la couleur de l'axe de temps. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage est effectif si l'option "dans la couleur de la courbe" n'est pas activée ou si "TimeAxisInTrendColor" "FALSE".

L'attribut est dynamisable avec le nom **TimeAxisColor**. Le type de données est LONG.

Propriété TimeAxisCount

TimeAxisCount

Cet attribut indique le nombre d'axes des temps configurés.

L'attribut est dynamisable avec le nom **TimeAxisCount**. Le type de données est LONG.

Propriété TimeAxisDateFormat

Format de la date - TimeAxisDateFormat

Détermine quel format de la date sera utilisé pour représenter l'axe de temps sélectionné.

Les formats de date suivants sont disponibles :

Valeur	Explication
Automatique	Le format de la date est automatiquement défini.
dd.MM.yy	Jour.Mois.Année, par exemple 24.12.07.
dd.MM.yyyy	Jour.Mois.Année, par exemple 24.12.2007.
dd/MM/yy	Jour/Mois/Année, par exemple 24/12/07.
dd/MM/yyyy	Jour/Mois/Année, par exemple 24/12/2007.

L'attribut est dynamisable avec le nom **TimeAxisDateFormat**. Le type de données est STRING.

Propriété TimeAxisEndTime

Heure de fin- TimeAxisEndTime

Détermine l'heure de fin de la plage de temps pour l'axe de temps sélectionné.

L'attribut est dynamisable avec le nom **TimeAxisEndTime**. Le type de données est Date.

Si vous dynamisez l'intervalle de temps, utilisez le format "aaaa-mm-jj hh:mm:ss".

Propriété TimeAxisFormat

Description

Détermine le format des données spécifiées le long de l'axe des temps.

- 0: Les spécifications sont données sous la forme hh:mm
- -1: Les spécifications sont données sous la forme hh:mm:ss
- -2: Les spécifications sont données sous la forme hh:mm:ss.ms
- -3: Les spécifications sont données sous la forme hh:mm (heures entières)
- -4: Les spécifications sont données sous la forme hh:mm:ss (minutes entières)
- -5: Les spécifications sont données sous la forme hh:mm:ss.ms (secondes entières)

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété TimeAxisIndex

TimeAxisIndex

Cet attribut fait référence à un axe de temps configuré. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un axe de temps précis.

Les valeurs admissibles pour "TimeAxisIndex" et "TimeAxisCount" se situent entre 0 et -1. L'attribut "TimeAxisCount" indique le nombre d'axes des temps configurés.

L'attribut "TimeAxisIndex" est dynamisable via l'attribut **TimeAxisRepos**. Le type de données est LONG.

Propriété TimeAxisInTrendColor

Dans la couleur de la courbe - TimeAxisInTrendColor

Détermine si l'axe de temps sélectionné s'affiche dans la couleur de la courbe. Si plusieurs courbes s'affichent dans le fenêtre de courbes, c'est la couleur de la première courbe qui est utilisée. Vous déterminez l'ordre des courbes sur l'onglet "Courbes".

Valeur	Explication
TRUE	L'axe de temps sélectionné s'affiche dans la couleur de la courbe. Les paramétrages dans le champ "Couleur" ou "TimeAxisColor" restent sans effet.
FALSE	L'axe de temps sélectionné s'affiche dans la couleur paramétrée dans le champ "Couleur" ou "TimeAxisColor".

L'attribut est dynamisable avec le nom **TimeAxisInTrendColor**. Le type de données est BOOLEAN.

Propriété TimeAxisLabel

Etiquette - TimeAxisLabel

Détermine le texte accompagnant l'axe de temps sélectionné.

L'attribut est dynamisable avec le nom **TimeAxisLabel**. Le type de données est STRING.

Propriété TimeAxisMeasurePoints

Nombre de points de mesure - TimeAxisMeasurePoints

Détermine le nombre de points de mesure affichés pour l'axe de temps sélectionné.

L'attribut est dynamisable avec le nom **TimeAxisMeasurePoints**. Le type de données est LONG.

Propriété TimeAxisName

Nom d'objet - TimeAxisName

Fixe le nom de l'axe de temps sélectionné.

L'attribut "TimeAxisName" est dynamisable via l'attribut **TimeAxisRename**. Le type de données est STRING.

Propriété TimeAxisRangeType

Paramétrage Plage de temps - TimeAxisRangeType

Détermine le paramétrage pour la plage de temps utilisée pour l'axe de temps sélectionné.

Valeur	Description	Explication
0	Plage de temps	Pour l'axe de temps, déterminez l'heure de début et la plage de temps.
1	Heure de début à heure de fin	Pour l'axe de temps, déterminez l'heure de début et l'heure de fin.
2	Nombre de points de mesure	Pour l'axe de temps, déterminez l'heure de début et le nombre des points de mesure.

L'attribut est dynamisable avec le nom **TimeAxisRangeType**. Le type de données est LONG.

Propriété TimeAxisRemove

Supprimer - TimeAxisRemove

Supprime l'axe de temps sélectionné dans la liste.

L'attribut est dynamisable avec le nom **TimeAxisRemove**. Le type de données est STRING.

Propriété TimeAxisRename

TimeAxisRename

Cet attribut modifie le nom de l'axe de temps qui est référencé via l'attribut "TimeAxisIndex".

L'attribut est dynamisable avec le nom **TimeAxisRename**. "TimeAxisRename" permet également de dynamiser l'attribut "TimeAxisName". Le type de données est STRING.

Propriété TimeAxisRepos

Haut/bas - TimeAxisRepos

Modifie l'ordre des axes de temps. "Haut" et "Bas" déplacent l'axe de temps sélectionné dans la liste vers le haut ou le bas.

L'ordre dans la liste définit en Runtime la position de l'axe de temps dans la fenêtre de courbes. Si l'alignement est identique et si l'axe de temps se trouve plus haut dans la liste, il est représenté sur une position plus éloignée par rapport à la courbe.

L'attribut est dynamisable avec le nom **TimeAxisRepos**. Le type de données est LONG.

Propriété TimeAxisShowDate

Affichage de la date - TimeAxisShowDate

Détermine si l'axe de temps sélectionné s'affiche avec la date et l'heure.

Valeur	Explication
TRUE	La date et l'heure s'affichent. Le format de la date est déterminé dans le champ "Format de la date".
FALSE	La date ne s'affiche pas. C'est uniquement l'heure qui s'affiche.

L'attribut est dynamisable avec le nom **TimeAxisShowDate**. Le type de données est BOOLEAN.

Propriété TimeAxisTimeFormat

Format de l'heure - TimeAxisTimeFormat

Détermine quel format de l'heure sera utilisé pour représenter l'axe de temps sélectionné.

Les formats d'heure suivants sont disponibles :

Valeur	Explication
Automatique	Le format de l'heure est automatiquement défini.
hh:mm:ss.ms	Heures:Minutes:Secondes.Millisecondes, par exemple 15:35:44.240.
hh:mm:ss tt	Heures:Minutes:Secondes AM/PM, par exemple 03:35:44 PM.
hh:mm:ss.ms tt	Heures:Minutes:Secondes.Millisecondes AM/PM, par exemple 03:35:44.240 PM.

L'attribut est dynamisable avec le nom **TimeAxisTimeFormat**. Le type de données est STRING.

Propriété TimeAxisTimeRangeBase

Plage de temps - TimeAxisTimeRangeBase

Détermine l'unité de temps pour définir la plage de temps.

Les unités de temps suivantes sont disponibles :

Valeur	Description
500	500 ms
1000	1 seconde
60000	1 minute
3600000	1 heure
86400000	1 jour

L'attribut est dynamisable avec le nom **TimeAxisTimeRangeBase**. Le type de données est LONG.

Propriété TimeAxisTimeRangeFactor

Plage de temps - TimeAxisTimeRangeFactor

Détermine le facteur pour définir la plage de temps. Seuls les facteurs entiers sont admis.

L'attribut est dynamisable avec le nom **TimeAxisTimeRangeFactor**. Le type de données est SHORT.

Propriété TimeAxisTrendWindow

Fenêtre de courbe - TimeAxisTrendWindow

Détermine dans quelle fenêtre de courbes l'axe de temps sélectionné sera utilisé. Vous définissez les fenêtres de courbes disponibles sur l'onglet "Fenêtre de courbes" ou via "TrendWindowAdd".

L'attribut est dynamisable avec le nom **TimeAxisTrendWindow**. Le type de données est STRING.

Propriété TImeAxisVisible

Axe de temps - TimeAxisVisible

Les axes des temps que vous avez créés sont énumérés dans la liste. Cliquez sur un axe de temps dans la liste pour adapter les propriétés et pour affecter une fenêtre de courbes à l'axe de temps.

Activez dans la liste les axes des temps que vous souhaitez afficher dans les fenêtres de courbes.

Spécifie si l'axe de temps sélectionné s'affiche.

Valeur	Explication
TRUE	L'axe de temps est affiché.
FALSE	L'axe de temps n'est pas affiché.

L'attribut est dynamisable avec le nom **TimeAxisVisible**. Le type de données est BOOLEAN.

Propriété TimeAxisX

Description

TRUE, lorsque dans la fenêtre des courbes un axe X commun est utilisé pour toutes les courbes. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété TimeBase

Base de temps - TimeBase

Ce champ de sélection permet de définir la base de temps valable pour les indications de temps dans le contrôle.

Valeur	Désignation
0	Fuseau horaire local
1	Temps universel coordonné (TUC)
2	Paramètre de projet

L'attribut est dynamisable avec le nom **TimeBase**. Le type de données est LONG.

TimeColumn

Propriété TimeColumnActualize

TimeColumnActualize

Définit si les valeurs de la colonne sélectionnée sont mises à jour.

Valeur	Explication
TRUE	La colonne heure est mise à jour.
FALSE	La colonne heure n'est pas mise à jour. Ce paramétrage est judicieux lorsqu'un tableau doit être comparé à un autre tableau.

L'attribut est dynamisable avec le nom **TimeColumnActualize**. Le type de données est BOOLEAN.

Propriété TimeColumnAdd

Nouveau - TimeColumnAdd

Crée une nouvelle colonne heure.

L'attribut est dynamisable avec le nom **TimeColumnAdd**. Le type de données est STRING.

Propriété TimeColumnAlign

Alignement - TimeColumnAlign

Spécifie la façon dont s'aligne la colonne heure sélectionnée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	à gauche	La colonne heure sélectionnée est affichée à gauche.
1	centré	La colonne heure sélectionnée est centrée.
2	à droite	La colonne heure sélectionnée est affichée à droite.

L'attribut est dynamisable avec le nom **TimeColumnAlign**. Le type de données est LONG.

Propriété TimeColumnAlignment

Description

La propriété "Index" référence un couple de colonnes. "TimeColumnAlignment" détermine l'alignement des colonnes de temps pour ce couple de colonnes.

- 0: Les valeurs temporelles sont inscrites alignées à gauche.
- 1: Les valeurs temporelles sont inscrites centrées.
- 2: Les valeurs temporelles sont inscrites alignées à droite.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété TimeColumnBackColor

Couleur d'arrière-plan - TimeColumnBackColor

Indique la couleur d'arrière-plan de la colonne heure sélectionnée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage est effectif :

- Si l'option "dans les couleurs de la colonne de valeurs" n'est pas activée ou si "TimeColumnUseValueColumnColors" "FALSE".
- Si l'option "Couleur d'arrière-plan" ou si "UseColumnBackColor" "TRUE dans la plage "Utiliser la couleur de colonne" de l'onglet "Général" est activée.

L'attribut est dynamisable avec le nom **TimeColumnBackColor**. Le type de données est LONG.

Propriété TimeColumnBeginTime

Heure de début - TimeColumnBeginTime

Détermine l'heure de début de la plage de temps pour la colonne heure sélectionnée.

L'attribut est dynamisable avec le nom **TimeColumnBeginTime**. Le type de données est Date.

Si vous dynamisez l'intervalle de temps, utilisez le format "aaaa-mm-jj hh:mm:ss".

Propriété TimeColumnCaption

Désignation - TimeColumnCaption

Détermine la désignation de la colonne heure.

L'attribut est dynamisable avec le nom **TimeColumnCaption**. Le type de données est STRING.

Propriété TimeColumnCount

TimeColumnCount

Cet attribut indique le nombre des colonnes heure configurées.

L'attribut est dynamisable avec le nom **TimeColumnCount**. Le type de données est LONG.

Propriété TimeColumnDateFormat

Format de la date - TimeColumnDateFormat

Détermine quel format de la date sera utilisé pour représenter la colonne heure sélectionnée.

Les formats de date suivants sont disponibles :

Valeur	Explication
Automatique	Le format de la date est automatiquement défini.
dd.MM.yy	Jour.Mois.Année, par exemple 24.12.07.
dd.MM.yyyy	Jour.Mois.Année, par exemple 24.12.2007.
dd/MM/yy	Jour/Mois/Année, par exemple 24/12/07.
dd/MM/yyyy	Jour/Mois/Année, par exemple 24/12/2007.

L'attribut est dynamisable avec le nom **TimeColumnDateFormat**. Le type de données est STRING.

Propriété TimeColumnEndTime

Heure de fin- TimeColumnEndTime

Détermine l'heure de fin de la plage de temps pour la colonne heure sélectionnée.

L'attribut est dynamisable avec le nom **TimeColumnEndTime**. Le type de données est Date.

Si vous dynamisez l'intervalle de temps, utilisez le format "aaaa-mm-jj hh:mm:ss".

Propriété TimeColumnForeColor

Couleur de police - TimeColumnForeColor

Indique la couleur de texte de la colonne heure sélectionnée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage est effectif :

- Si l'option "dans les couleurs de la colonne de valeurs" n'est pas activée ou si "TimeColumnUseValueColumnColors" "FALSE".
- Si l'option "Couleur de texte" dans la plage "Utiliser la couleur de colonne" de l'onglet "Général" est activée ou si "UseColumnForeColor" "TRUE".

L'attribut est dynamisable avec le nom **TimeColumnForeColor**. Le type de données est LONG.

Propriété TimeColumnHideText

TimeColumnHideText

Détermine si le contenu de la colonne heure doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le contenu ne s'affiche pas sous forme de texte.
FALSE	Le contenu s'affiche sous forme de texte.

L'attribut est dynamisable avec le nom **TimeColumnHideText**. Le type de données est BOOLEAN.

Propriété TimeColumnHideTitleText

TimeColumnHideTitleText

Détermine si le titre de la colonne heure doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le titre ne s'affiche pas sous forme de texte.
FALSE	Le titre s'affiche sous forme de texte.

L'attribut est dynamisable avec le nom **TimeColumnHideTitleText**. Le type de données est BOOLEAN.

Propriété TimeColumnIndex

TimeColumnIndex

Cet attribut fait référence à une colonne heure configurée. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à une colonne heure précise.

Les valeurs admissibles pour "TimeColumnIndex" et "TimeColumnCount" se situent entre 0 et -1. L'attribut "TimeColumnCount" indique le nombre de colonnes heure configurées.

L'attribut "TimeColumnIndex" est dynamisable via l'attribut **TimeColumnRepos**. Le type de données est LONG.

Propriété TimeColumnLength

Longueur en caractères - TimeColumnLength

Détermine la largeur de la colonne heure sélectionnée.

L'attribut est dynamisable avec le nom **TimeColumnLength**. Le type de données est LONG.

Propriété TimeColumnMeasurePoints

Nombre de points de mesure - TimeColumnMeasurePoints

Détermine le nombre de points de mesure s'affichant dans la colonne heure sélectionnée.

L'attribut est dynamisable avec le nom **TimeColumnMeasurePoints**. Le type de données est LONG.

Propriété TimeColumnName

Nom d'objet - TimeColumnName

Fixe le nom de la colonne heure sélectionnée.

L'attribut "TimeColumnName" est dynamisable via l'attribut **TimeColumnRename**. Le type de données est STRING.

Propriété TimeColumnRangeType

Paramétrage Plage de temps - TimeColumnRangeType

Détermine la plage de temps utilisée pour la colonne heure sélectionnée.

Valeur	Description	Explication
0	Plage de temps	Pour la colonne heure, déterminez l'heure de début et la plage de temps.
1	Heure de début à heure de fin	Pour la colonne heure, déterminez l'heure de début et l'heure de fin..
2	Nombre de points de mesure	Pour la colonne heure, déterminez l'heure de début et le nombre des points de mesure.

L'attribut est dynamisable avec le nom **TimeColumnRangeType**. Le type de données est LONG.

Propriété TimeColumnRemove

Supprimer - TimeColumnRemove

Supprime la colonne heure sélectionnée dans la liste.

L'attribut est dynamisable avec le nom **TimeColumnRemove**. Le type de données est STRING.

Propriété TimeColumnRename

TimeColumnRename

Cet attribut modifie le nom de la colonne heure qui est référencée via l'attribut "TimeColumnIndex".

L'attribut est dynamisable avec le nom **TimeColumnRename**. "TimeColumnRename" permet également de dynamiser l'attribut "TimeColumnName". Le type de données est STRING.

Propriété TimeColumnRepos

Haut/bas - TimeColumnRepos

Modifie l'ordre des colonnes heure avec les colonnes de valeurs correspondantes. "Haut" et "Bas" déplacent la colonne heure sélectionnée dans la liste vers le haut ou le bas. Ce qui a pour conséquence que la colonne heure avec les colonnes de valeurs correspondantes dans le tableau est positionnée plus en avant ou plus en arrière.

L'attribut est dynamisable avec le nom **TimeColumnRepos**. Le type de données est LONG.

Propriété TimeColumnShowDate

Affichage de la date - TimeColumnShowDate

Détermine si la colonne heure sélectionnée s'affiche avec la date et l'heure.

Valeur	Explication
TRUE	La date et l'heure s'affichent. Le format de la date est déterminé dans le champ "Format de la date" ou via "TimeColumnDateFormat".
FALSE	La date ne s'affiche pas. C'est uniquement l'heure qui s'affiche.

L'attribut est dynamisable avec le nom **TimeColumnShowDate**. Le type de données est BOOLEAN.

Propriété TimeColumnShowIcon

TimeColumnShowIcon

Détermine si le contenu de la colonne heure doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le contenu s'affiche sous forme de symbole.
FALSE	Le contenu ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **TimeColumnShowIcon**. Le type de données est BOOLEAN.

Propriété TimeColumnShowTitleIcon

TimeColumnShowTitleIcon

Détermine si le titre de la colonne heure doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le titre s'affiche sous forme de symbole.
FALSE	Le titre ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **TimeColumnShowTitleIcon**. Le type de données est BOOLEAN.

Propriété TimeColumnSort

TimeColumnSort

Cet attribut définit comment la colonne heure référencée dans "TimeColumnIndex" est triée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	non	Aucun tri
1	croissant	Tri croissant de la plus petite à la plus grande valeur
2	décroissant	Tri décroissant de la plus grande à la plus petite valeur

L'attribut est dynamisable avec le nom **TimeColumnSort** . Le type de données est LONG.

Propriété TimeColumnSortIndex

TimeColumnSortIndex

Cet attribut indique l'ordre de tri de la colonne heure référencée dans "TimeColumnIndex". Si vous activez la valeur sur "0", le critère de tri est supprimé dans "TimeColumnSort".

L'attribut est dynamisable avec le nom **TimeColumnSortIndex**. Le type de données est LONG.

Propriété TimeColumnTimeFormat

Format de l'heure - TimeColumnTimeFormat

Détermine quel format de l'heure sera utilisé pour représenter la colonne heure sélectionnée.

Les formats d'heure suivants sont disponibles :

Valeur	Explication
Automatique	Le format de l'heure est automatiquement défini.
HH:mm:ss.ms	Heures:Minutes:Secondes.Millisecondes, par exemple 15:35:44.240.
hh:mm:ss tt	Heures:Minutes:Secondes AM/PM, par exemple 03:35:44 PM.
hh:mm:ss.ms tt	Heures:Minutes:Secondes.Millisecondes AM/PM, par exemple 03:35:44.240 PM.

L'attribut est dynamisable avec le nom **TimeColumnTimeFormat**. Le type de données est STRING.

Propriété TimeColumnTimeRangeBase

Plage de temps - TimeColumnTimeRangeBase

Détermine l'unité de temps pour définir la plage de temps.

Les unités de temps suivantes sont disponibles :

Valeur	Description
500	500 ms
1000	1 seconde
60000	1 minute
3600000	1 heure
86400000	1 jour

L'attribut est dynamisable avec le nom **TimeColumnTimeRangeBase**. Le type de données est LONG.

Propriété TimeColumnTimeRangeFactor

Plage de temps - TimeColumnTimeRangeFactor

Détermine le facteur pour définir la plage de temps. Seuls les facteurs entiers sont admis.

L'attribut est dynamisable avec le nom **TimeColumnTimeRangeFactor**. Le type de données est SHORT.

Propriété TimeColumnUseValueColumnColors

dans les couleurs de la colonne de valeurs - TimeColumnUseValueColumnColors

Détermine si la colonne heure sélectionnée s'affiche dans les couleurs de la colonne de valeurs.

Valeur	Explication
TRUE	La colonne heure sélectionnée s'affiche dans les couleurs de la colonne de valeurs. Les paramétrages effectués dans les champs "Couleur de police" et "Couleur d'arrière-plan" restent sans effet.
FALSE	La colonne heure sélectionnée s'affiche dans les couleurs déterminée dans les champs "Couleur de police" et "Couleur d'arrière-plan".

L'attribut est dynamisable avec le nom **TimeColumnUseValueColumnColors**. Le type de données est BOOLEAN.

Propriété TimeColumnVisible

Colonnes heure - TimeColumnVisible

Les colonnes heure que vous avez créées sont énumérées dans la liste. Cliquez sur une colonne heure dans la liste pour adapter les propriétés et pour définir la plage de temps de la colonne heure.

Activez dans la liste les colonnes heure que vous souhaitez afficher dans le tableau.

Spécifie si la colonne heure sélectionnée s'affiche.

L'attribut est dynamisable avec le nom **TimeColumnVisible**. Le type de données est BOOLEAN.

TimeFormat - Tolerance

Propriété TimeFormat

Description

Détermine le format des indications de temps.

- 0: Les spécifications sont données sous la forme hh:mm
- -1: Les spécifications sont données sous la forme hh:mm:ss
- -2: Les spécifications sont données sous la forme hh:mm:ss.ms
- -3: Les spécifications sont données sous la forme hh:mm (heures entières)
- -4: Les spécifications sont données sous la forme hh:mm:ss (minutes entières)
- -5: Les spécifications sont données sous la forme hh:mm:ss.ms (secondes entières)

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété TimeJump

Description

WinCC Online Trend Control

La propriété "Index" référence une courbe. "TimeJump" détermine si les sauts de temps existant dans l'archive doivent être identifiés par la couleur définie dans "TimeJumpColor".

WinCC Online Table Control

La valeur de cet attribut ne peut être modifiée. Accès en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété TimeJumpColor

Description

WinCC Online Trend Control

La propriété "Index" référence une courbe. "TimeJumpColor" détermine la couleur qui est utilisée pour identifier les sauts de temps existant dans l'archive. L'indication sera ou non exploitée selon la propriété "TimeJump". La couleur doit être spécifiée sous la forme RVB. LONG Accès en écriture et en lecture.

WinCC Online Table Control

La valeur de cette propriété ne peut être modifiée. Accès en lecture.

Voir aussi

Objet ScreenItem (Page 132)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Propriété TimeOverlap

Description

WinCC Online Trend Control

La propriété "Index" référence une courbe. "TimeOverlap" détermine si les chevauchements de temps existant dans l'archive doivent être identifiés par la couleur définie dans "TimeOverlapColor".

WinCC Online Table Control

La valeur de cette propriété ne peut être modifiée. Accès en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété TimeOverlapColor

Description

WinCC Online Trend Control

La propriété "Index" référence une courbe. "TimeOverlapColor" détermine la couleur qui est utilisée pour identifier les chevauchements de temps existant dans l'archive. L'indication sera ou non exploitée selon l'attribut "TimeOverlap". La couleur doit être spécifiée sous la forme RVB.

WinCC Online Table Control

La valeur de cette propriété ne peut être modifiée. Accès en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

Objet ScreenItem (Page 132)

Propriété TimeRange

Description

La propriété "Index" référence un couple de colonnes ou une courbe. "TimeRange" détermine comment est définie la plage de temps à représenter.

- 0: La plage de temps à représenter est déterminée par un moment de départ (BeginTime) et une fin (EndTime)
- -1: La plage de temps à représenter est déterminée par un moment de départ ("BeginTime") et une période ("TimeRangeBase" et "TimeRangeFactor").

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété TimeRangeBase

Description

La propriété "Index" référence un couple de colonnes ou une courbe. La plage de temps à représenter pour ce couple de colonnes/cette courbe est obtenue en multipliant les valeurs "TimeRangeBase" et "TimeRangeFactor", où la valeur "TimeRangeBase" est interprétée en millisecondes.

Les propriétés "TimeRangeBase" et "TimeRangeFactor" ne sont exploitées que si la propriété "TimeRange" est positionnée, et que la valeur est donc "-1".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété TimeRangeFactor

Description

La propriété "Index" référence une courbe ou un couple de colonnes. La plage de temps à représenter pour cette courbe/ce couple de colonnes est obtenue en multipliant les valeurs "TimeRangeBase" et "TimeRangeFactor", où la valeur "TimeRangeBase" est interprétée en millisecondes.

Les propriétés "TimeRangeBase" et "TimeRangeFactor" ne sont exploitées que si la propriété "TimeRange" porte la valeur "-1".

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété TimeStamp

Description

Lit l'horodatage du dernier accès en lecture à une variable. L'horodatage est restitué en heure locale. DATE (readonly)

La fonction standard VBS "FormatDateTime(Date[, NamedFormat])" permet d'afficher en texte clair la propriété TimeStamp. L'affichage dépend du réglage actuel de la langue. Le réglage de la langue peut être réglé via la fonction standard VBS SetLocale().

Le deuxième paramètre de la fonction FormatDate() et d'autres fonctions standard VBS telles que Year, WeekDay, Day, Hour, Minute, Second, l'utilisateur peut séparer les indications comme il le veut. Si l'on veut en plus de WeekDay avoir le nom du jour de la semaine, il faut utiliser la fonction WeekdayName.

Exemple :

```
'VBS87
Dim objTag
Dim lngCount
lngCount = 0
Set objTag = HMIRuntime.Tags("Tag11")
objTag.Read
SetLocale("en-gb")
MsgBox FormatDateTime(objTag.TimeStamp)    'Output: e.g. 06/08/2002 9:07:50
MsgBox Year(objTag.TimeStamp)    'Output: e.g. 2002
MsgBox Month(objTag.TimeStamp)    'Output: e.g. 8
MsgBox Weekday(objTag.TimeStamp)    'Output: e.g. 3
MsgBox WeekdayName(Weekday(objTag.TimeStamp))    'Output: e.g. Tuesday
MsgBox Day(objTag.TimeStamp)    'Output: e.g. 6
MsgBox Hour(objTag.TimeStamp)    'Output: e.g. 9
MsgBox Minute(objTag.TimeStamp)    'Output: e.g. 7
MsgBox Second(objTag.TimeStamp)    'Output: e.g. 50
For lngCount = 0 To 4
MsgBox FormatDateTime(objTag.TimeStamp, lngCount)
Next
'lngCount = 0: Output: e.g. 06/08/2002 9:07:50
'lngCount = 1: Output: e.g. 06 August 2002
'lngCount = 2: Output: e.g. 06/08/2002
'lngCount = 3: Output: e.g. 9:07:50
'lngCount = 4: Output: e.g. 9:07
```

Exemple

L'exemple suivant affiche l'horodatage de la variable "Tag1":

```
'VBS88
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
MsgBox objTag.TimeStamp
```

Voir aussi

Objet Tag (Page 144)

Objet d'alarmes (liste) (Page 118)

Propriété TimeStepBase

Précision - TimeStepBase

Définit la précision de l'horodatage représenté dans le tableau.

Vous déterminez la précision en multipliant facteur et unité de temps. Si vous voulez par ex. représenter dans la même ligne toutes les valeurs survenant en 3 secondes, spécifiez "3" pour le facteur et "1 s" pour l'unité de temps.

Valeur	Description	Explication
0	exactement	Seules les valeurs avec exactement le même horodatage sont représentées dans une même ligne du tableau.
100	100 ms	Toutes les valeurs qui se situent dans un intervalle de 100 ms sont regroupées dans une ligne du tableau.
250	250 ms	Toutes les valeurs qui se situent dans un intervalle de 250 ms sont regroupées dans une ligne du tableau.
500	500 ms	Toutes les valeurs qui se situent dans un intervalle de 500 ms sont regroupées dans une ligne du tableau.
1000	1 s	Toutes les valeurs qui se situent dans un intervalle d'une seconde sont regroupées dans une ligne du tableau.

L'attribut est dynamisable avec le nom **TimeStepBase**. Le type de données est LONG.

Propriété TimeStepFactor

Précision - TimeStepFactor

Définit la précision de l'horodatage représenté dans le tableau.

Vous déterminez la précision en multipliant facteur et unité de temps. Si vous voulez par ex. représenter dans la même ligne toutes les valeurs survenant en 3 secondes, spécifiez "3" pour le facteur et "1 s" pour l'unité de temps.

Le facteur entré n'a aucun effet si l'unité de temps sélectionnée est "Exactement" ou si la valeur sélectionnée pour "TimeStepBase" est "0".

L'attribut est dynamisable avec le nom **TimeStepFactor**. Le type de données est LONG.

Propriété TimeZone

Description

Détermine le fuseau horaire sur la base duquel les valeurs horaires s'affichent. 4 paramétrages sont possibles:

- Fuseau horaire local
- Fuseau horaire du serveur
- UTC (Universal Time Coordinated)
- Adopter les paramétrages de projet (=> Le mode horloge propre à l'ordinateur peut être déterminé, côté propriété de l'ordinateur, via WinCC-Explorer. La sélection proposée est la suivante: WinCC V50 (mode de compatibilité => Représentation comme c'était usuel jusqu'à V5 dans les parties individuelles de l'affichage), heure locale et UTC.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété TitleColor

Arrière-plan Titre de tableau - TitleColor

Indique la couleur d'arrière-plan du titre de tableau. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TitleColor**. Le type de données est LONG.

Propriété TitleCut

Abréger les contenus - TitleCut

Détermine si les titres de colonnes sont abrégés si les colonnes ne sont pas assez larges.

Valeur	Explication
TRUE	Les titres de colonnes sont abrégés.
FALSE	Les titres de colonnes ne sont pas abrégés.

L'attribut est dynamisable avec le nom **TitleCut** . Le type de données est BOOLEAN.

Propriété TitleCut (avant WinCC V7)

Description

Détermine si le contenu des champs d'une barre de titre doit être raccourci lorsque la colonne n'est pas assez large. Accès en écriture et en lecture.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété TitleDarkShadowColor

Couleur d'ombre foncée - TitleDarkShadowColor

Indique la couleur pour le côté foncé de l'ombre. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif que si l'option "Couleur d'ombre" ou "TitleStyle" est activée.

L'attribut est dynamisable avec le nom **TitleDarkShadowColor**. Le type de données est LONG.

Propriété TitleForeColor

Couleur du texte Titre de tableau - TitleForeColor

Indique la couleur de texte du titre de tableau. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TitleForeColor**. Le type de données est LONG.

Propriété TitleGridLineColor

Couleur de la ligne de séparation / Titre - GridLineColor

Indique la couleur des lignes de séparation dans le titre du tableau. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TitleGridLineColor**. Le type de données est LONG.

Propriété TitleLightShadowColor

Couleur d'ombre claire - TitleLightShadowColor

Indique la couleur pour le côté clair de l'ombre. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif que si l'option "Couleur d'ombre" ou "TitleStyle" est activée.

L'attribut est dynamisable avec le nom **TitleLightShadowColor**. Le type de données est LONG.

Propriété Titleline

Description

TRUE, lorsque le Control est doté d'une barre de titre et peut être déplacé en Runtime. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété TitleSort

Tri par titre de colonne - TitleSort

Détermine comment déclencher le tri par titre de colonne. L'option "Défilement automatique" doit être désactivée afin de pouvoir trier par titre de colonne.

Valeur	Description	Explication
0	non	Un tri par titre de colonne n'est pas possible.
1	par clic	Le tri est déclenché par un clic sur le titre de colonne.
2	par double-clic	Le tri est déclenché par un double clic sur le titre de colonne.

L'attribut est dynamisable avec le nom **TitleSort**. Le type de données est LONG.

Propriété TitleStyle

Couleur d'ombre - TitleStyle

Détermine si une couleur d'ombre est utilisée pour le titre de tableau.

Valeur	Description	Explication
0	arbres parallèles	Une couleur d'ombre n'est pas utilisée. Représentation du titre semblant plate.
1	Bouton	Une couleur d'ombre est utilisée. Représentation du titre semblant ressortir.

L'attribut est dynamisable avec le nom **TitleStyle**. Le type de données est LONG.

Propriété Toggle

Description

TRUE, lorsque le bouton ou bouton rond doit basculer en Runtime après actionnement.
BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bouton rond (Page 217)

Objet ScreenItem (Page 132)

Propriété ToleranceHigh

Description

Détermine le seuil pour "Tolerance high" ou le restitue.
Vous déterminez le type d'analyse (en pourcent ou en absolu) avec la propriété TypeToleranceHigh.

La surveillance du seuil n'est effective que si la propriété CheckToleranceHigh est réglée sur "True".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété ToleranceLow

Description

Détermine le seuil pour "Tolerance low" ou le restitue.
Vous déterminez le type d'analyse (en pourcent ou en absolu) avec la propriété TypeToleranceLow.
La surveillance du seuil n'est effective que si la propriété CheckToleranceLow est réglée sur "True".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Toolbar

Propriété Toolbar

Description

TRUE, lorsqu'une barre d'outils s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété ToolbarAlignment (avant WinCC V7)

Description

Détermine la position de la barre d'outils ou la restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)
WinCC Online Trend Control (avant WinCC V7) (Page 288)
WinCC Function Trend Control (avant WinCC V7) (Page 284)
Objet ScreenItem (Page 132)

Propriété ToolbarAlignment

Alignement - ToolbarAlignment

Définit l'alignement de la barre d'outils dans le contrôle.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	en haut	La barre d'outils est affichée sur le bord supérieur.
1	vers le bas	La barre d'outils est affichée sur le bord inférieur.
2	à gauche	La barre d'outils est affichée sur le bord gauche.
3	vers la droite	La barre d'outils est affichée sur le bord droit.

L'attribut est dynamisable avec le nom **ToolbarAlignment**. Le type de données est LONG.

Propriété ToolbarBackColor

Couleur d'arrière-plan - ToolbarBackColor

Indique la couleur d'arrière-plan de la barre d'outils. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

La couleur d'arrière-plan configurée n'est affichée que si l'option "Afficher" est activée ou si "ToolbarUseBackColor" "TRUE".

L'attribut est dynamisable avec le nom **ToolbarBackColor**. Le type de données est LONG.

Propriété ToolbarButtonActive

Active - ToolbarButtonActive

Détermine si la fonction liée à la touche est activée en Runtime. Déclenchez la fonction correspondante en cliquant sur la touche en Runtime.

Valeur	Explication
TRUE	La fonction liée à la touche est active.
FALSE	La fonction liée à la touche n'est pas active. Vous pouvez relier votre propre fonction à la touche via un script.

L'attribut est dynamisable avec le nom **ToolbarButtonActive**. Le type de données est BOOLEAN.

Propriété ToolbarButtonAdd

Nouveau - ToolbarButtonAdd

Crée une nouvelle fonction de touche définie par l'utilisateur. Vous pouvez modifier le nom attribué par WinCC dans le champ "Nom d'objet".

L'attribut est dynamisable avec le nom **ToolbarButtonAdd** . Le type de données est STRING.

Propriété ToolbarButtonBeginGroup

Separateur - ToolbarButtonBeginGroup

Détermine si un séparateur est inséré avant la fonction de touche sélectionnée. Les séparateurs vous permettent de regrouper les boutons des fonctions de touche.

Valeur	Explication
TRUE	Le séparateur est inséré avant la fonction de touche sélectionnée.
FALSE	Aucun séparateur n'est inséré avant la fonction de touche sélectionnée.

L'attribut est dynamisable avec le nom **ToolbarButtonBeginGroup**. Le type de données est BOOLEAN.

Propriété ToolbarButtonClick AlarmControl

ToolbarButtonClick

Déclenche la fonction liée à la touche de la barre d'outils. Le programmeur peut utiliser l'"ID" pour appeler la fonction de touche correspondante.

ID	Fonction des touches	ID	Fonction des touches
1	"Aide"	21	"Alarme suivante"
2	"Dialogue de configuration"	22	"Dernière alarme"
3	"Liste d'alarmes"	23	"Dialogue texte d'info"
4	"Liste d'archives courte durée"	24	"Dialogue Commentaire"
5	Liste d'archives longue durée	25	"Loop in Alarm"
6	"Liste des alarmes verrouillées"	26	"Verrouiller alarme"
7	"Liste d'occurences"	27	"Déverrouiller l'alarme"
8	"Liste des alarmes à masquer"	28	"Masquage de l'alarme"
9	"Acquittement de l'avertisseur central"	29	"Affichage de l'alarme"
10	"Acquittement individuel"	30	"Dialogue Tri"
11	"Acquittement groupé"	31	"Dialogue Base de temps"
18	"Acquittement d'urgence"	32	"Copier les lignes"
13	"Dialogue de sélection"	33	"Connecter le backup"
14	"Dialogue des options de la visualisation"	34	"Déconnecter le backup"
15	"Dialogue de verrouillage"	36	"Première page"
17	"Imprimer"	37	"Page précédente"
35	"Exportation des données"	38	"Page suivante"
12	"Défilement automatique"	39	"Dernière page"
19	"Première alarme"	1001	"Utilisateur 1"
20	"Alarme précédente"		

L'attribut est dynamisable avec le nom ToolbarButtonClick. Le type de données est LONG.

Propriété ToolbarButtonClick FunctionTrendControl

ToolbarButtonClick

Déclenche la fonction liée à la touche de la barre d'outils. Le programmeur peut utiliser l'"ID" pour appeler la fonction de touche correspondante.

ID	Fonction des touches	ID	Fonction des touches
1	"Aide"	13	"Sélectionner plage de temps"
2	"Dialogue de configuration"	14	"Courbe précédente"
4	"Zoom - Zone"	15	"Courbe suivante"
5	"Zoom +/-"	16	"Stop"
6	"Zoom +/- axe X"	16	"Démarrer"
7	"Zoom +/- axe Y"	17	"Imprimer"
8	"Déplacer la plage de valeurs"	20	"Exportation des données"
9	"Déplacer la zone d'axe"	3	"Règle"
10	"Vue d'origine"	18	"Connecter le backup"
11	"Sélectionner l'intégration des données"	19	"Déconnecter le backup"
12	"Sélectionner courbes"	1001	"Utilisateur 1"

L'attribut est dynamisable avec le nom **ToolbarButtonClick**. Le type de données est LONG.

Propriété ToolbarButtonClick OnlineTableControl

ToolbarButtonClick

Déclenche la fonction liée à la touche de la barre d'outils. Le programmeur peut utiliser l'"ID" pour appeler la fonction de touche correspondante.

ID	Fonction des touches	ID	Fonction des touches
1	"Aide"	12	"Colonne précédente"
2	"Dialogue de configuration"	13	"Colonne suivante"
3	"Premier enregistrement"	14	"Stop"
4	"Enregistrement précédent"	14	"Démarrer"
5	"Enregistrement suivant"	15	"Imprimer"
6	"Dernier enregistrement"	20	"Exportation des données"
7	"éditer"	16	"Définir la plage de statistiques"
8	"Copier les lignes"	17	"Calculer les statistiques"
9	"Sélectionner l'intégration des données"	18	"Connecter le backup"
10	"Sélection de colonne"	19	"Déconnecter le backup"
11	"Sélectionner plage de temps"	1001	"Utilisateur 1"

L'attribut est dynamisable avec le nom **ToolbarButtonClick**. Le type de données est LONG.

Propriété ToolbarButtonClick OnlineTrendControl

ToolbarButtonClick

Déclenche la fonction liée à la touche de la barre d'outils. Le programmeur peut utiliser l'"ID" pour appeler la fonction de touche correspondante.

ID	Fonction des touches	ID	Fonction des touches
1	"Aide"	17	"Sélectionner plage de temps"
2	"Dialogue de configuration"	18	"Courbe précédente"
3	"Premier enregistrement"	19	"Courbe suivante"
4	"Enregistrement précédent"	20	"Stop"
5	"Enregistrement suivant"	20	"Démarrer"
6	"Dernier enregistrement"	21	"Imprimer"
8	"Zoom - Zone"	26	"Exportation des données"
9	"Zoom +/-"	7	"Règle"
10	"Zoomer axe du temps +/-"	22	"Définir la plage de statistiques"
11	"Zoomer l'axe de valeurs +/-"	23	"Calculer les statistiques"
12	"Déplacer la plage de valeurs"	24	"Connecter le backup"
13	"Déplacer la zone d'axe"	25	"Déconnecter le backup"

14	"Vue d'origine"	27	"Axe relative"
15	"Sélectionner l'intégration des données"	1001	"Utilisateur 1"
16	"Sélectionner courbes"		

L'attribut est dynamisable avec le nom **ToolbarButtonClick**. Le type de données est LONG.

Propriété ToolbarButtonClick RulerControl

ToolbarButtonClick

Déclenche la fonction liée à la touche de la barre d'outils. Le programmeur peut utiliser l'"ID" pour appeler la fonction de touche correspondante.

ID	Fonction de touche
1	"Aide"
2	"Dialogue de configuration"
3	"Fenêtre Règle"
4	"Plage de statistiques"
5	"Statistiques"
6	"Imprimer"
7	"Exportation des données"
1001	"Utilisateur 1"

L'attribut est dynamisable avec le nom **ToolbarButtonClick**. Le type de données est LONG.

Propriété ToolbarButtonClick UserArchiveControl

ToolbarButtonClick

Déclenche la fonction liée à la touche de la barre d'outils. Le programmeur peut utiliser l'"ID" pour appeler la fonction de touche correspondante.

ID	Fonction des touches	ID	Fonction des touches
1	"Aide"	12	"Lire variables"
2	"Dialogue de configuration"	13	"Ecrire variables"
3	"Sélectionner l'intégration des données"	14	"Importer archive"
4	"Première ligne"	15	"Exporter archive"
5	"Ligne précédente"	16	"Dialogue Tri"
6	"Ligne suivante"	17	"Dialogue de sélection"
7	"Dernière ligne"	18	"Imprimer"
8	"Supprimer lignes"	20	"Exportation des données"
9	"Couper lignes"	19	"Dialogue Base de temps"
10	"Copier les lignes"	1001	"Utilisateur 1"
11	"Insérer ligne"		

L'attribut est dynamisable avec le nom **ToolbarButtonClick**. Le type de données est LONG.

Propriété ToolbarButtonCount

ToolbarButtonCount

Cet attribut indique le nombre des fonctions de touche configurables.

L'attribut est dynamisable avec le nom **ToolbarButtonCount**. Le type de données est LONG.

Propriété ToolbarButtonEnabled

ToolbarButtonEnabled

Cet attribut définit si une touche personnalisée de la barre d'outils peut être commandée.

L'attribut est dynamisable avec le nom **ToolbarButtonEnabled**. Le type de données est BOOLEAN.

Propriété ToolbarButtonHotKey

Touche de raccourci - ToolbarButtonHotKey

Indique la touche de raccourci pour la fonction de touche sélectionnée.

Pour créer ou modifier une touche de raccourci, cliquez sur le champ "Hotkey" et appuyez la touche voulue ou la combinaison de touches.

L'attribut est dynamisable avec le nom **ToolbarButtonHotKey**. Le type de données est LONG.

Propriété ToolbarButtonID

ID objets - ToolbarButtonID

Numéro d'identification univoque pour la fonction de touche sélectionnée. Le numéro d'identification est attribué par WinCC et ne peut être modifié.

L'attribut est dynamisable avec le nom **ToolbarButtonID**. Le type de données est LONG.

Propriété ToolbarButtonIndex

ToolbarButtonIndex

Cet attribut fait référence à une fonction de touche. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à une fonction de touche précise.

Les valeurs admissibles pour "ToolbarButtonIndex" et "ToolbarButtonCount" se situent entre 0 et -1. L'attribut "ToolbarButtonCount" indique le nombre de fonctions de touche configurables.

L'attribut "ToolbarButtonIndex" est dynamisable via l'attribut **ToolbarButtonRepos**. Le type de données est LONG.

Propriété ToolbarButtonLocked

ToolbarButtonLocked

Cet attribut définit pour une touche personnalisée de la barre d'outils si l'état verrouillé et enfoncé de la touche est représenté.

L'attribut est dynamisable avec le nom **ToolbarButtonLocked**. Le type de données est BOOLEAN.

Propriété ToolbarButtonName

Nom d'objet - ToolbarButtonName

Affiche le nom pour la fonction de touche sélectionnée. Vous pouvez modifier le nom des fonctions de touche personnalisées.

L'attribut "ToolbarButtonName" pour les fonctions de touche personnalisées est dynamisable via l'attribut **ToolbarButtonRename**. Le type de données est STRING.

Propriété ToolbarButtonPasswordLevel

Autorisation de conduite - ToolbarButtonPasswordLevel

Indique l'autorisation pour la fonction de touche sélectionnée. Vous pouvez modifier l'autorisation par le biais du bouton de sélection.

Les autorisations sont configurées dans l'éditeur "User Administrator".

L'attribut est dynamisable avec le nom **ToolbarButtonPasswordLevel**. Le type de données est LONG.

Propriété ToolbarButtonRemove

Supprimer - ToolbarButtonRemove

Supprime la fonction de touche sélectionnée dans la liste. Vous ne pouvez supprimer que les fonctions de touche personnalisées.

L'attribut est dynamisable avec le nom **ToolbarButtonRemove**. Le type de données est STRING.

Propriété ToolbarButtonRename

ToolbarButtonRename

Cet attribut modifie le nom de l'élément configurable de la barre d'outils qui est référencé via l'attribut "ToolbarButtonIndex".

L'attribut est dynamisable avec le nom **ToolbarButtonRename** pour les éléments personnalisés. "ToolbarButtonRename" permet également de dynamiser l'attribut "ToolbarButtonName". Le type de données est STRING.

Propriété ToolbarButtonRepos

Haut/bas- ToolbarButtonRepos

Modifie l'ordre des fonctions de touche. "Haut" et "Bas" déplacent la fonction de touche sélectionnée dans la liste vers le haut ou le bas. Ce qui a pour conséquence que la fonction de touche dans la barre d'outils du contrôle est positionnée plus en avant ou plus en arrière.

L'attribut est dynamisable avec le nom **ToolbarButtonRepos**. Le type de données est LONG.

Propriété ToolbarButtonTooltipText

ToolbarButtonTooltipText

Cet attribut définit le texte pour l'info-bulle de la touche.

L'attribut est dynamisable avec le nom **ToolbarButtonTooltipText**. Le type de données est STRING.

Propriété ToolbarButtonUserDefined

ToolbarButtonUserDefined

Cet attribut affiche si la touche de la barre d'outils a été ajoutée par l'auteur de la configuration en tant que nouvelle touche personnalisée.

Valeur	Explication
TRUE	La touche de la barre d'outils est personnalisée.
FALSE	La touche de la barre d'outils est spécifiée par le système.

L'attribut est dynamisable avec le nom **ToolbarButtonUserDefined**. Le type de données est BOOLEAN.

Propriété ToolbarButtonVisible

Fonctions de touche - ToolbarButtonVisible

Activez dans la liste les fonctions de touche que vous souhaitez afficher dans la barre d'outils.

Cliquez sur une entrée dans la liste afin d'adapter les propriétés ou pour modifier la position dans la barre d'outils du contrôle avec les touches "Haut" et "Bas".

L'attribut est dynamisable avec le nom **ToolbarButtonVisible** . Le type de données est BOOLEAN.

Propriété ToolbarButtons

Description

Détermine les boutons contenus dans la barre d'outils ou les restitue, en mettant à 1 ou en réinitialisant le bit correspondant. Un bit est attribué à chaque bouton. Il n'y a pas de restriction au niveau de la combinaison des bits.

Bit - Valeur (hex); Valeur (déc); Bouton :

- 0 - 0x00000001; 1; Liste d'alarmes
- 1 - 0x00000002; 2; Liste d'archives courte durée
- 2 - 0x00000004; 4; Liste d'archives longue durée
- 3 - 0x00000008; 8; Acquittement avertisseurs centraux
- 4 - 0x00000010; 16; Acquittement individuel
- 5 - 0x00000020; 32; Acquittement groupé
- 6 - 0x00000040; 64; Défilement automatique
- 7 - 0x00000080; 128; Dialogue de sélection
- 8 - 0x00000100; 256; Dialogue de verrouillage
- 9 - 0x00000200; 512; Impression du journal d'alarmes
- 11 - 0x00000800; 2048; Acquittement d'urgence
- 12 - 0x00001000; 4096; Première alarme
- 13 - 0x00002000; 8192; Dernière alarme
- 14 - 0x00004000; 16384; Alarme suivante
- 15- 0x00008000; 32768; Alarme précédente
- 16 - 0x00010000; 65536; Dialogue Texte d'info
- 17 -0x00020000; 131072; Dialogue Commentaire
- 18 - 0x00040000; 262144; Loop in Alarm
- 20 - 0x00100000; 1048576; Impression de la vue actuelle
- 21 - 0x00200000; 2097152; Liste de verrouillage

- 22 - 0x00400000; 4194304; Verrouiller/Valider l'alarme
- 23 - 0x00800000; 8388608; Dialogue de tri
- 24 - 0x01000000; 16777216; Dialogue Base de temps
- 25 - 0x02000000; 33554432; Liste d'occurences

Pour pouvoir afficher plusieurs boutons, leurs valeurs doivent être liées logiquement par OU. Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété ToolbarHotKeys

Description

Détermine les raccourcis clavier sur la barre des outils, ou les restitue. Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété ToolbarShowTooltips

Info-bulles - ToolbarShowTooltips

Détermine si les infos-bulles relatives aux fonctions de touche s'affichent en Runtime.

Valeur	Explication
TRUE	Les infos-bulles s'affichent.
FALSE	Les infos-bulles ne s'affichent pas.

L'attribut est dynamisable avec le nom **ToolbarShowTooltips**. Le type de données est BOOLEAN.

L'attribut permettant de déterminer le texte d'info-bulle est "ToolbarButtonTooltipText".

Propriété ToolbarUseBackColor

Afficher Couleur d'arrière-plan - ToolbarUseBackColor

Spécifie si la couleur d'arrière-plan de la barre d'outils s'affiche.

Valeur	Explication
TRUE	La couleur d'arrière-plan de la barre d'outils s'affiche.
FALSE	La couleur d'arrière-plan de la barre d'outils ne s'affiche pas.

L'attribut est dynamisable avec le nom **ToolbarUseBackColor**. Le type de données est BOOLEAN.

Propriété ToolbarUseHotKeys

Touches de raccourci - ToolbarUseHotKeys

Détermine si les touches de raccourci sont activée pour les fonctions de touche en Runtime. Vous insérez les touches de raccourci pour les fonctions de touche dans le champ "Touche de raccourci".

Valeur	Explication
TRUE	Les touches de raccourci sont activées.
FALSE	Les touches de raccourci ne sont pas activées.

L'attribut est dynamisable avec le nom **ToolbarUseHotKeys**. Le type de données est BOOLEAN.

Propriété ToolbarVisible

Afficher barre d'outils - ToolbarVisible

Spécifie si la barre d'outils du contrôle s'affiche.

Valeur	Explication
TRUE	La barre d'outils s'affiche.
FALSE	La barre d'outils ne s'affiche pas.

L'attribut est dynamisable avec le nom **ToolbarVisible** . Le type de données est BOOLEAN.

ToolTip - TrendLower

Propriété ToolTipText

Description

Détermine le texte qui s'affichera sous forme de Tooltip lorsque vous passez avec la souris sur l'objet, ou le restitue.

STRING (Accès en écriture et en lecture)

Exemple

L'exemple suivant assigne à chaque objet de la vue "NewPDL1" un TooltipText. Il n'existe dans la vue "NewPDL1" que des objets qui intègrent la propriété ToolTipText :

```
'VBS89
Dim objScreen
Dim objScrItem
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems(lngIndex).ObjectName
Set objScrItem = objScreen.ScreenItems(strName)
'
'Assign tooltiptexts to the objects
objScrItem.ToolTipText = "Name of object is " & strName
Next
```

Voir aussi

Case d'option (Page 215)
Indicateur d'état (Page 208)
Connecteur (Page 178)
Liste de textes (Page 206)
Texte statique (Page 176)
Slider (Page 219)
Signalisation groupée (Page 203)
Rectangle arrondi (Page 173)
Bouton rond (Page 217)
Rectangle (Page 170)
Trait polygonal (Page 168)
Polygone (Page 166)
Objet OLE (Page 201)
Trait (Page 164)
Segment de cercle (Page 162)
Arc de cercle (Page 160)
Cercle (Page 158)
Groupe (Page 291)
Objet graphique (Page 196)
Segment d'ellipse (Page 156)
Arc d'ellipse (Page 154)
Ellipse (Page 151)
Champ d'E/S (Page 193)
Case à cocher (Page 213)
Bouton (Page 210)
Bargraphe (Page 184)
Objet utilisateur (Page 290)
Bargraphes tridimensionnels (Page 179)

Propriété Top

Fonction

Détermine en Pixel la coordonnée Y d'un objet (mesurée du bord gauche supérieur de la vue) ou la restitue. La coordonnée Y se réfère à l'angle gauche supérieur du rectangle entourant l'objet.

LONG (Accès en écriture et en lecture)

Exemple

L'exemple suivant déplace tous les objets de la vue "NewPDL1" de 5 Pixel vers le haut:

```
'VBS90
Dim objScreen
Dim objScrItem
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems(lngIndex).ObjectName
Set objScrItem = objScreen.ScreenItems(strName)
objScrItem.Top = objScrItem.Top - 5
Next
```

Voir aussi

Propriété Left (Page 449)
Objet ScreenItem (Page 132)

Propriété TopConnectedConnectionPointIndex

Description

Spécifie le numéro d'indice du point supérieur de connecteur ou le positionne.

Long Accès en écriture et en lecture.

Voir aussi

Connecteur (Page 178)
Objet ScreenItem (Page 132)

Propriété TopConnectedObjectName

Description

Spécifie le nom de l'objet qui est attaché au point supérieur de connecteur ou le positionne.

Long Accès en écriture et en lecture.

Voir aussi

Connecteur (Page 178)

Objet ScreenItem (Page 132)

Propriété Transparency

Description

Définit le pourcentage de transparence de l'objet et le retourne.

0 = pas de transparence ; 100 = transparence complète (invisibilité)

Les textes et champs des objets graphiques sont affichés en transparence uniquement avec "100".

Un objet entièrement transparent et donc invisible fonctionne également en runtime.

Propriété Transparent

Description

TRUE, lorsque le bouton apparaît dans la couleur opaque "BackColor". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Push Button Control (Page 268)

Objet ScreenItem (Page 132)

Propriété Trend

Description

TRUE, lorsque la tendance (croissante ou décroissante) de la valeur à surveiller doit être visualisée par une petite flèche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TrendActualize

Rafraîchissement -TrendActualize

Spécifie si la courbe sélectionnée est rafraîchie.

Valeur	Explication
TRUE	La courbe sélectionnée est toujours mise à jour.
FALSE	La courbe sélectionnée n'est pas rafraîchie. Ce paramétrage est judicieux lorsqu'une courbe archivée doit être comparée à une courbe actuelle.

L'attribut est dynamisable avec le nom **TrendActualize**. Le type de données est BOOLEAN.

Propriété TrendAdd

Nouveau - TrendAdd

Crée une nouvelle courbe.

L'attribut est dynamisable avec le nom **TrendAdd**. Le type de données est STRING.

Propriété TrendAutoRangeBeginTagName

TrendAutoRangeBeginTagName

Lorsque la plage de valeurs automatique est déterminée via des variables en ligne, l'attribut définit la variable pour le seuil inférieur de la plage de valeurs.

L'attribut est dynamisable avec le nom **TrendAutoRangeBeginTagName**. Le type de données est STRING.

Propriété TrendAutoRangeBeginValue

TrendAutoRangeBeginValue

Lorsque la plage de valeurs automatique est déterminée via la configuration des seuils inférieur et supérieur, l'attribut définit la valeur pour le seuil inférieur de la plage de valeurs.

L'attribut est dynamisable avec le nom **TrendAutoRangeBeginValue**. Le type de données est DOUBLE.

Propriété TrendAutoRangeEndTagName

TrendAutoRangeEndTagName

Lorsque la plage de valeurs automatique est déterminée via des variables en ligne, l'attribut définit la variable pour le seuil supérieur de la plage de valeurs.

L'attribut est dynamisable avec le nom **TrendAutoRangeEndTagName**. Le type de données est STRING.

Propriété TrendAutoRangeEndValue

TrendAutoRangeEndValue

Lorsque la plage de valeurs automatique est déterminée via la configuration des seuils inférieur et supérieur, l'attribut définit la valeur pour le seuil supérieur de la plage de valeurs.

L'attribut est dynamisable avec le nom **TrendAutoRangeEndValue**. Le type de données est DOUBLE.

Propriété TrendAutoRangeSource

TrendAutoRangeSource

Cet attribut définit comment la plage de valeurs automatique des données de courbe est déterminée.

Valeur	Description	Explication
0	Données d'affichage	La plage de valeurs est automatiquement déterminée via les données affichées.
1	Valeurs autorisées	La plage de valeurs est définie via la configuration des seuils inférieur et supérieur de la plage de valeurs. Les valeurs des seuils inférieur et supérieur sont représentées dans les attributs "TrendAutoRangeBeginValue" et "TrendAutoRangeEndValue".
2	Variables en ligne	Les seuils inférieur et supérieur de la plage de valeurs sont constitués à partir des valeurs des variables en ligne liées. Les variables des seuils inférieur et supérieur sont représentées dans les attributs "TrendAutoRangeBeginTagName" et "TrendAutoRangeEndTagName".

L'attribut est dynamisable avec le nom **TrendAutoRangeSource**. Le type de données est LONG.

Propriété TrendBeginTime

Heure de début - TrendBeginTime

Détermine l'heure de début de la plage de temps pour la sélection de données de la courbe sélectionnée.

L'attribut est dynamisable avec le nom **TrendBeginTime**. Le type de données est Date.

Propriété TrendColor

Couleur de courbe - TrendColor

Indique la couleur de la courbe. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TrendColor**. Le type de données est LONG.

Propriété TrendColor (avant WinCC V7)

Description

Détermine la couleur de l'indicateur de tendance ou la restitue.
L'indicateur de tendance représente par une petite flèche la tendance (croissante ou décroissante) de la mesure à surveiller. La propriété Trend doit être positionnée sur "True" pour activer son affichage. LONG Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TrendCount

TrendCount

Cet attribut indique le nombre des courbes configurées.

L'attribut est dynamisable avec le nom **TrendCount**. Le type de données est LONG.

Propriété TrendEndTime

Heure de fin - TrendEndTime

Détermine l'heure de fin de la plage de temps pour l'intégration des données de la courbe sélectionnée.

L'attribut est dynamisable avec le nom **TrendEndTime**. Le type de données est Date.

Propriété TrendExtendedColorSet

Etendu - TrendExtendedColorSet

Détermine si vous pouvez configurer la couleur de point et la couleur de remplissage et si les couleurs sont représentées en Runtime.

Valeur	Explication
TRUE	Les paramétrages effectués dans les champs "Couleur de point" et "Couleur de remplissage" sont configurable et s'appliquent en Runtime.
FALSE	Les paramétrages effectués dans les champs "Couleur de point" et "Couleur de remplissage" ne sont pas configurable et ne s'appliquent pas en Runtime.

L'attribut est dynamisable avec le nom **TrendExtendedColorSet**. Le type de données est BOOLEAN.

Propriété TrendFill

Remplie - TrendFill

Détermine si la surface sous la courbe est représentée remplie.

Valeur	Explication
TRUE	La surface sous la courbe est remplie. Si l'option "Etendu" n'est pas activée, vous pouvez utiliser la couleur de la courbe comme couleur de remplissage. Pour le type de courbe "Représenter les valeurs", l'arrière-plan du texte est représenté dans la couleur de la courbe. La couleur d'arrière-plan du contrôle est utilisée comme couleur du texte.
FALSE	La courbe n'est pas remplie.

L'attribut est dynamisable avec le nom **TrendFill**. Le type de données est BOOLEAN.

Propriété TrendFillColor

Couleur de remplissage - TrendFillColor

Indique la couleur de remplissage de la courbe. Pour le type de courbe "Représenter les valeurs", c'est la couleur d'arrière-plan du texte qui est fixée.

La couleur de remplissage est utilisée si l'option "Remplie " est activée ou si "TrendFill" "TRUE". Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

La configuration n'est possible que si l'option "Etendu" est activée ou si "TrendExtendedColorSet" "TRUE".

L'attribut est dynamisable avec le nom **TrendFillColor**. Le type de données est LONG.

Propriété TrendIndex

TrendIndex

Cet attribut fait référence à une courbe configurée. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à une courbe précise. L'indice doit toujours être activé avant de modifier les propriétés d'une courbe en Runtime.

Les valeurs admissibles pour "TrendIndex" et "TrendCount" se situent entre 0 et -1. L'attribut "TrendCount" indique le nombre de courbes configurées.

L'attribut "TrendIndex" est dynamisable via l'attribut **TrendRepos**. Le type de données est LONG.

Propriété TrendLabel

Désignation - TrendLabel

Permet de désigner la courbe sélectionnée. La désignation est affichée en Runtime si l'attribut "UseTrendNameAsLabel" possède la valeur "FALSE".

L'attribut est dynamisable avec le nom **TrendLabel**. Le type de données est STRING.

Propriété TrendLineStyle

Style de trait - TrendLineStyle

Détermine quel style de trait est utilisé pour représenter la courbe.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Trait plein	La courbe est représentée en trait continu.
1	Tirets	La courbe est représentée avec des tirets
2	Points	La courbe est représentée par une ligne composée de points.
3	Traits mixtes	La courbe est représentée par une ligne composée de traits mixtes.
4	Trait point point	La courbe est représentée par une ligne trait point point.

L'attribut est dynamisable avec le nom **TrendLineStyle**. Le type de données est LONG.

Propriété TrendLineType

Type de courbe - TrendLineType

Détermine comment la courbe est représentée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	aucune	Seuls les points sont représentés.
1	Relier les points par une ligne	Une courbe est représentée avec les points reliés de manière linéaire.

Valeur	Description	Explication
2	Courbe en escalier	Une courbe en escalier est représentée avec les points reliés.
3	Représenter les valeurs	Configurable uniquement pour OnlineTrendControl. Au lieu de points de courbe, une valeur est indiquée sur chaque horodatage ou sur le quadrillage principal de l'axe des temps.

L'attribut est dynamisable avec le nom **TrendLineType**. Le type de données est LONG.

Propriété TrendLineWidth

Largeur de ligne - TrendLineWidth

Détermine la largeur de la ligne représentée.

L'attribut est dynamisable avec le nom **TrendLineWidth**. Le type de données est LONG.

Propriété TrendLowerLimit

TrendLowerLimit

Cet attribut indique le seuil inférieur pour une variable. Si la variable n'atteint pas la valeur de "TrendLowerLimit", les valeurs sont identifiées par la couleur paramétrée dans "TrendLowerLimitColor". L'indication ne s'applique que si l'attribut "TrendLowerLimitColoring" possède la valeur "TRUE".

L'attribut est dynamisable avec le nom **TrendLowerLimit**. Le type de données est DOUBLE.

Propriété TrendLowerLimitColor

TrendLowerLimitColor

Cet attribut définit la couleur identifiant les valeurs de variable se situant en dessous de la valeur de "TrendLowerLimit". Le paramétrage ne s'applique que si l'attribut "TrendLowerLimitColoring" possède la valeur "TRUE".

L'attribut est dynamisable avec le nom **TrendLowerLimitColor**. Le type de données est LONG.

Propriété TrendLowerLimitColoring

TrendLowerLimitColoring

Définit si l'attribut "TrendLowerLimit" est utilisé pour identifier les valeurs de variable se trouvant en dessous de la valeur "TrendLowerLimit".

Valeur	Explication
TRUE	L'attribut "TrendLowerLimitColor" s'applique.
FALSE	L'attribut "TrendLowerLimitColor" ne s'applique pas.

L'attribut est dynamisable avec le nom **TrendLowerLimitColoring**. Le type de données est BOOLEAN.

TrendMeasure - TrendVisible

Propriété TrendMeasurePoints

Nombre de points de mesure - TrendMeasurePoints

Détermine le nombre de points de mesure utilisés pour représenter la courbe sélectionnée.

Détermine le nombre de couples de valeurs si la courbe est desservie par une archive utilisateur.

L'attribut est dynamisable avec le nom **TrendMeasurePoints**. Le type de données est LONG.

Propriété TrendName

Nom d'objet - TrendName

Indique le nom de la courbe sélectionnée. Vous déterminez le nom sur l'onglet "Courbes".

L'attribut "TrendName" est dynamisable via l'attribut **TrendRename**. Le type de données est STRING.

Propriété TrendPointColor

Couleur de point - TrendPointColor

Indique la couleur des points sur la courbe. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

La configuration n'est possible que si l'option "Etendu" est activée ou si "TrendExtendedColorSet" "TRUE".

L'attribut est dynamisable avec le nom **TrendPointColor**. Le type de données est LONG.

Propriété TrendPointStyle

Style de point - TrendPointStyle

Détermine comment les points sur la courbe sont représentés.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Aucun	Les points ne sont pas représentés.
1	Points	Les points de la courbe sont représentés en tant que points de 1 pixel. Le paramétrage dans le champ "Largeur de point" ne s'applique pas.
2	Carrés	Les points sont représentés en tant que carrés. Le paramétrage dans le champ "Largeur de point" s'applique.
3	cercles	Les points sont représentés en tant que cercles. Le paramétrage dans le champ "Largeur de point" s'applique.

L'attribut est dynamisable avec le nom **TrendPointStyle**. Le type de données est LONG.

Propriété TrendPointWidth

Largeur de point - TrendPointWidth

Définit la largeur de point en pixels. Vous ne pouvez configurer la largeur de point que pour les styles de points "Carrés" et "Cercles".

L'attribut est dynamisable avec le nom **TrendPointWidth**. Le type de données est LONG.

Propriété TrendProvider

Sélection de données - TrendProvider

Permet la sélection des données de la courbe sélectionnée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Aucun	Aucune sélection des données n'est configurée pour établir une liaison en Runtime via le script.
1	Variables d'archive	Sélection de données avec des variables d'archive d'une archive de valeurs de process.
2	Variables en ligne	Sélection de données avec des variables en ligne dans le stock de variables.
3	Archive utilisateur	Sélection de données avec les colonnes d'une archive utilisateur.

L'attribut est dynamisable avec le nom **TrendProvider**. Le type de données est LONG.

Propriété TrendProviderCLSID_FunctionTrend

TrendProviderCLSID_FunctionTrend

Affiche la sélection de données de la courbe sélectionnée.

Valeur	Explication
	Aucune sélection des données n'est configurée pour établir une liaison en Runtime via le script.
{416A09D2-8B5A-11D2-8B81-006097A45D48}	Sélection de données avec des variables d'archive d'une archive de valeurs de process.
{A3F69593-8AB0-11D2-A440-00A0C9DBB64E}	Sélection de données avec des variables en ligne dans le stock de variables.
{2DC9B1C8-4FC1-41B1-B354-3E469A13FBFD}	Sélection de données avec les colonnes d'une archive utilisateur.

L'attribut est dynamisable avec le nom **TrendProviderCLSID**. Le type de données est STRING.

Propriété TrendProviderCLSID_OnlineTrend

TrendProviderCLSID_OnlineTrend

Affiche la sélection de données de la courbe sélectionnée.

Valeur	Explication
	Aucune sélection des données n'est configurée pour établir une liaison en Runtime via le script.
{416A09D2-8B5A-11D2-8B81-006097A45D48}	Sélection de données avec des variables d'archive d'une archive de valeurs de process.
{A3F69593-8AB0-11D2-A440-00A0C9DBB64E}	Sélection de données avec des variables en ligne dans le stock de variables.

L'attribut est dynamisable avec le nom **TrendProviderCLSID**. Le type de données est STRING.

Propriété TrendRangeType

Paramétrage Plage de temps - TrendRangeType

Détermine pour la courbe sélectionnée la plage de temps dans laquelle la courbe est approvisionnée en données.

Vous pouvez uniquement définir le nombre des points de mesure en cas de sélection de données via les archives utilisateur.

Les possibilités de configuration suivantes sont disponibles :

Valeur	Description	Explication
0	Plage de temps	Pour l'intégration des données, déterminez l'heure de début et la plage de temps.
1	Heure de début à heure de fin	Pour l'intégration des données, déterminez l'heure de début et l'heure de fin.
2	Nombre de points de mesure	Pour l'intégration des données, déterminez l'heure de début et le nombre des points de mesure.

L'attribut est dynamisable avec le nom **TrendRangeType**. Le type de données est LONG.

Propriété TrendRemove

Supprimer - TrendRemove

Supprime les courbes sélectionnées dans la liste.

L'attribut est dynamisable avec le nom **TrendRemove**. Le type de données est STRING.

Propriété TrendRename

TrendRename

Cet attribut modifie le nom de la courbe qui est référencée via l'attribut "TrendIndex".

L'attribut est dynamisable avec le nom **TrendRename**. "TrendRename" permet également de dynamiser l'attribut "TrendName". Le type de données est STRING.

Propriété TrendRepos

Haut/bas - TrendRepos

Modifie l'ordre de la courbe sélectionnée dans la fenêtre de courbes. "Haut" et "Bas" déplacent la courbe sélectionnée dans la liste vers le haut ou le bas. Ce qui a pour conséquence que la courbe est représentée plus au premier plan ou plus en arrière-plan.

L'attribut est dynamisable avec le nom **TrendRepos**. Le type de données est LONG.

Propriété TrendSelectTagName

TrendSelectTagName

Cet attribut ouvre la boîte de dialogue pour sélectionner le nom de variable pour la sélection de données de l'axe Y dans WinCC OnlineTrendControl. Les programmeurs peuvent utiliser l'attribut pour permettre par ex. à l'utilisateur de sélectionner un nom de variable via un bouton.

L'attribut est dynamisable avec le nom **TrendSelectTagName**. Le type de données est BOOLEAN.

Propriété TrendSelectTagNameX

TrendSelectTagNameX

Cet attribut ouvre la boîte de dialogue pour sélectionner le nom de variable pour la sélection de données de l'axe X dans WinCC FunctionTrendControl. Les programmeurs peuvent utiliser l'attribut pour permettre par ex. à l'utilisateur de sélectionner un nom de variable via un bouton.

L'attribut est dynamisable avec le nom **TrendSelectTagNameX**. Le type de données est BOOLEAN.

Propriété TrendSelectTagNameY

TrendSelectTagNameY

Cet attribut ouvre la boîte de dialogue pour sélectionner le nom de variable pour la sélection de données de l'axe Y dans WinCC FunctionTrendControl. Les programmeurs peuvent utiliser l'attribut pour permettre par ex. à l'utilisateur de sélectionner un nom de variable via un bouton.

L'attribut est dynamisable avec le nom **TrendSelectTagNameY**. Le type de données est BOOLEAN.

Propriété TrendState

TrendState

Indique en runtime le statut de la liaison de données de la courbe sélectionnée.

L'attribut est dynamisable avec le nom **TrendState**. Le type de données est LONG.

Propriété TrendTagName

Nom de variable - TrendTagName

Indique le nom de la variable liée. Le bouton vous permet d'ouvrir une boîte de dialogue pour la sélection d'une variable en ligne ou d'une variable d'archive.

L'attribut est dynamisable avec le nom **TrendTagName**. Le type de données est STRING.

Propriété TrendTagNameX

Nom de variable X / Colonne X - TrendTagNameX

Indique le nom des variables liées ou de la colonne pour l'axe X. Le bouton de sélection vous permet de choisir une variable ou une colonne en ce qui concerne la sélection de données configurée.

L'attribut est dynamisable avec le nom **TrendTagNameX**. Le type de données est STRING.

Propriété TrendTagNameY

Nom de variable Y / Colonne Y - TrendTagNameY

Indique le nom des variables liées ou de la colonne pour l'axe X. Le bouton de sélection vous permet de choisir une variable ou une colonne en ce qui concerne la sélection de données configurée.

L'attribut est dynamisable avec le nom **TrendTagNameY**. Le type de données est STRING.

Propriété TrendTimeAxis

Axe de temps - TrendTimeAxis

Détermine l'axe de temps qui sera utilisé pour la courbe sélectionnée. Vous définissez les axes des temps disponibles sur l'onglet "Axes des temps".

L'attribut est dynamisable avec le nom **TrendTimeAxis**. Le type de données est STRING.

Propriété TrendTimeRangeBase

Plage de temps - TrendTimeRangeBase

Détermine l'unité de temps pour définir la plage de temps.

Les unités de temps suivantes sont disponibles :

Valeur	Description
500	500 ms
1000	1 seconde
60000	1 minute
3600000	1 heure
86400000	1 jour

L'attribut est dynamisable avec le nom **TrendTimeRangeBase**. Le type de données est LONG.

Propriété TrendTimeRangeFactor

Plage de temps - TrendTimeRangeFactor

Détermine le facteur pour définir la plage de temps. Seuls les facteurs entiers sont admis.

L'attribut est dynamisable avec le nom **TrendTimeRangeFactor**. Le type de données est SHORT.

Propriété TrendTrendWindow

Fenêtre de courbe - TrendTrendWindow

Détermine dans quelle fenêtre de courbes la courbe sélectionnée sera représentée. Vous définissez les fenêtres de courbes disponibles sur l'onglet "Fenêtre de courbes".

L'attribut est dynamisable avec le nom **TrendTrendWindow**. Le type de données est STRING.

Propriété TrendUncertainColor

TrendUncertainColor

Des valeurs possèdent un statut incertain si la valeur initiale n'est pas connue après l'activation du Runtime ou bien si une valeur de substitution est utilisée. L'attribut "TrendUncertainColor" permet de définir la couleur utilisée pour le repérage de ces valeurs. L'indication sera ou non exploitée selon l'attribut "TrendUncertainColoring".

L'attribut est dynamisable avec le nom **TrendUncertainColor**. Le type de données est LONG.

Propriété TrendUncertainColoring

TrendUncertainColoring

Des valeurs possèdent un statut incertain si la valeur initiale n'est pas connue après l'activation du Runtime ou bien si une valeur de substitution est utilisée. "TrendUncertainColoring" détermine si ce genre de valeurs doit être identifié par la couleur paramétrée dans "TrendUncertainColor".

Valeur	Explication
TRUE	Le paramétrage de l'attribut "TrendUncertainColor" s'applique.
FALSE	Le paramétrage de l'attribut "TrendUncertainColor" ne s'applique pas.

L'attribut est dynamisable avec le nom **TrendUncertainColoring**. Le type de données est BOOLEAN.

Propriété TrendUpperLimit

TrendUpperLimit

Cet attribut indique le seuil supérieur pour une variable. Si la variable n'atteint pas la valeur de "TrendUpperLimit", les valeurs sont identifiées par la couleur paramétrée dans "TrendUpperLimitColor". L'indication ne s'applique que si l'attribut "TrendUpperLimitColoring" possède la valeur "TRUE".

L'attribut est dynamisable avec le nom **TrendUpperLimit**. Le type de données est DOUBLE.

Propriété TrendUpperLimitColor

TrendUpperLimitColor

Cet attribut définit la couleur identifiant les valeurs de variable se situant en dessous de la valeur de "TrendLowerLimit". Le paramétrage ne s'applique que si l'attribut "TrendUpperLimitColoring" possède la valeur "TRUE".

L'attribut est dynamisable avec le nom **TrendUpperLimitColor**. Le type de données est LONG.

Propriété TrendUpperLimitColoring

TrendUpperLimitColoring

Définit si l'attribut "TrendUpperLimitColor" est utilisé pour identifier les valeurs de variable se trouvant en dessous de la valeur "TrendUpperLimit".

Valeur	Explication
TRUE	Le paramétrage de l'attribut "TrendUpperLimitColor" s'applique.
FALSE	Le paramétrage de l'attribut "TrendUpperLimitColor" ne s'applique pas.

L'attribut est dynamisable avec le nom **TrendUpperLimitColoring**. Le type de données est BOOLEAN.

Propriété TrendValueAlignment

Alignement - TrendValueAlignment

Définit l'alignement des valeurs représentées pour le type de courbe "Représenter les valeurs".

Les paramétrages suivants sont disponibles en fonction de l'orientation du texte de la courbe :

- Le sens d'écriture de la courbe est "de la droite" ou "de la gauche".

Valeur	Description	Explication
0	en bas	Les valeurs sont affichées en bas dans la fenêtre de courbe.
1	centré	Les valeurs sont centrées dans la fenêtre de courbe.
2	en haut	Les valeurs sont affichées en haut dans la fenêtre de courbe.

- Le sens d'écriture de la courbe est "du haut" ou "du bas".

Valeur	Description	Explication
0	à gauche	Les valeurs sont affichées à gauche dans la fenêtre de courbe.
1	centré	Les valeurs sont centrées dans la fenêtre de courbe.
2	à droite	Les valeurs sont affichées à droite dans la fenêtre de courbe.

L'attribut est dynamisable avec le nom **TrendValueAlignment**. Le type de données est LONG.

Propriété TrendValueAxis

Axe de valeurs - TrendValueAxis

Détermine l'axe de valeurs qui sera utilisé pour la courbe sélectionnée. Vous définissez les axes de valeurs disponibles sur l'onglet "Axes de valeurs".

L'attribut est dynamisable avec le nom **TrendValueAxis**. Le type de données est STRING.

Propriété TrendValueUnit

Unité - TrendValueUnit

Définit, pour le type de courbe "Représenter les valeurs", une unité qui est adjointe à la valeur à représenter, par exemple "%" ou "°C".

L'attribut est dynamisable avec le nom **TrendValueUnit**. Le type de données est STRING.

Propriété TrendVisible

Courbes - TrendVisible

Les courbes que vous avez créées sont énumérées dans la liste.

Activez dans la liste les courbes que vous souhaitez afficher dans les fenêtres de courbes.

Cliquez sur une courbe dans la liste pour adapter les propriétés et pour affecter des axes et des fenêtres de courbes à la courbe.

L'attribut est dynamisable avec le nom **TrendVisible**. Le type de données est BOOLEAN.

TrendWindow - TrendYAxis

Propriété TrendWindowAdd

Nouveau - TrendWindowAdd

Crée une nouvelle fenêtre de courbe.

L'attribut est dynamisable avec le nom **TrendWindowAdd**. Le type de données est STRING.

Propriété TrendWindowCoarseGrid

Mise à l'échelle principale - TrendWindowCoarseGrid

Détermine si les quadrillages s'affichent pour la mise à l'échelle principale.

Valeur	Explication
TRUE	Les quadrillages s'affichent pour la mise à l'échelle principale.
FALSE	Les quadrillages ne s'affichent pas pour la mise à l'échelle principale.

L'attribut est dynamisable avec le nom **TrendWindowCoarseGrid**. Le type de données est BOOLEAN.

Propriété TrendWindowCoarseGridColor

Couleur Mise à l'échelle principale - TrendWindowCoarseGridColor

Indique la couleur des quadrillages pour la mise à l'échelle principale. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TrendWindowCoarseGridColor**. Le type de données est LONG.

Propriété TrendWindowCount

TrendWindowCount

Cet attribut indique le nombre de fenêtres de courbes configurées.

L'attribut est dynamisable avec le nom **TrendWindowCount**. Le type de données est LONG.

Propriété TrendWindowFineGrid

Mise à l'échelle auxiliaire - TrendWindowFineGrid

Détermine si les quadrillages s'affichent pour la mise à l'échelle auxiliaire.

Valeur	Explication
TRUE	Les quadrillages s'affichent pour la mise à l'échelle auxiliaire.
FALSE	Les quadrillages ne s'affichent pas pour la mise à l'échelle auxiliaire.

L'attribut est dynamisable avec le nom **TrendWindowFineGrid**. Le type de données est BOOLEAN.

Propriété TrendWindowFineGridColor

Couleur Mise à l'échelle auxiliaire - TrendWindowFineGridColor

Indique la couleur des quadrillages pour la mise à l'échelle auxiliaire. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

L'attribut est dynamisable avec le nom **TrendWindowFineGridColor**. Le type de données est LONG.

Propriété TrendWindowForegroundTrendGrid

uniquemement pour la courbe de premier plan - TrendWindowForegroundTrendGrid

Détermine si les quadrillages s'affichent uniquemement pour la courbe de premier plan dans la fenêtre de courbes sélectionnée.

Valeur	Explication
TRUE	Les quadrillages s'affichent pour la courbe de premier plan dans la fenêtre de courbes.
FALSE	Les quadrillages s'affichent pour toutes les courbes dans la fenêtre de courbes.

L'attribut est dynamisable avec le nom **TrendWindowForegroundTrendGrid**. Le type de données est BOOLEAN.

Propriété TrendWindowGridInTrendColor

Dans la couleur de la courbe - TrendWindowGridInTrendColor

Détermine si les quadrillages s'affichent pour la mise à l'échelle principale dans la couleur de la courbe.

Valeur	Explication
TRUE	Les quadrillages sont représentés dans la couleur de la courbe.
FALSE	Les quadrillages sont représentés dans la couleur paramétrée dans le champ "Couleur".

L'attribut est dynamisable avec le nom **TrendWindowGridInTrendColor**. Le type de données est BOOLEAN.

Propriété TrendWindowHorizontalGrid

pour l'axe X - TrendWindowHorizontalGrid

Détermine si les quadrillages horizontaux sont affichés.

Valeur	Explication
TRUE	Les quadrillages horizontaux sont affichés.
FALSE	Les quadrillages horizontaux ne sont pas affichés.

L'attribut est dynamisable avec le nom **TrendWindowHorizontalGrid**. Le type de données est BOOLEAN.

Propriété TrendWindowIndex

TrendWindowIndex

Cet attribut fait référence à une fenêtre de courbes configurée. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à une fenêtre de courbes précise.

Les valeurs admissibles pour "TrendWindowIndex" et "TrendWindowCount" se situent entre 0 et -1. L'attribut "TrendWindowCount" indique le nombre de fenêtres de courbes configurées.

L'attribut "TrendWindowIndex" est dynamisable via l'attribut **TrendWindowRepos**. Le type de données est LONG.

Propriété TrendWindowName

Nom d'objet - TrendWindowName

Définit le nom de la fenêtre de courbes sélectionnée.

L'attribut "TrendWindowName" est dynamisable via l'attribut **TrendWindowRename**. Le type de données est STRING.

Propriété TrendWindowRemove

Supprimer - TrendWindowRemove

Supprime la fenêtre de courbe sélectionnée dans la liste.

L'attribut est dynamisable avec le nom **TrendWindowRemove**. Le type de données est STRING.

Propriété TrendWindowRename

TrendWindowRename

Cet attribut modifie le nom de la fenêtre de courbes qui est référencée via l'attribut "TrendWindowIndex".

L'attribut est dynamisable avec le nom **TrendWIndowRename**. "TrendWindowRename" permet également de dynamiser l'attribut "TrendWindowName". Le type de données est STRING.

Propriété TrendWindowRepos

Haut/bas - TrendWindowRepos

Modifie l'ordre des fenêtes de courbe. "Haut" et "Bas" déplacent la courbe sélectionnée dans la liste vers le haut ou le bas.

L'ordre dans la liste définit la position dans le contrôle. La première fenêtre de courbes s'affiche en dernière position tandis que la dernière fenêtre de courbes s'affiche en première position.

L'attribut est dynamisable avec le nom **TrendWindowRepos**. Le type de données est LONG.

Propriété TrendWindowRulerColor

Couleur de la règle - TrendWindowRulerColor

Indique la couleur de la règle. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

La couleur n'est paramétrable et représentable que si "1 - graphique" est paramétré pour la représentation ou pour "TrendWindowRulerStyle".

L'attribut est dynamisable avec le nom **TrendWindowRulerColor**. Le type de données est LONG.

Propriété TrendWindowRulerLayer

Plan des règles - TrendWindowRulerLayer

Définit le plan de représentation de la règle dans la fenêtre de courbes.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	derrière le quadrillage	La règle est placée derrière le quadrillage.
1	derrière des courbes	La règle est placée derrière les courbes et devant le quadrillage.
2	devant les courbes	La règle est placée devant les courbes.

L'attribut est dynamisable avec le nom **TrendWindowRulerLayer**. Le type de données est LONG.

Propriété TrendWindowRulerStyle

Représentation de la règle - TrendWindowRulerStyle

Définit la représentation de la règle.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	simple	La règle est représentée comme une simple ligne noire.
1	graphique	La règle est représentée dans la "Couleur" et la "Largeur" configurées.

L'attribut est dynamisable avec le nom **TrendWindowRulerStyle**. Le type de données est LONG.

Propriété TrendWindowRulerWidth

Largeur de la règle - TrendWindowRulerWidth

Détermine la largeur de la règle en pixels.

La largeur n'est paramétrable et représentable que si "1 - graphique" est paramétré pour la représentation ou pour "TrendWindowRulerStyle".

L'attribut est dynamisable avec le nom **TrendWindowRulerWidth**. Le type de données est LONG.

Propriété TrendWindowSpacePortion

Portion de zone - TrendWindowSpacePortion

Détermine la portion de la fenêtre de courbes sélectionnée lors de la représentation dans le contrôle.

L'attribut est dynamisable avec le nom **TrendWindowSpacePortion**. Le type de données est LONG.

Propriété TrendWindowStatisticRulerColor

Couleur Règle pour plage de statistiques - TrendWindowStatisticRulerColor

Indique la couleur de la règle pour la plage de statistiques. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

La couleur n'est paramétrable et représentable que si "1 - graphique" est paramétré pour la plage de statistiques ou pour "TrendWindowStatisticRulerStyle".

L'attribut est dynamisable avec le nom **TrendWindowStatisticRulerColor**. Le type de données est LONG.

Propriété TrendWindowStatisticRulerStyle

Représentation Règle pour plage de statistiques - TrendWindowStatisticRulerStyle

Définit la représentation de la règle pour la définition de la plage de statistiques.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	simple	La règle est représentée comme une simple ligne noire.
1	graphique	La règle est représentée dans la "Couleur" et la "Largeur" configurées.

L'attribut est dynamisable avec le nom **TrendWindowStatisticRulerStyle**. Le type de données est LONG.

Propriété TrendWindowStatisticRulerWidth

Largeur Règle pour plage de statistiques - TrendWindowStatisticRulerWidth

Détermine en pixels la largeur de la règle pour la plage de statistiques.

La largeur n'est paramétrable et représentable que si "1 - graphique" est paramétré pour la plage de statistiques ou pour "TrendWindowStatisticRulerStyle".

L'attribut est dynamisable avec le nom **TrendWindowStatisticRulerWidth**. Le type de données est LONG.

Propriété TrendWindowVerticalGrid

pour l'axe Y - TrendWindowVerticalGrid

Détermine si les quadrillages verticaux sont affichés.

Valeur	Explication
TRUE	Les quadrillages verticaux sont affichés.
FALSE	Les quadrillages verticaux ne sont pas affichés.

L'attribut est dynamisable avec le nom **TrendWindowVerticalGrid**. Le type de données est BOOLEAN.

Propriété TrendWindowVisible

Fenêtre de courbes - TrendWindowVisible

Les fenêtres de courbes que vous avez créées sont énumérées dans la liste.

Activez dans la liste les fenêtres de courbes que vous souhaitez afficher dans le contrôle.

Cliquez sur une entrée dans la liste afin d'adapter les propriétés de la règle et les quadrillages.

L'attribut est dynamisable avec le nom **TrendWindowVisible**. Le type de données est BOOLEAN.

Propriété TrendXAxis

Axe X - TrendXAxis

Détermine l'axe X qui sera utilisé pour la courbe sélectionnée. Vous définissez les axes X disponibles sur l'onglet "Axes X".

L'attribut est dynamisable avec le nom **TrendXAxis**. Le type de données est STRING.

Propriété TrendYAxis

Axe Y - TrendYAxis

Détermine l'axe Y qui sera utilisé pour la courbe sélectionnée. Vous définissez les axes X disponibles sur l'onglet "Axes X".

L'attribut est dynamisable avec le nom **TrendYAxis**. Le type de données est STRING.

Type

Propriété Type

Description

Lit le type d'un objet, p. ex. "Rectangle", "Circle" ou "Line".

Le type d'objet est retransmis sous la forme d'une chaîne de caractères. Readonly

Pour tous les éléments graphiques mis à disposition par WinCC, une identification spéciale est retransmise comme type. Elle se trouve sous la rubrique "Identification du type dans VBS" dans les descriptions individuelles des types d'objets WinCC.

Particularité

Pour les Controls WinCC et les objets OLE d'une autre marque, c'est l'identification ProgID indépendante de la version qui est retransmise comme type.

A partir de ProgID, vous pouvez déterminer l'identification ProgID dépendant de la version ou le "User friendly Name": Dans l'exemple suivant, "Control1" est un Control incorporé dans la vue et qui retransmet déjà l'identification ProgID indépendante de la version, via la propriété Type.

Remarque

Etant donné que ce ne sont pas tous les Controls qui ont des ProgID dépendant de la version, il vous faut intégrer une fonction de traitement des erreurs pour sortir l'identification ProgID dépendant de la version ou le UserFriendlyName. Si une fonction de traitement des erreurs est utilisée, le code se termine immédiatement sans résultat si aucune ProgID n'est trouvée.

Comment déterminer l'identification ProgID dépendant de la version:

```
'VBS91
Dim objControl
Dim strCurrentVersion
Set objControl = ScreenItems("Control1")
strCurrentVersion = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type &
"\CurVer\")
MsgBox strCurrentVersion
```

Remarque

Pour que l'exemple donné ci-dessus fonctionne, il vous faut intégrer un contrôle multimédia dans la vue.

Déterminez le User friendly Name comme suit:

```
'VBS92
Dim objControl
Dim strFriendlyName
Set objControl = ScreenItems("Control1")
strFriendlyName = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type & "\")
MsgBox strFriendlyName
```

Remarque

Pour que l'exemple donné ci-dessus fonctionne, il vous faut intégrer un contrôle multimédia dans la vue.

Exemple

L'exemple suivant affiche le type pour tous les objets de la vue "NewPDL1":

```
'VBS93
Dim objScreen
Dim objScrItem
Dim lngIndex
Dim lngAnswer
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems(lngIndex).ObjectName
Set objScrItem = objScreen.ScreenItems(strName)
lngAnswer = MsgBox(objScrItem.Type, vbOKCancel)
If vbCancel = lngAnswer Then Exit For
Next
```

Voir aussi

Objet ScreenItem (Page 132)

Type d'objet de l'objet ScreenItem (Page 150)

Propriété TypeAlarmHigh

Description

TRUE, lorsque le seuil supérieur déclenchant l'alarme est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Objet ScreenItem (Page 132)

Bargraphe (Page 184)

Propriété TypeAlarmLow

Description

TRUE, lorsque le seuil inférieur déclenchant l'alarme est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeLimitHigh4

Description

TRUE, lorsque le seuil supérieur "Reserve 4" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeLimitHigh5

Description

TRUE, lorsque le seuil supérieur "Reserve 5" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeLimitLow4

Description

TRUE, lorsque le seuil inférieur "Reserve 4" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeLimitLow5

Description

TRUE, lorsque le seuil inférieur "Reserve 5" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeToleranceHigh

Description

TRUE, lorsque le seuil inférieur "ToleranceHigh" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeToleranceLow

Description

TRUE, lorsque le seuil inférieur "ToleranceLow" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeWarningHigh

Description

TRUE, lorsque le seuil inférieur "WarningHigh" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété TypeWarningLow

Description

TRUE, lorsque le seuil inférieur "WarningLow" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

1.14.4.20 U

Un - Up

Propriété UnitColor

Description

Détermine la couleur de texte pour désigner l'unité de mesure. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété UnitFont

Description

Commande l'affichage du texte pour l'unité de mesure. Accès en lecture uniquement.

Les propriétés suivantes sont paramétrables :

- Police de caractères

- Style de police

- Taille de police

- Effet "barré"
- Effet "souligné"

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété UnitOffset

Description

L'attribut détermine la distance du texte de l'unité de mesure en se référant au haut de l'objet. L'inscription n'est positionnable que le long du diamètre vertical du disque gradué. La valeur de la propriété se réfère à la hauteur de l'objet et se mesure du haut de l'objet au bas de l'inscription.

La plage des valeurs est comprise entre 0 et 1:

0: Le bas de l'inscription se trouve à la limite supérieure de l'objet. Le texte n'est plus visible étant donné qu'il se trouve à l'extérieur de l'objet.

1: Le bas de l'inscription se trouve à la limite inférieure de l'objet.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété UnitText

Description

Détermine le texte pour l'unité de mesure. Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété UnselBGColor

Description

Détermine pour l'objet TextList la couleur d'arrière-plan des inscriptions non sélectionnées sur la liste de sélection, ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété UnselTextColor

Description

Détermine pour l'objet TextList la couleur de texte des inscriptions non sélectionnées sur la liste de sélection, ou la restitue. LONG Accès en écriture et en lecture.

Voir aussi

Liste de textes (Page 206)

Objet ScreenItem (Page 132)

Propriété UpdateCycle

Description

Retransmet en Runtime le type et la fréquence de mise à jour pour la fenêtre. Accès en lecture uniquement.

Voir aussi

Fenêtre de vue (Page 188)

Objet ScreenItem (Page 132)

Propriété UpperLimit

Description

TRUE, lorsque la spécification "UpperLimitColor" est utilisée pour identifier les valeurs variable (d'une courbe référencée via "Index") qui sont supérieures à la valeur "UpperLimitValue". BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété UpperLimitColor

Description

Détermine la couleur qui est utilisée pour identifier les valeurs variable (d'une courbe référencée via "Index") qui sont supérieures à la valeur "UpperLimitValue". L'indication sera ou non exploitée selon la valeur de la propriété "UpperLimit". La couleur doit être spécifiée sous la forme RVB. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Objet ScreenItem (Page 132)

Propriété UpperLimitTagName

Description

Définit le limite supérieure de la zone de courbe automatiquement reprise des propriétés de variables configurées dans PCS 7. Accès en écriture et en lecture.

Propriété UpperLimitValue

Description

Les valeurs variable (d'une courbe référencée via "Index") qui sont supérieures à la valeur de "UpperLimitValue" sont identifiées par la couleur spécifiée dans "UpperLimitColor". L'indication sera ou non exploitée selon la valeur de la propriété "UpperLimit".

Voir aussi

Objet ScreenItem (Page 132)

WinCC Online Table Control (avant WinCC V7) (Page 286)

WinCC Online Trend Control (avant WinCC V7) (Page 288)

WinCC Function Trend Control (avant WinCC V7) (Page 284)

Us

Propriété UseColumnBackColor

Utiliser la couleur de colonne / Arrière-plan - UseColumnBackColor

Détermine quels sont les paramétrages effectifs pour la couleur d'arrière-plan des colonnes.

Valeur	Explication
TRUE	Les paramétrages des couleurs d'arrière-plan sur les onglets "Colonne heure" ou "TimeColumnBackColor" et "Colonne de valeurs" ou "ValueColumnBackColor" sont effectifs.
FALSE	Les paramétrages des couleurs d'arrière-plan sur l'onglet "Représentation" sont effectifs.

L'attribut est dynamisable avec le nom **UseColumnBackColors**. Le type de données est BOOLEAN.

Propriété UseColumnForeColor

Utiliser la couleur de colonne / Texte - UseColumnForeColor

Détermine quels sont les paramétrages effectifs pour les couleurs du texte des colonnes.

Valeur	Explication
TRUE	Les paramétrages des couleurs de texte sur les onglets "Colonne heure" ou "TimeColumnForeColor" et "Colonne de valeurs" ou "ValueColumnForeColor" sont effectifs.
FALSE	Les paramétrages des couleurs de texte sur l'onglet "Représentation" sont effectifs.

L'attribut est dynamisable avec le nom **UseColumnForeColors**. Le type de données est BOOLEAN.

Propriété UseMessageColor

Afficher la couleur d'alarme - UseMessageColor

Détermine si les couleurs définies pour les alarmes s'affichent ou non.

Valeur	Explication
TRUE	Les couleurs des alarmes sont affichées.
FALSE	Les couleurs des alarmes ne sont pas affichées. C'est le réglage de couleurs spécifié pour le contenu du tableau qui est à la place activé sur l'onglet "Représentation".

L'attribut est dynamisable avec le nom **UseMessageColor**. Le type de données est BOOLEAN.

Propriété UseOnlineTags

Description

Définit si les propriétés de variables configurées dans PCS 7 sont adoptées comme paramètres de courbe. Accès en écriture et en lecture.

UseRangeSubstitutes-propriété

Description

TRUE, lorsqu'une mise à l'échelle propre de l'axe de valeurs est affichée pour la courbe du Contrôle de courbe. BOOLEAN Accès en écriture et en lecture.

Propriété UserData

Description

Contient la valeur qui est transmise lau script VB ors de l'exécution d'une entrée de menu personnalisée ou d'un symbole. STRING-Accès en écriture et en lecture.

Exemple

Utilisez dans l'éditeur "Menus et barres d'outils" le champ "Données utilisateur" pour transmettre un paramètre à une procédure.

L'exemple suivant montre la procédure "ActivateScreen" qui exécute un changement de vue. Entrez le nom de la vue dans le champ "Données utilisateur" :

```
Sub ActivateScreen (ByVal Item)
Dim objScreen
Dim strScreenName
' "UserData" contient le nom d'écran spécifié
' dans l'éditeur menus and toolbars.
strScreenName = Item.Userdata
HMIRuntime.BaseScreenName = strScreenName
End Sub
```

Propriété UserName

Description

Retourne le nom de l'utilisateur qui a déclenché l'objet d'alarme.

Voir aussi

Objet d'alarmes (liste) (Page 118)

Propriété UserValue1

Description

Détermine une valeur quelconque ou la restitue.
Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Voir aussi

Objet ScreenItem (Page 132)

Signalisation groupée (Page 203)

Propriété UserValue2

Description

Détermine une valeur quelconque ou la restitue.
Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété UserValue3

Description

Détermine une valeur quelconque ou la restitue.
Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété UserValue4

Description

Détermine une valeur quelconque ou la restitue.
Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Voir aussi

Signalisation groupée (Page 203)

Objet ScreenItem (Page 132)

Propriété UseSelectedTitleColor

Couleur de marquage - UseSelectedTitleColor

Détermine si une couleur de marquage est utilisée pour les titres de tableau de cellules de tableau marquées.

Valeur	Explication
TRUE	Une couleur de marquage est utilisée. Les paramétrages relatifs aux champs "Arrière-plan" ou "SelectedTitleColor" et "Texte" ou "SelectedTitleForeColor" sont effectifs en Runtime.
FALSE	Une couleur de marquage n'est pas utilisée. Les paramétrages relatifs aux champs "Arrière-plan" et "Texte" ne sont pas effectifs en Runtime.

L'attribut est dynamisable avec le nom **UseSelectedTitleColor**. Le type de données est BOOLEAN.

Propriété UseSourceBackColors

Appliquer les couleurs d'arrière-plan - UseSourceBackColors

Détermine si la couleur d'arrière-plan est acceptée par le contrôle spécifié dans le champ "Source".

Valeur	Explication
TRUE	La couleur d'arrière-plan est acceptée par le contrôle lié.
FALSE	La couleur d'arrière-plan n'est pas acceptée par le contrôle lié. Le paramétrage sur l'onglet "Représentation" est utilisé.

L'attribut est dynamisable avec le nom **UseSourceBackColors**. Le type de données est BOOLEAN.

Propriété UseSourceForeColors

Appliquer les couleurs de police - UseSourceForeColors

Détermine si la couleur de police est acceptée par le contrôle spécifié dans le champ "Source".

Valeur	Explication
TRUE	La couleur de police est acceptée par le contrôle lié.
FALSE	La couleur de police n'est pas acceptée par le contrôle lié. Le paramétrage sur l'onglet "Représentation" est utilisé.

L'attribut est dynamisable avec le nom **UseSourceForeColors**. Le type de données est BOOLEAN.

Propriété UseTableColor2

Couleur de ligne 2 - UseTableColor2

Définit si une seconde couleur de ligne est utilisée pour la représentation du tableau.

Valeur	Explication
TRUE	Les paramétrages de la "Couleur de ligne 2" sont utilisés en alternance avec la "Couleur de ligne 1".
FALSE	Les paramétrages de la "Couleur de ligne 1" sont utilisés pour toutes les lignes.

L'attribut est dynamisable avec le nom **UseTableColor2**. Le type de données est BOOLEAN.

Propriété UseTrendNameAsLabel

UseTrendNameAsLabel

Cet attribut détermine si l'attribut "TrendName" ou "TrendLabel" est utilisé comme désignation de la courbe en Runtime.

Valeur	Explication
TRUE	L'attribut "TrendName" est utilisé comme désignation de la courbe en Runtime.
FALSE	L'attribut "TrendLabel" est utilisé comme désignation de la courbe en Runtime.

L'attribut est dynamisable avec le nom **UseTrendNameAsLabel**. Le type de données est BOOLEAN.

1.14.4.21 V

Val - ValueAxis

Propriété Value

Description Objet Tag

Affiche la valeur des variables lors du dernier accès en lecture ou bien la valeur qui doit être ou a été écrite. Value représente la valeur d'une variable. Une fois que la méthode "Read" a été exécutée, elle contient la valeur lue pour les variables. Avant l'opération d'écriture, il est possible d'assigner à cette propriété la nouvelle valeur variable souhaitée. Après avoir démarré la méthode "Write", la propriété contient la valeur que l'on a tenté d'écrire en dernier.

VARIANT (Accès en écriture et en lecture)

Exemple

Dans l'exemple suivant, une nouvelle valeur est inscrite dans la variable "Tag1" :

```
'VBS94
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Value = 50
objTag.Write
```

Description WinCC Gauge Control

Détermine la valeur indiquée par l'aiguille. Plage des valeurs : "ValueMin" à "ValueMax".

Description de l'objet DataItem

Renvoie une copie de la valeur ou la référence de l'objet. La propriété Value permet également de modifier une valeur déjà ajoutée.

Exemple

Cet exemple montre comment la valeur est acceptée dans la liste et sortie en tant que Trace La valeur est ensuite modifiée, à nouveau sortie puis supprimée. Il est logique que ces opérations s'effectuent au cours d'actions séparées.

```
'VBS198
HMIRuntime.DataSet.Add "motor1", 23
HMIRuntime.Trace "motor1: " & HMIRuntime.DataSet("motor1").Value & vbNewLine
HMIRuntime.DataSet("motor1").Value = 55
```

```
HMIRuntime.Trace "motor1: " & HMIRuntime.DataSet("motor1").Value & vbNewLine
HMIRuntime.DataSet.Remove("motor1")
```

Remarque

Dans le cas des références d'objet, assurez-vous que les objets sont aptes au 'multithread'.

Voir aussi

WinCC Gauge Control (Page 258)
Methode Write (Page 783)
Methode Read (Page 753)
Objet Tag (Page 144)
Objet DataItem (Page 120)
Objet ProcessValues (liste) (Page 131)

Propriété ValueAxisAdd

Nouveau - ValueAxisAdd

Crée un nouvel axe de valeurs.

L'attribut est dynamisable avec le nom **ValueAxisAdd**. Le type de données est STRING.

Propriété ValueAxisAlign

Alignement - ValueAxisAlign

Spécifie la façon dont s'aligne l'axe des valeurs sélectionné.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	à gauche	L'axe de valeurs sélectionné est affiché à gauche de la courbe.
1	à droite	L'axe de valeurs sélectionné est affiché à droite de la courbe.

L'attribut est dynamisable avec le nom **ValueAxisAlign**. Le type de données est LONG.

Propriété ValueAxisAutoPrecisions

Chiffres après la virgule Automatique - ValueAxisAutoPrecisions

Détermine si le nombre de chiffres après la virgule est déterminé automatiquement.

Valeur	Explication
TRUE	Le nombre de chiffres après la virgule est défini automatiquement. La valeur du champ "Chiffres après la virgule" ou "ValueAxisPrecisions" reste sans effet.
FALSE	La valeur du champ "Chiffres après la virgule" ou "ValueAxisPrecisions" est effective.

L'attribut est dynamisable avec le nom **ValueAxisAutoPrecisions**. Le type de données est BOOLEAN.

Propriété ValueAxisAutoRange

Plage de valeurs Automatique - ValueAxisAutoRange

Fixe si la plage de valeurs est automatiquement déterminée.

Valeur	Explication
TRUE	La plage de valeurs est automatiquement déterminée.
FALSE	La plage de valeurs est déterminée par les valeurs configurées dans les champs "de" et "à" ou "ValueAxisBeginValue" et "ValueAxisEndValue"

L'attribut est dynamisable avec le nom **ValueAxisAutoRange**. Le type de données est BOOLEAN.

Propriété ValueAxisBeginValue

Plage de valeurs de - ValueAxisBeginValue

Définit la valeur initiale de l'axe de valeurs sélectionné. Vous pouvez configurer la valeur si l'option "Automatique" est désactivée ou si "ValueAxisAutoRange" "FALSE".

L'attribut est dynamisable avec le nom **ValueAxisBeginValue**. Le type de données est DOUBLE.

Propriété ValueAxisColor

Couleur Axe de valeurs - ValueAxisColor

Indique la couleur de l'axe de temps. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage est effectif si l'option "dans la couleur de la courbe" n'est pas activée ou si "ValueAxisInTrendColor" est égal à "FALSE".

L'attribut est dynamisable avec le nom **ValueAxisColor**. Le type de données est LONG.

Propriété ValueAxisCount

ValueAxisCount

Cet attribut indique le nombre d'axes de valeurs configurés.

L'attribut est dynamisable avec le nom **ValueAxisCount**. Le type de données est LONG.

Propriété ValueAxisEndValue

Plage de valeurs jusqu'à - ValueAxisEndValue

Fixe la valeur finale de l'axe de valeurs sélectionné. Vous pouvez configurer la valeur si l'option "Automatique" est désactivée ou si "ValueAxisAutoRange" est "FALSE".

L'attribut est dynamisable avec le nom **ValueAxisEndValue**. Le type de données est DOUBLE.

Propriété ValueAxisExponentialFormat

Représentation exponentielle - ValueAxisExponentialFormat

Détermine si les valeurs de l'axe de valeurs sélectionné s'affichent en représentation exponentielle.

Valeur	Explication
TRUE	Les valeurs sont affichées en représentation exponentielle.
FALSE	Les valeurs sont affichées en représentation décimale.

L'attribut est dynamisable avec le nom **ValueAxisExponentialFormat**. Le type de données est BOOLEAN.

Propriété ValueAxisIndex

ValueAxisIndex

Cet attribut fait référence à un axe de valeurs. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un axe de valeurs précis.

Les valeurs admissibles pour "ValueAxisIndex" et "ValueAxisCount" se situent entre 0 et -1. L'attribut "ValueAxisCount" indique le nombre d'axes de valeurs configurés.

L'attribut "ValueAxisIndex" est dynamisable via l'attribut **ValueAxisRepos**. Le type de données est LONG.

Propriété ValueAxisInTrendColor

Dans la couleur de la courbe - ValueAxisInTrendColor

Détermine si l'axe de valeurs sélectionné s'affiche dans la couleur de la courbe. Si plusieurs courbes s'affichent dans le fenêtre de courbes, c'est la couleur de la première courbe qui est utilisée. Vous déterminez l'ordre des courbes sur l'onglet "Courbes".

Valeur	Explication
TRUE	L'axe de valeurs sélectionné s'affiche dans la couleur de la courbe. Les paramétrages dans le champ "Couleur" ou "ValueAxisColor" restent sans effet.
FALSE	L'axe de valeur sélectionné s'affiche dans la couleur paramétrée dans le champ "Couleur" ou "ValueAxisColor".

L'attribut est dynamisable avec le nom **ValueAxisInTrendColor**. Le type de données est BOOLEAN.

Propriété ValueAxisInTrendColor

Dans la couleur de la courbe - ValueAxisInTrendColor

Détermine si l'axe de valeurs sélectionné s'affiche dans la couleur de la courbe. Si plusieurs courbes s'affichent dans le fenêtre de courbes, c'est la couleur de la première courbe qui est utilisée. Vous déterminez l'ordre des courbes sur l'onglet "Courbes".

Valeur	Explication
TRUE	L'axe de valeurs sélectionné s'affiche dans la couleur de la courbe. Les paramétrages dans le champ "Couleur" ou "ValueAxisColor" restent sans effet.
FALSE	L'axe de valeur sélectionné s'affiche dans la couleur paramétrée dans le champ "Couleur" ou "ValueAxisColor".

L'attribut est dynamisable avec le nom **ValueAxisInTrendColor**. Le type de données est BOOLEAN.

Propriété ValueAxisLabel

Etiquette - ValueAxisLabel

Définit l'étiquette de l'axe de valeurs sélectionné.

L'attribut est dynamisable avec le nom **ValueAxisLabel**. Le type de données est STRING.

Propriété ValueAxisName

Nom d'objet - ValueAxisName

Fixe le nom de l'axe de valeurs sélectionné.

L'attribut "ValueAxisName" est dynamisable via l'attribut **ValueAxisRename**. Le type de données est STRING.

Propriété ValueAxisPrecisions

Chiffres après la virgule - ValueAxisPrecisions

Détermine avec combien de chiffres après la virgule les valeurs de l'axe de valeurs sélectionné sont affichées. La valeur est configurable et s'applique en Runtime si l'option "Automatique" est désactivée ou si "ValueAxisAutoPrecisions" est "FALSE".

L'attribut est dynamisable avec le nom **ValueAxisPrecisions**. Le type de données est SHORT.

Propriété ValueAxisRemove

Supprimer - ValueAxisRemove

Supprime l'axe de valeurs sélectionné dans la liste.

L'attribut est dynamisable avec le nom **ValueAxisRemove**. Le type de données est STRING.

Propriété ValueAxisRename

ValueAxisRename

Cet attribut modifie le nom de l'axe de valeurs qui est référencé via l'attribut "ValueAxisIndex".

L'attribut est dynamisable avec le nom **ValueAxisRename**. "ValueAxisRename" permet également de dynamiser l'attribut "ValueAxisName". Le type de données est STRING.

Propriété ValueAxisRepos

Haut/bas - ValueAxisRepos

Modifie l'ordre des axes de valeurs. "Haut" et "Bas" déplacent l'axe de valeurs sélectionné dans la liste vers le haut ou le bas.

L'ordre dans la liste définit en Runtime la position de l'axe de valeurs dans la fenêtre de courbes. Si l'alignement est identique et que l'axe de valeurs se trouve plus haut dans la liste, il est représenté sur une position plus éloignée par rapport à la courbe.

L'attribut est dynamisable avec le nom **ValueAxisRepos** . Le type de données est LONG.

Propriété ValueAxisScalingType

Mise à l'échelle - ValueAxisScalingType

Spécifie la façon dont l'axe des valeurs sélectionné est mis à l'échelle.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	Linéaire	Mise à l'échelle linéaire de l'axe des valeurs sélectionné.
1	Logarithmique	Mise à l'échelle logarithmique de l'axe des valeurs sélectionné.
2	logarithmique inverse	Mise à l'échelle logarithmique inverse de l'axe des valeurs sélectionné.

L'attribut est dynamisable avec le nom **ValueAxisScalingType**. Le type de données est LONG.

Propriété ValueAxisTrendWindow

Fenêtre de courbes - ValueAxisTrendWindow

Détermine dans quelle fenêtre de courbes l'axe de valeurs sélectionné sera utilisé. Vous définissez les fenêtres de courbes disponibles sur l'onglet "Fenêtre de courbes".

L'attribut est dynamisable avec le nom **ValueAxisTrendWindow**. Le type de données est STRING.

Propriété ValueAxisVisible

Axes de valeurs - ValueAxisVisible

Les axes de valeurs que vous avez créés sont énumérés dans la liste. Cliquez sur un axe de valeurs dans la liste pour adapter les propriétés et pour affecter une fenêtre de courbes à l'axe de valeurs.

Activez dans la liste les axes de valeurs que vous souhaitez afficher dans les fenêtres de courbes.

L'attribut est dynamisable avec le nom **ValueAxisVisible** . Le type de données est BOOLEAN.

ValueColumn - Vi

Propriété ValueColumnAdd

Nouveau - ValueColumnAdd

Crée une nouvelle colonne de valeurs.

L'attribut est dynamisable avec le nom **ValueColumnAdd**. Le type de données est STRING.

Propriété ValueColumnAlign

Alignement - ValueColumnAlign

Spécifie la façon dont s'aligne la colonne de valeurs sélectionnée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	à gauche	La colonne de valeurs sélectionnée est affichée à gauche.
1	centré	La colonne de valeurs sélectionnée est centrée.
2	vers la droite	La colonne de valeurs sélectionnée est affichée à droite.

L'attribut est dynamisable avec le nom **ValueColumnAlign**. Le type de données est LONG.

Propriété ValueColumnAlignment

Description

La propriété "Index" référence un couple de colonnes. "ValueColumnAlignment" détermine l'alignement des valeurs variable pour ce couple de colonnes.

- 0: Les valeurs variable sont inscrites alignées à gauche.
- 1: Les valeurs variable sont inscrites centrées.
- 2: Les valeurs variable sont inscrites alignées à droite.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)

Objet ScreenItem (Page 132)

Propriété ValueColumnAutoPrecisions

Automatique - ValueColumnAutoPrecisions

Détermine si le nombre de chiffres après la virgule est déterminé automatiquement.

Valeur	Explication
TRUE	Le nombre de chiffres après la virgule est défini automatiquement. La valeur du champ "Chiffres après la virgule" ou "ValueColumnPrecisions" reste sans effet.
FALSE	La valeur du champ "Chiffres après la virgule" ou "ValueColumnPrecisions" est effective.

L'attribut est dynamisable avec le nom **ValueColumnAutoPrecisions**. Le type de données est BOOLEAN.

Propriété ValueColumnBackColor

Couleur d'arrière-plan - ValueColumnBackColor

Indique la couleur d'arrière-plan de la colonne de valeurs sélectionnée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif que si l'option "Couleur d'arrière-plan" dans la plage "Utiliser la couleur de colonne" de l'onglet "Général" est activée ou si "UseColumnBackColor" "TRUE".

L'attribut est dynamisable avec le nom **ValueColumnBackColor**. Le type de données est LONG.

Propriété ValueColumnCaption

Désignation - ValueColumnCaption

Détermine la désignation de la colonne de valeurs.

L'attribut est dynamisable avec le nom **ValueColumnCaption**. Le type de données est STRING.

Propriété ValueColumnCount

ValueColumnCount

Cet attribut indique le nombre des colonnes de valeurs configurées.

L'attribut est dynamisable avec le nom **ValueColumnCount**. Le type de données est LONG.

Propriété ValueColumnExponentialFormat

Représentation exponentielle - ValueColumnExponentialFormat

Détermine si les valeurs de la colonne de valeurs sélectionnée s'affichent en représentation exponentielle.

Valeur	Explication
TRUE	Affichage en représentation exponentielle.
FALSE	Affichage sous forme de chiffre décimal.

L'attribut est dynamisable avec le nom **ValueColumnExponentialFormat**. Le type de données est BOOLEAN.

Propriété ValueColumnForeColor

Couleur de police - ValueColumnForeColor

Indique la couleur de texte de la colonne de valeurs sélectionnée. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif que si l'option "Couleur de texte" dans la plage "Utiliser la couleur de colonne" de l'onglet "Général" est activée ou si "UseColumnForeColor" "TRUE".

L'attribut est dynamisable avec le nom **ValueColumnForeColor**. Le type de données est LONG.

Propriété ValueColumnHideText

ValueColumnHideText

Détermine si le contenu de la colonne de valeurs doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le contenu ne s'affiche pas sous forme de texte.
FALSE	Le contenu s'affiche sous forme de texte.

L'attribut est dynamisable avec le nom **ValueColumnHideText**. Le type de données est BOOLEAN.

Propriété ValueColumnHideTitleText

ValueColumnHideTitleText

Détermine si le titre de la colonne de valeurs doit s'afficher sous forme de texte.

Valeur	Explication
TRUE	Le titre ne s'affiche pas sous forme de texte.
FALSE	Le titre s'affiche sous forme de texte.

L'attribut est dynamisable avec le nom **ValueColumnHideTitleText**. Le type de données est BOOLEAN.

Propriété ValueColumnIndex

ValueColumnIndex

Cet attribut fait référence à une colonne de valeurs configurée. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à une colonne de valeurs précise.

Les valeurs admissibles pour "ValueColumnIndex" et "ValueColumnCount" se situent entre 0 et -1. L'attribut "ValueColumnCount" indique le nombre de colonnes de valeurs configurées.

L'attribut "ValueColumnIndex" est dynamisable via l'attribut **ValueColumnRepos**. Le type de données est LONG.

Propriété ValueColumnLength

Longueur en caractères - ValueColumnLength

Détermine la largeur de la colonne de valeurs sélectionnée.

L'attribut est dynamisable avec le nom **ValueColumnLength**. Le type de données est LONG.

Propriété ValueColumnName

Nom d'objet - ValueColumnName

Fixe le nom de la colonne de valeurs sélectionnée.

L'attribut "ValueColumnName" est dynamisable via l'attribut **ValueColumnRename**. Le type de données est STRING.

Propriété ValueColumnPrecisions

Chiffres après la virgule - ValueColumnPrecisions

Détermine avec combien de chiffres après la virgule les données de la colonne de valeurs sélectionnée sont affichées. La valeur peut être saisie si l'option "Automatique" est désactivée ou si "ValueColumnAutoPrecisions" est "FALSE".

L'attribut est dynamisable avec le nom **ValueColumnPrecisions**. Le type de données est SHORT.

Propriété ValueColumnProvider

Sélection de données - ValueColumnProvider

Permet la sélection des données de la colonne de valeurs sélectionnée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
1	Variables d'archive	Sélection de données avec des variables d'archive d'une archive de valeurs de process.
2	Variables en ligne	Sélection de données avec des variables en ligne dans le stock de variables.

L'attribut est dynamisable avec le nom **ValueColumnProvider**. Le type de données est LONG.

Propriété ValueColumnProviderCLSID

ValueColumnProviderCLSID

Affiche la sélection de données de la colonne de valeurs sélectionnée.

Valeur	Explication
{416A09D2-8B5A-11D2-8B81-006097A45D48}	Sélection de données avec des variables d'archive d'une archive de valeurs de process.
{A3F69593-8AB0-11D2-A440-00A0C9DBB64E}	Sélection de données avec des variables en ligne dans le stock de variables.

L'attribut est dynamisable avec le nom **ValueColumnProviderCLSID**. Le type de données est STRING.

Propriété ValueColumnRemove

Supprimer - ValueColumnRemove

Supprime la colonne de valeurs sélectionnée dans la liste.

L'attribut est dynamisable avec le nom **ValueColumnRemove**. Le type de données est STRING.

Propriété ValueColumnRename

ValueColumnRename

Cet attribut modifie le nom de la colonne de valeurs qui est référencée via l'attribut "ValueColumnIndex".

L'attribut est dynamisable avec le nom **ValueColumnRename**. "ValueColumnRename" permet également de dynamiser l'attribut "ValueColumnName". Le type de données est STRING.

Propriété ValueColumnRepos

Haut/bas - ValueColumnRepos

Modifie l'ordre des colonnes de valeurs. "Haut" et "Bas" déplacent la colonne de valeurs sélectionnée dans la liste vers le haut ou le bas.

Lorsque plusieurs colonnes de valeurs sont affectées à une colonne heure, l'ordre dans la liste détermine l'ordre des colonnes de valeurs derrière la colonne heure. Plus la colonne de valeurs est placée vers le haut, plus elle sera positionnée près de la colonne heure.

Vous modifiez l'ordre des colonnes heure avec les colonnes de valeurs affectées sur l'onglet "Colonnes heure".

L'attribut est dynamisable avec le nom **ValueColumnRepos**. Le type de données est LONG.

Propriété ValueColumnSelectTagName

ValueColumnSelectTagName

Cet attribut ouvre la boîte de dialogue pour sélectionner le nom de variable pour la sélection de données de la colonne de valeurs dans WinCC OnlineTableControl. Les programmeurs peuvent utiliser l'attribut pour permettre par ex. à l'utilisateur de sélectionner un nom de variable via un bouton.

L'attribut est dynamisable avec le nom **ValueColumnSelectTagName**. Le type de données est BOOLEAN.

Propriété ValueColumnShowIcon

ValueColumnShowIcon

Détermine si le contenu de la colonne de valeurs doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le contenu s'affiche sous forme de symbole.
FALSE	Le contenu ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **ValueColumnShowIcon**. Le type de données est BOOLEAN.

Propriété ValueColumnShowTitleIcon

ValueColumnShowTitleIcon

Détermine si le titre de la colonne de valeurs doit s'afficher sous forme de symbole.

Valeur	Explication
TRUE	Le titre s'affiche sous forme de symbole.
FALSE	Le titre ne s'affiche pas sous forme de symbole.

L'attribut est dynamisable avec le nom **ValueColumnShowTitleIcon**. Le type de données est BOOLEAN.

Propriété ValueColumnSort

ValueColumnSort

Cet attribut définit comment la colonne de valeurs référencée dans "ValueColumnIndex" est triée.

Les paramétrages suivants sont disponibles :

Valeur	Description	Explication
0	non	Aucun tri
1	croissant	Tri croissant de la plus petite à la plus grande valeur
2	décroissant	Tri décroissant de la plus grande à la plus petite valeur

L'attribut est dynamisable avec le nom **ValueColumnSort** . Le type de données est LONG.

Propriété ValueColumnSortIndex

ValueColumnSortIndex

Cet attribut indique l'ordre de tri de la colonne de valeurs référencée dans "ValueColumnIndex". Si vous activez la valeur sur "0", le critère de tri est supprimé dans "ValueColumnSort".

L'attribut est dynamisable avec le nom **ValueColumnSortIndex**. Le type de données est LONG.

Propriété ValueColumnState

ValueColumnState

Cet attribut affiche l'état de l'intégration des données de la colonne de valeurs sélectionnée en Runtime.

L'attribut est dynamisable avec le nom **ValueColumnState**. Le type de données est LONG.

Propriété ValueColumnTagName

Nom de variable - ValueColumnTagName

Indique le nom de la variable liée. Vous pouvez modifier la liaison de variable par le biais du bouton de sélection.

L'attribut est dynamisable avec le nom **ValueColumnTagName**. Le type de données est STRING.

Propriété ValueColumnTimeColumn

Colonne heure - ValueColumnTimeColumn

Détermine avec quelle colonne heure est représentée la colonne de valeurs sélectionnée. Vous définissez les colonnes heure disponibles sur l'onglet "Colonnes heure".

L'attribut est dynamisable avec le nom **ValueColumnTimeColumn**. Le type de données est STRING.

Propriété ValueColumnVisible

Colonnes de valeurs - ValueColumnVisible

Les colonnes de valeurs que vous avez créées sont énumérées dans la liste. Cliquez sur une colonne de valeurs dans la liste pour adapter les propriétés, pour affecter la colonne heure et pour définir l'intégration des données.

Activez dans la liste les colonnes de valeurs que vous souhaitez afficher dans le tableau. Une colonne de valeurs est affichée lorsqu'elle est reliée à une colonne heure.

L'attribut est dynamisable avec le nom **ValueColumnVisible** . Le type de données est BOOLEAN.

Propriété ValueMax

Description

Détermine la valeur se trouvant à l'extrémité de l'échelle. Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété ValueMin

Description

Détermine la valeur se trouvant au début de l'échelle. Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété Variable

Description

La propriété "Index" référence un couple de colonnes. "Variable" détermine le nom de la variable qui est associée à ce couple de colonnes.

Voir aussi

WinCC Online Table Control (avant WinCC V7) (Page 286)
Objet ScreenItem (Page 132)

Propriété VerticalGridLines

Verticale - VerticalGridLines

Détermine si les lignes de séparation verticales seront affichées.

Valeur	Explication
TRUE	Les lignes de séparation verticales seront affichées.
FALSE	Les lignes de séparation verticales ne seront pas affichées.

L'attribut est dynamisable avec le nom **VerticalGridLines**. Le type de données est BOOLEAN.

Propriété Visible

Description

Commande la visibilité ou l'invisibilité d'un objet ou affiche une valeur correspondante:

- TRUE: L'objet est visible
- FALSE: L'objet est invisible

VARIANT_BOOL (Accès en lecture et en écriture)

Exemple

Dans l'exemple suivant, tous les objets de la vue "NewPDL1" sont positionnés sur 'invisible':

```
'VBS95
Dim objScreen
Dim objScrItem
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
strName = objScreen.ScreenItems(lngIndex).ObjectName
Set objScrItem = objScreen.ScreenItems(strName)
objScrItem.Visible = False
Next
```

Voir aussi

Objet ScreenItem (Page 132)

Objet Layer (Page 127)

Objet HMIRuntime (Page 125)

1.14.4.22 W

Propriété Warning

Description

Détermine le début de la zone d'avertissement "Warning" sous forme de valeur graduée. Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété WarningColor

Description

Détermine la couleur de la zone "Warning" sur l'échelle. LONG Accès en écriture et en lecture.

Voir aussi

WinCC Gauge Control (Page 258)

Objet ScreenItem (Page 132)

Propriété WarningHigh

Description

Détermine le seuil supérieur pour "Warning High" ou le restitue.
La propriété "CheckWarningHigh" doit être positionnée sur TRUE pour que le seuil soit surveillé.
La représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "ColorWarningHigh" et "TypeWarningHigh".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété WarningLow

Description

Détermine le seuil inférieur pour "Warning Low" ou le restitue.
La propriété "CheckWarningLow" doit être positionnée sur TRUE pour que le seuil soit surveillé.
La représentation en cas d'atteinte du seuil et le type d'exploitation sont définis via les propriétés "ColorWarningLow" et "TypeWarningLow".

Voir aussi

Bargraphe (Page 184)

Objet ScreenItem (Page 132)

Propriété Width

Description

Règle la largeur d'un objet en Pixel ou l'affiche.

LONG

Exemple

Dans l'exemple suivant la largeur de tous les objets de la vue "NewPDL1" dont le nom commence par "Button" est multipliée par deux:

```
'VBS96
Dim objScreen
Dim cmdButton
Dim lngIndex
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
'
'Get all "Buttons"
strName = objScreen.ScreenItems(lngIndex).ObjectName
If "Button" = Left(strName, 6) Then
Set cmdButton = objScreen.ScreenItems(strName)
cmdButton.Width = cmdButton.Width * 2
    End If
```

```
Next
```

Voir aussi

Propriété Height (Page 417)

Objet ScreenItem (Page 132)

Propriété WinCCStyle

Description

Définit dans quel style un objet est représenté.

Défini par l'utilisateur	Représente l'objet en fonction des paramétrages individuels.
Global	Représente l'objet dans le design des paramètres globaux.
Style Windows	Représente l'objet dans le style Windows.

Propriété WindowBorder

Description

TRUE, lorsque la fenêtre doit être représentée avec un cadre en Runtime. Accès en lecture uniquement.

Voir aussi

Fenêtre de vue (Page 188)

Fenêtre d'application (Page 183)

Objet ScreenItem (Page 132)

Propriété WindowPositionMode

Description

Définit la position et la mise à l'échelle de la fenêtre de vue sur l'écran. Ceci n'a d'effet que si l'attribut "Fenêtre indépendante" (IndependentWindow) est réglé sur "oui" (TRUE).

Par défaut	La fenêtre de vue est positionnée dans sa taille d'origine sur l'endroit configuré sur l'écran.
Centrer	La fenêtre de vue est positionnée dans sa taille d'origine au centre de l'écran.
Maximiser	La fenêtre de vue est mise à l'échelle de la taille de l'écran.

Propriété WindowsStyle

Description

Définit si l'objet doit être représenté dans le style Windows de WinCC Version 6.2. Il est uniquement sélectionnable si le design actuellement choisi est "WinCC Classic".

TRUE, si l'objet est représenté dans le style Windows de WinCC Version 6.2.

FALSE, si l'objet n'est pas représenté dans le style Windows de WinCC Version 6.2.

Propriété WindowsStyle

Description

TRUE, lorsque l'objet correspond au style général de Windows (p. ex. des boutons gris sans bordure). BOOLEAN Accès en écriture et en lecture. Nota :

- Lorsque cette propriété est positionnée sur "True", les propriétés qui ne correspondent pas au style de Windows sont ignorées ("BorderWidth" p. ex.).
- D'un autre côté, si "BorderWidth" ou un arrière-plan autre que gris sont positionnés, "WindowsStyle" aura la valeur "False".
- Les attributs de clignotement constituent cependant une exception: Si des attributs de clignotement sont positionnés, cela n'entraînera pas automatiquement la désactivation de l'attribut "WindowsStyle".

Voir aussi

Slider (Page 219)

Bouton (Page 210)

Objet ScreenItem (Page 132)

Propriété WindowType

Description

Détermine l'objectif de la fenêtre d'alarmes.

- 0 - Liste d'alarmes : pour la représentation d'alarmes actuellement présentes.
- 1 - Liste d'archives courtes : pour la représentation des alarmes archivées.
- 2 - Liste d'archives longues : pour la représentation des alarmes archivées.
- 3 - Liste des alarmes verrouillées : pour la représentation des alarmes actuellement verrouillées.
- 4 - Liste d'occurences : permet de représenter les statistiques des alarmes.

Voir aussi

WinCC Alarm Control (avant WinCC V7) (Page 282)

Objet ScreenItem (Page 132)

Propriété WithAxes

Description

TRUE, lorsque l'échelle s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

Propriété WithLabels

Description

TRUE, lorsque le texte de l'échelle s'affiche. BOOLEAN Accès en écriture et en lecture.

Voir aussi

WinCC Slider Control (Page 275)

Objet ScreenItem (Page 132)

1.14.4.23 X - Z

Propriété XAxisColor (avant WinCC V7)

Description

L'attribut permet de définir la couleur utilisée pour l'axe X commun. La couleur est indiquée sous forme de valeur RVB. LONG Accès en écriture et en lecture.

Propriété X/YAxisAdd

Nouveau - X/YAxisAdd

Crée un nouvel axe X ou Y.

L'attribut pour l'axe X est dynamisable avec le nom **XAxisAdd** .

L'attribut pour l'axe Y est dynamisable avec le nom YAxisAdd .

Le type de données est STRING.

Propriété X/YAxisAlign

Alignement - X/YAxisAlign

Spécifie la façon dont s'aligne l'axe sélectionné.

Les paramétrages suivants sont disponibles pour l'axe X :

Valeur	Description	Explication
0	en bas	L'axe X sélectionné est affiché en dessous de la courbe.
1	en haut	L'axe X sélectionné est affiché au dessus de la courbe.

L'attribut pour l'axe X est dynamisable avec le nom XAxisAlign. Le type de données est LONG.

Les paramétrages suivants sont disponibles pour l'axe Y :

Valeur	Description	Explication
0	à gauche	L'axe Y sélectionné est affiché à gauche de la courbe.
1	vers la droite	L'axe Y sélectionné est affiché à droite de la courbe.

L'attribut pour l'axe Y est dynamisable avec le nom YAxisAlign. Le type de données est LONG.

Propriété X/YAxisAutoPrecisions

Nombre de chiffres après la virgule Automatique - X/YAxisAutoPrecisions

Détermine si le nombre de chiffres après la virgule est déterminé automatiquement.

Valeur	Explication
TRUE	Le nombre de chiffres après la virgule est défini automatiquement. La valeur du champ "Chiffres après la virgule" ou "X/YAxisPrecisions" reste sans effet.
FALSE	La valeur du champ "Chiffres après la virgule" ou "X/YAxisPrecisions" est effective.

L'attribut pour l'axe X est dynamisable avec le nom XAxisAutoPrecisions.

L'attribut pour l'axe Y est dynamisable avec le nom YAxisAutoPrecisions.

Le type de données est BOOLEAN.

Propriété X/YAxisAutoRange

Plage de valeurs Automatique - X/YAxisAutoRange

Fixe si la plage de valeurs de l'axe sélectionné est automatiquement déterminée.

Valeur	Explication
TRUE	La plage de valeurs est automatiquement déterminée.
FALSE	La plage de valeurs est déterminée par les valeurs configurées dans les champs "de" et "à" ou "X/YAxisBeginValue" et "X/YAxisEndValue".

L'attribut pour l'axe X est dynamisable avec le nom XAxisAutoRange.

L'attribut pour l'axe Y est dynamisable avec le nom YAxisAutoRange.

Le type de données est BOOLEAN.

Propriété X/YAxisBeginValue

Plage de valeurs de - X/YAxisBeginValue

Définit la plage de valeurs inférieure de l'axe sélectionné. Vous pouvez configurer la valeur si l'option "Automatique" est désactivée ou si "X/YAxisAutoRange" "FALSE".

L'attribut pour l'axe X est dynamisable avec le nom XAxisBeginValue.

L'attribut pour l'axe Y est dynamisable avec le nom YAxisBeginValue.

Le type de données est DOUBLE.

Propriété X/YAxisColor

Couleur axe X/Y - X/YAxisColor

Indique la couleur de l'axe sélectionné. Ouvrez avec le bouton le dialogue "Sélection des couleurs" permettant de sélectionner la couleur.

Le paramétrage n'est effectif que si le champ "dans la couleur de la courbe" n'est pas activé ou si "X/YAxisInTrendColor" "FALSE".

L'attribut pour l'axe X est dynamisable avec le nom XAxisColor.

L'attribut pour l'axe Y est dynamisable avec le nom YAxisColor.

Le type de données est LONG.

Propriété X/YAxisEndValue

Plage de valeurs jusqu'à - X/YAxisEndValue

Définit la plage de valeurs supérieure de l'axe sélectionné. Vous pouvez configurer la valeur si l'option "Automatique" est désactivée ou si "X/YAxisAutoRange" est "FALSE".

L'attribut pour l'axe X est dynamisable avec le nom **XAxisEndValue**.

L'attribut pour l'axe Y est dynamisable avec le nom **YAxisEndValue**.

Le type de données est DOUBLE.

Propriété X/YAxisExponentialFormat

Représentation exponentielle - X/YAxisExponentialFormat

Détermine si les valeurs de l'axe sélectionné s'affichent par représentation exponentielle.

Valeur	Explication
TRUE	Les valeurs sont affichées en représentation exponentielle.
FALSE	Les valeurs sont affichées en représentation décimale.

L'attribut pour l'axe X est dynamisable avec le nom **XAxisExponentialFormat**.

L'attribut pour l'axe Y est dynamisable avec le nom **YAxisExponentialFormat**.

Le type de données est BOOLEAN.

Propriété X/YAxisInTrendColor

Dans la couleur de la courbe - X/YAxisInTrendColor

Détermine si l'axe sélectionné s'affiche dans la couleur de la courbe. Si plusieurs courbes s'affichent dans le fenêtre de courbes, c'est la couleur de la première courbe qui est utilisée. Vous déterminez l'ordre des courbes sur l'onglet "Courbes".

Valeur	Explication
TRUE	L'axe sélectionné s'affiche dans la couleur de la courbe. Les paramétrages dans le champ "Couleur" ou "X/YAxisColor" restent sans effet.
FALSE	L'axe sélectionné s'affiche dans la couleur paramétrée dans le champ "Couleur" ou "X/YAxisColor".

L'attribut pour l'axe X est dynamisable avec le nom **XAxisInTrendColor**.

L'attribut pour l'axe Y est dynamisable avec le nom **YAxisInTrendColor**.

Le type de données est BOOLEAN.

Propriété X/YAxisLabel

Etiquette - X/YAxisLabel

Détermine le texte accompagnant l'axe sélectionné.

L'attribut pour l'axe X est dynamisable avec le nom **XAxisLabel**.

L'attribut pour l'axe Y est dynamisable avec le nom **YAxisLabel**.

Le type de données est STRING.

Propriété X/YAxisName

Nom d'objet - X/YAxisName

Fixe le nom de l'axe sélectionné.

L'attribut "XAxisName" est dynamisable via l'attribut **XAxisRename** pour l'axe X.

L'attribut "YAxisName" est dynamisable via l'attribut **YAxisRename** pour l'axe Y.

Le type de données est STRING.

Propriété X/YAxisPrecisions

Chiffres après la virgule - X/YAxisPrecisions

Détermine avec combien de chiffres après la virgule les valeurs de l'axe sélectionné sont affichées. La valeur est configurable et s'applique en Runtime si l'option "Automatique" est désactivée ou si "X/YAxisAutoPrecisions" est "FALSE".

L'attribut pour l'axe X est dynamisable avec le nom **XAxisPrecisions**.

L'attribut pour l'axe Y est dynamisable avec le nom **YAxisPrecisions**.

Le type de données est SHORT.

Propriété X/YAxisRemove

Supprimer - X/YAxisRemove

Supprime l'axe sélectionné dans la liste.

L'attribut pour l'axe X est dynamisable avec le nom **XAxisRemove** .

L'attribut pour l'axe Y est dynamisable avec le nom **YAxisRemove** .

Le type de données est STRING.

Propriété X/YAxisRepos

Haut/bas - X/YAxisRepos

Modifie l'ordre des axes. "Haut" et "Bas" déplacent l'axe sélectionné dans la liste vers le haut ou le bas.

L'ordre dans la liste définit en Runtime la position de l'axe dans la fenêtre de courbes. Si l'alignement est identique et que l'axe se trouve plus haut dans la liste, il est représenté sur une position plus éloignée par rapport à la courbe.

L'attribut pour l'axe X est dynamisable avec le nom XAxisRepos .

L'attribut pour l'axe Y est dynamisable avec le nom YAxisRepos .

Le type de données est LONG.

Propriété X/YAxisScalingType

Mise à l'échelle - X/YAxisScalingType

Spécifie la façon dont l'axe sélectionné est mis à l'échelle.

Les paramétrages suivants sont disponibles :

Valeur	Description
0	Linéaire
1	Logarithmique
2	logarithmique inverse

L'attribut pour l'axe X est dynamisable avec le nom XAxisScalingType.

L'attribut pour l'axe Y est dynamisable avec le nom YAxisScalingType.

Le type de données est LONG.

Propriété X/YAxisTrendWindow

Fenêtre de courbe - X/YAxisTrendWindow

Détermine dans quelle fenêtre de courbes l'axe sélectionné sera utilisé. Vous définissez les fenêtres de courbes disponibles sur l'onglet "Fenêtre de courbes".

L'attribut pour l'axe X est dynamisable avec le nom XAxisTrendWindow.

L'attribut pour l'axe Y est dynamisable avec le nom YAxisTrendWindow.

Le type de données est STRING.

Propriété X/YAxisVisible

Axes X/Y - X/YAxisVisible

Les axes que vous avez créés sont énumérés dans la liste. Cliquez sur un axe dans la liste pour adapter les propriétés et pour affecter l'axe à une fenêtre de courbes.

Activez dans la liste les axes que vous souhaitez afficher dans les fenêtres de courbes.

L'attribut pour l'axe X est dynamisable avec le nom **XAxisVisible** .

L'attribut pour l'axe Y est dynamisable avec le nom **YAxisVisible** .

Le type de données est BOOLEAN.

Propriété XAxisCount

XAxisCount

Cet attribut indique le nombre d'axes X configurés.

L'attribut est dynamisable avec le nom **XAxisCount**. Le type de données est LONG.

Propriété XAxisIndex

XAxisIndex

Cet attribut fait référence à un axe X configuré. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un axe X précis.

Les valeurs admissibles pour "Index" et "XAxisCount" se situent entre 0 et -1. L'attribut "XAxisCount" indique le nombre d'axes X configurés.

L'attribut "XAxisIndex" est dynamisable via l'attribut **XAxisRepos**. Le type de données est LONG.

Propriété XAxisRename

XAxisRename

Cet attribut modifie le nom de l'axe X qui est référencé via l'attribut "XAxisIndex".

L'attribut est dynamisable avec le nom **XAxisRename**. "XAxisRename" permet également de dynamiser l'attribut "XAxisName". Le type de données est STRING.

Propriété YAxisCount

YAxisCount

Cet attribut indique le nombre d'axes Y configurés.

L'attribut est dynamisable avec le nom **YAxisCount**. Le type de données est LONG.

Propriété YAxisIndex

YAxisIndex

Cet attribut fait référence à un axe Y configuré. Si vous utilisez l'attribut, vous pouvez affecter les valeurs d'autres attributs à un axe Y précis.

Les valeurs admissibles pour "YAxisIndex" et "YAxisCount" se situent entre 0 et -1. L'attribut "YAxisCount" indique le nombre d'axes Y configurés.

L'attribut "YAxisIndex" est dynamisable via l'attribut **YAxisRepos**. Le type de données est LONG.

Propriété YAxisRename

YAxisRename

Cet attribut modifie le nom de l'axe Y qui est référencé via l'attribut "XAxisIndex".

L'attribut est dynamisable avec le nom **YAxisRename**. "YAxisRename" permet également de dynamiser l'attribut "YAxisName". Le type de données est STRING.

Propriété ZeroPoint

Description

Détermine la position de l'origine du bargraphe ou la restitue.
Spécifiez la valeur en % par rapport à la hauteur totale du bargraphe. L'origine peut aussi se trouver en dehors de la zone représentée.
La propriété "ScalingType" doit être positionnée sur "2" et "Scaling" sur TRUE.

Voir aussi

Objet ScreenItem (Page 132)

Bargraphe (Page 184)

Propriété ZeroPointValue

Description

Détermine la valeur de l'origine pour l'affichage de l'échelle.

Détermine la valeur absolue pour l'origine ou la restitue.

Voir aussi

Bargraphe (Page 184)

Bargraphes tridimensionnels (Page 179)

Objet ScreenItem (Page 132)

Propriété Zoom

Description

Règle ou lit le facteur de zoom d'une vue ou d'une fenêtre de vue.

Si le facteur de zoom indiqué est inférieur à la valeur minimale, il est automatiquement réglé sur la valeur minimale. Si le facteur de zoom indiqué est supérieur à la valeur maximale, il est réglé sur la valeur maximale.

La valeur minimale du facteur de zoom est de 2 pourcent, la valeur maximale de 800 pourcent.

Pour l'objet Screen, le facteur de zoom est indiqué comme valeur numérique et pour l'objet Fenêtre de vue, en pourcent.

Exemple

Dans l'exemple suivant, le facteur de zoom double la vue actuelle:

```
'VBS97
HMIRuntime.ActiveScreen.Zoom = HMIRuntime.ActiveScreen.Zoom * 2
```

Voir aussi

Fenêtre de vue (Page 188)

Objet Screen (Page 138)

1.14.5 Méthodes

1.14.5.1 Méthodes

Liste récapitulative

Les méthodes que vous employez sur les divers objets vous permettent par exemple de lire des valeurs de variable pour poursuivre leur traitement, ou bien d'afficher les messages de diagnostic dans Runtime.

Méthodes disponibles dans VBS

Activate	GetStatusbarElement	MoveToNext	ShowInfoText
ActivateDynamic	GetStatusbarElementCollection	MoveToNextLine	ShowLockDialog
Add	GetTimeAxis	MoveToNextPage	ShowLockList
AttachDB	GetTimeAxisCollection	MoveToPrevious	ShowLongTermArchiveList
CalculateStatistic	GetTimeColumn	MoveToPreviousLine	ShowMessageList
CopyRows	GetTimeColumnCollection	MoveToPreviousPage	ShowPercentageAxis
CreateTagSet	GetToolbarButton	NextColumn	ShowPropertyDialog
CutRows	GetToolbarButtonCollection	NextTrend	ShowSelectArchive
DeactivateDynamic	GetTrend	OneToOneView	ShowSelection
DeleteRows	GetTrendCollection	PasteRows	ShowSelectionDialog
DetachDB	GetTrendWindow	PreviousColumn	ShowSelectTimeBase
Edit	GetTrendWindowCollection	PreviousTrend	ShowShortTermArchiveList
Export	GetValueAxis	Print	ShowSort
GetColumn	GetValueAxisCollection	QuitHorn	ShowSortDialog
GetColumnCollection	GetValueColumn	QuitSelected	ShowTagSelection
GetHitlistColumn	GetValueColumnCollection	QuitVisible	ShowTimebaseDialog
GetHitlistColumnCollection	GetXAxis	Read	ShowTimeSelection
GetMessageBlock	GetXAxisCollection	ReadTags	ShowTrendSelection
GetMessageBlockCollection	GetYAxis	Refresh	StartStopUpdate
GetMessageColumn	GetYAxisCollection	Remove	Stop
GetMessageColumnCollection	HideAlarm	RemoveAll	Trace
GetOperatorMessage	Item-Methode	Restore	UnhideAlarm
GetOperatorMessageCollection	LockAlarm	SelectedStatisticArea	UnlockAlarm
GetRulerBlock	LoopInAlarm	ServerExport	Write
GetRulerBlockCollection	MoveAxis	ServerImport	WriteTags
GetRulerColumn	MoveRuler (Page 742)	ShowColumnSelection	ZoomArea
GetRulerColumnCollection	MoveToFirst	ShowComment	ZoomInOut
GetRulerData	MoveToFirstLine	ShowDisplayOptionsDialog	ZoomInOutTime

GetStatisticAreaColumn	MoveToFirstPage	ShowEmergencyQuitDialog	ZoomInOutValues
GetStatisticAreaColumnCollection	MoveToLast	ShowHelp	ZoomInOutX
GetStatisticResultColumn	MoveToLastLine	ShowHideList	ZoomInOutY
GetStatisticResultColumnCollection	MoveToLastPage	ShowHitList	ZoomMove

1.14.5.2 Méthodes A à E

Méthode Activate

Fonction

Active la vue ou l'élément de vue spécifié.

Remarque

Il est préférable que les affectations du focus ne soient pas configurées dans l'événement ButtonDown. Des états peuvent s'avérer être non valides étant donné que le focus est explicitement demandé lors de l'événement ButtonDown.

Syntaxe

```
Expression.Activate
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "Screen" ou "ScreenItem".

Paramètres

- -

Exemples

L'exemple suivant montre l'utilisation pour le type "Screen" :

```
'VBS98
Dim objScreen
MsgBox HMIRuntime.ActiveScreen.ObjectName    'Output of active screen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
```

```
objScreen.Activate    'Activate "ScreenWindow1"
MsgBox HMIRuntime.ActiveScreen.ObjectName    'New output of active screen
```

L'exemple suivant montre l'utilisation pour le type "ScreenItem" :

```
'VBS158
MsgBox HMIRuntime.ActiveScreen.ActiveScreenItem.ObjectName  'Output of active screen item
HMIRuntime.ActiveScreen.ScreenItems("IOField1").Activate
MsgBox HMIRuntime.ActiveScreen.ActiveScreenItem.ObjectName  'New output of active screen
item
```

Voir aussi

Objet ScreenItem (Page 132)

Objet Screen (Page 138)

Méthode ActivateDynamic

Fonction

Active de manière dynamique un déclenchement pour la propriété définie et avec le cycle défini pendant le runtime. À chaque activation du déclenchement, un autre cycle d'activation peut être utilisé.

Vous trouverez des exemples de cette méthode au chapitre "VBS pour créer des procédures et actions > Créer et traiter des actions > Déclenchement > Déclenchement d'animation".

Syntaxe

```
Expression.ActivateDynamic (ByVAl bstrPropertyName As String, ByVal
bstrCycleName As String)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

Paramètre	Description
bstrPropertyName	Nom de la propriété à laquelle le déclenchement se réfère.
bstrCycleName	Nom du cycle d'activation, p. ex. "CycleTime1s".

Voir aussi

Déclenchement d'animation (Page 75)

Méthode Add

Description pour TagSet

Ajoute une variable à la liste. La variable peut être ajoutée par le nom ou par une référence à un objet Tag.

Syntaxe

```
Expression.Add [Tag]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "TagSet".

Paramètres

VARIANT

Paramètres	Description
Jour	Nom d'une variable WinCC ou référence à un objet Tag qu'il faut ajouter à la liste.

Exemple

Dans l'exemple suivant, un objet TagSet est créé et une variable est ajoutée.

```
'VBS170
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
```

Les objets Tag peuvent également être admis comme suit.

```
'VBS171
Dim Tag
Set Tag = HMIRuntime.Tags("Motor2")
Dim group2
Set group2 = HMIRuntime.Tags.CreateTagSet
group2.Add Tag
```

Description pour l'objet DataSet

Ajoute à la liste une valeur ou une référence sur l'objet.

Remarque

L'objet DataSet ne prend en charge aucune classe.
Les objets de type Screen, Screens, ScreenItem, ScreenItems, Tag et TagSet ne peuvent pas être acceptés dans la liste DataSet.
Dans le cas des références d'objet, assurez-vous que les objets sont aptes au 'multithread'.

Syntaxe

```
Expresion.Add [vtName], [vtUserData]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "DataSet".

Paramètres

VARIANT

Paramètres	Description
vtName	Nom sous lequel la valeur ou la variable doit être ajoutée à la liste.
vtUserData	Valeur qui doit être ajoutée à la liste.

Exemple

Dans cet exemple, une valeur est acceptée dans la liste DataSet.

```
'VBS172
HMIRuntime.DataSet.Add "Motor1",23
```

Voir aussi

Objet TagSet (Liste) (Page 149)

Objet DataSet (Liste) (Page 123)

Méthode AttachDB

Fonction

Exécute la fonction de touche "Connecter le backup" du contrôle.

Syntaxe

```
Expression.AttachDB()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode CalculateStatistic

Fonction

Exécute la fonction de touche "Calculer les statistiques" du OnlineTrendControl et du OnlineTableControl.

Syntaxe

```
Expression.CalculateStatistic()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode CopyRows

Fonction

Exécute la fonction de touche "Copier lignes" du contrôle.

Syntaxe

```
Expression.CopyRows()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode Create

Fonction

Crée un nouvel objet d'alarme.

Syntaxe

```
Expression.Create (VARIANT vtApplication)
```

Expression

Impératif. Une expression qui renvoie un objet de type "Alarm".

Paramètre

```
VARIANT
```

Paramètre	Description
vtApplication	Nom de l'objet d'alarme (optionnel)

Voir aussi

Objet d'alarmes (liste) (Page 118)

Méthode CreateTagSet

Fonction

Crée un nouvel objet TagSet. Cet objet peut être utilisé pour les accès Multi-Tag optimisés.

Syntaxe

```
Expression.CreateTagSet()
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "TagSet".

Paramètres

VARIANT

Exemple

L'exemple suivant montre comment créer un objet TagSet.

```
'VBS168
'Build a Reference to the TagSet Object
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
```

Voir aussi

Objet TagSet (Liste) (Page 149)
Objet Tags (Liste) (Page 147)

Méthode CutRows

Fonction

Exécute la fonction de touche "Couper les lignes" du UserArchiveControl.

Syntaxe

```
Expression.CutRows()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode DeactivateDynamic

Fonction

Désactive le déclenchement utilisé avec la méthode "ActivateDynamic" pour la propriété définie pendant le runtime.

Syntaxe

```
Expression.DeactivateDynamic(ByVal bstrPropertyName As String)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

String

Paramètre	Description
bstrPropertyName	Nom de la propriété à laquelle le déclenchement se réfère.

Méthode DeleteRows

Fonction

Exécute la fonction de touche "Supprimer les lignes" du UserArchiveControl.

Syntaxe

```
Expression.DeleteRows()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode DetachDB

Fonction

Exécute la fonction de touche "Déconnecter le backup" du contrôle.

Syntaxe

```
Expression.DetachDB()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode Edit

Fonction

Exécute la fonction de touche "Éditer" du OnlineTableControl.

Syntaxe

```
Expression.Edit()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode Export

Fonction

Exécute la fonction de touche "Exporter archive" ou "Exporter les données" du contrôle.

Syntaxe

```
Expression.Export()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

VARIANT

1.14.5.3 Méthodes Get

Méthode GetColumn

Fonction

Renvoie l'objet de colonne de WINCC UserArchiveControl désigné par un nom ou un index en tant que type "ICCAxUAColumn".

Syntaxe

```
Expression.GetColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne du UserArchiveControl.

Exemple

```
'VBS312
Dim ctrl
Dim objColumn
Set ctrl = ScreenItems("UAControl")
Set objColumn = ctrl.GetColumn("Field1")
objColumn.Length = 30
Set objColumn = ctrl.GetColumn(3)
objColumn.Align = 2
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "Column", vous écrivez p. ex. "objColumn.Align" et non "objColumn.ColumnAlign".

Voir aussi

Objet Column (liste) (Page 228)

Méthode GetColumnCollection

Fonction

Restitue la liste de tous les objets de colonne de WinCC UserArchiveControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetColumnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS313
Dim ctrl
Dim coll
Dim field
Set ctrl = ScreenItems("UAControl")
Set coll = ctrl.GetColumnCollection
HMIRuntime.Trace "Number of fields:" & coll.Count & vbCrLf
For Each field In coll
 HMIRuntime.Trace field.Name & vbCrLf
 HMIRuntime.Trace field.Type & vbCrLf
 HMIRuntime.Trace field.Length & vbCrLf
 HMIRuntime.Trace field.Caption & vbCrLf
Next
```

Voir aussi

Objet Column (liste) (Page 228)

Méthode GetHitlistColumn

Fonction

Restitue l'objet de colonne de la liste d'occurrences de WinCC AlarmControl désigné par un nom ou un index en tant que type "ICCAxMessageColumn".

Syntaxe

```
Expression.GetHitlistColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne de la liste d'occurrences.

Exemple

```
'VBS314
Dim ctrl
Dim objHitlistColumn
Set ctrl = ScreenItems("AlarmControl")
Set objHitlistColumn = ctrl.GetHitlistColumn("Date")
objHitlistColumn.Sort = 2
Set objHitlistColumn = ctrl.GetHitlistColumn("AverageComeGo")
objHitlistColumn.Visible = FALSE
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "HitlistColumn", vous écrivez p. ex. "objHitlistColumn.Visible" et non "objHitlistColumn.HitlistColumnVisible".

Voir aussi

Objet HitlistColumn (liste) (Page 229)

Méthode GetHistlistColumnCollection

Fonction

Restitue la liste de tous les objets de colonne de la liste d'occurrences de WinCC AlarmControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetHitlisteColumnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS315
Dim ctrl
Dim coll
Dim hitlistcol
Set ctrl = ScreenItems("AlarmControl")
Set coll = ctrl.GetHitlistColumnCollection
HMIRuntime.Trace "Number of hitlist columns:" & coll.Count & vbCrLf
For Each hitlistcol In coll
 HMIRuntime.Trace hitlistcol.Index & vbCrLf
 HMIRuntime.Trace hitlistcol.Name & vbCrLf
 HMIRuntime.Trace hitlistcol.Sort & vbCrLf
 HMIRuntime.Trace hitlistcol.SortIndex & vbCrLf
Next
```

Voir aussi

Objet HitlistColumn (liste) (Page 229)

Méthode GetMessageBlock

Fonction

Restitue l'objet de bloc d'alarme de WinCC AlarmControl désigné par un nom ou un index en tant que type "ICCAxMessageBlock".

Syntaxe

```
Expression.GetMessageBlock(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom du bloc d'alarme.

Exemple

```
'VBS316
Dim ctrl
Dim objMsgBlock
Set ctrl = ScreenItems("AlarmControl")
Set objMsgBlock = ctrl.GetMessageBlock("Date")
objMsgBlock.Align = 2
Set objMsgBlock = ctrl.GetMessageBlock("Number")
objMsgBlock.LeadingZeros = 4
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "MessageBlock", vous écrivez p. ex. "objMsgBlock.Align" et non "objMsgBlock.MessageBlockAlign".

Voir aussi

Objet MessageBlock (liste) (Page 230)

Méthode GetMessageBlockCollection

Fonction

Restitue la liste de tous les objets de bloc d'alarme de WinCC AlarmControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetMessageBlockCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS317
Dim ctrl
Dim coll
Dim msgblock
Set ctrl = ScreenItems("AlarmControl")
Set coll = ctrl.GetMessageBlockCollection
For Each msgblock In coll
 msgblock.Align = 1
 msgblock.Length = 12
 msgblock.Selected = TRUE
Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "MessageBlock", vous écrivez p. ex. "msgblock.Align" et non "msgblock.MessageBlockAlign".

Voir aussi

Objet MessageBlock (liste) (Page 230)

Méthode GetMessageColumn

Fonction

Restitue l'objet de colonne de WinCC AlarmControl désigné par un nom ou un index en tant que type "ICCAxMessageColumn".

Syntaxe

```
Expression.GetMessageColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne dans la liste d'alarmes.

Exemple

```
'VBS318
Dim ctrl
Dim objMessColumn
Set ctrl = ScreenItems("AlarmControl")
Set objMessColumn = ctrl.GetMessageColumn("Date")
objMessColumn.Visible = FALSE
Set objMessColumn = ctrl.GetMessageColumn("Number")
objMessColumn.Sort = 1
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "MessageColumn", vous écrivez p. ex. "objMessColumn.Visible" et non "objMessColumn.MessageColumnVisible".

Voir aussi

Objet MessageColumn (liste) (Page 231)

Méthode GetMessageColumnCollection

Fonction

Restitue la liste de tous les objets de colonne de WinCC AlarmControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetMessageColumnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS319
Dim ctrl
Dim coll
Dim msgcol
Set ctrl = ScreenItems("AlarmControl")
Set coll = ctrl.GetMessageColumnCollection
HMIRuntime.Trace "Number of message columns:" & coll.Count & vbCrLf
For Each msgcol In coll
 HMIRuntime.Trace msgcol.Index & vbCrLf
 HMIRuntime.Trace msgcol.Name & vbCrLf
 HMIRuntime.Trace msgcol.Sort & vbCrLf
 HMIRuntime.Trace msgcol.SortIndex & vbCrLf
Next
```

Voir aussi

Objet MessageColumn (liste) (Page 231)

Méthode GetOperatorMessage

Fonction

Restitue l'objet d'alarme de conduite de WinCC AlarmControl désigné par un nom ou un index en tant que type "ICCAxOperatorMessage".

Syntaxe

```
Expression.GetOperatorMessage(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de l'alarme de conduite.

Exemple

```
'VBS320
Dim ctrl
```

```
Dim objOpMess
Set ctrl = ScreenItems("AlarmControl")
Set objOpMess = ctrl.GetOperatorMessage(0)
objOpMess.Source1 = "Number"
objOpMess.SourceType1 = 1
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "OperatorMessage", vous écrivez p. ex. "objOpMess.Source1" et non "objOpMess.OperatorMessageSource1".

Voir aussi

Objet OperatorMessage (liste) (Page 232)

Méthode GetOperatorMessageCollection

Fonction

Restitue la liste de tous les objets d'alarme de conduite de WinCC AlarmControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetOperatorMessageCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS321
Dim ctrl
Dim coll
Dim opmsg
Set ctrl = ScreenItems("AlarmControl")
Set coll = ctrl.GetOperatorMessageCollection
For Each opmsg In coll
 HMIRuntime.Trace opmsg.Index & vbCrLf
 HMIRuntime.Trace opmsg.Name & vbCrLf
 HMIRuntime.Trace opmsg.Number & vbCrLf
 HMIRuntime.Trace opmsg.Selected & vbCrLf
Next
```

Voir aussi

Objet OperatorMessage (liste) (Page 232)

Méthode GetRow

Fonction

Restitue l'objet de ligne des Controls basés sur des tables désigné par un numéro de ligne en tant que type "ICCAxDataRow".

Syntaxe

```
Expression.GetRow(ByVal IRow As Long)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

Long

Paramètre	Description
IRow	Numéro de la ligne de Control voulue.

Exemple

```
'VBS356
Dim ctrl
Dim lIndex
Dim lCellIndex
Set ctrl = ScreenItems("UAControl")
Set coll = ctrl.GetRowCollection
'enumerate and trace out row numbers
For lIndex = 1 To coll.Count
 HMIRuntime.trace "Row: " & (ctrl.GetRow(lIndex).RowNumber) & "  "
 'enumerate and trace out column titles and cell texts
 For lCellIndex = 1 To ctrl.GetRow(lIndex).CellCount
  HMIRuntime.trace ctrl.GetColumn(lCellIndex -1).Name & " "
  HMIRuntime.trace ctrl.GetRow(lIndex).CellText(lCellIndex) & "  "
 Next
 HMIRuntime.trace vbNewLine
Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "Row", vous écrivez p. ex. "objRow.CellCount" et non "objRow.RowCellCount".

Voir aussi

Objet Row (liste) (Page 233)

Méthode GetRowCollection

Fonction

Restitue la liste de tous les objets de ligne des Controls basés sur des tables en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetRowCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés de ICCAxDataRowCollection

ICCAxDataRowCollection se rapporte aux données Runtime. Les données sont en lecture seule. Elles ne peuvent être ni modifiées, ni complétées.

Les propriétés suivantes sont disponibles pour l'ICCAxDataRowCollection :

- Count - détermine le nombre de lignes dans la collection.
- Item - Accès à une ligne de la collection via le numéro de ligne. La numérotation va de "1" à "Count". Un objet Row est restitué.

Exemple

```
'VBS357
Dim ctrl
Dim coll
Dim lIndex
Dim lCellIndex
Set ctrl = ScreenItems("AlarmControl")
Set coll = ctrl.GetRowCollection
HMIRuntime.Trace "Number of message rows:" & coll.Count & vbCrLf
'enumerate and trace out row numbers
For lIndex = 1 To coll.Count
 HMIRuntime.Trace "Row: " & (ctrl.GetRow(lIndex).RowNumber) & "  "
 'enumerate and trace out column titles and cell texts
 For lCellIndex = 1 To ctrl.GetRow(lIndex).CellCount
  HMIRuntime.Trace ctrl.GetMessageColumn(lCellIndex -1).Name & " "
  HMIRuntime.Trace ctrl.GetRow(lIndex).CellText(lCellIndex) & "  "
 Next
 HMIRuntime.Trace vbNewLine
Next
```

Voir aussi

Objet Row (liste) (Page 233)

Méthode GetRulerBlock

Fonction

Restitue l'objet de bloc de WinCC RulerControl désigné par un nom ou un index en tant que type "ICCAxRulerBlock".

Syntaxe

```
Expression.GetRulerBlock(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom du bloc dans le RulerControl

Exemple

```
'VBS322
Dim ctrl
Dim objRulerBlock
Set ctrl = ScreenItems("RulerControl")
Set objRulerBlock = ctrl.GetRulerBlock(0)
objRulerBlock.Caption = "RulerBlock1"
Set objRulerBlock = ctrl.GetRulerBlock("Name")
objRulerBlock.Length = 10
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "RulerBlock", vous écrivez p. ex. "objRulerBlock.Caption" et non "objRulerBlock.BlockCaption".

Voir aussi

Objet RulerBlock (liste) (Page 234)

Méthode GetRulerBlockCollection

Fonction

Restitue la liste de tous les objets de bloc de WinCC RulerControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetRulerBlockCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS323
Dim ctrl
Dim coll
Dim rulerblock
Set ctrl = ScreenItems("RulerControl")
Set coll = ctrl.GetRulerBlockCollection
For Each rulerblock In coll
 rulerblock.Align = 1
 rulerblock.Length = 12
Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "RulerBlock", vous écrivez p. ex. "rulerblock.Align" et non "rulerblock.RulerBlockAlign".

Voir aussi

Objet RulerBlock (liste) (Page 234)

Méthode GetRulerColumn

Fonction

Restitue l'objet de colonne de WinCC RulerControl désigné par un nom ou un index en tant que type "ICCAxRulerColumn".

Syntaxe

```
Expression.GetRulerColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne dans le RulerControl.

Exemple

```
'VBS324
Dim ctrl
Dim objRulercol
Set ctrl = ScreenItems("RulerControl")
Set objRulercol = ctrl.GetRulerColumn("Name")
objRulercol.Sort = 0
Set objRulercol = ctrl.GetRulerColumn("ValueY")
objRulercol.Visible = FALSE
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "RulerColumn", vous écrivez p. ex. "objRulercol.Visible" et non "objRulercol.ColumnVisible".

Voir aussi

Objet RulerColumn (liste) (Page 234)

Méthode GetRulerColumnCollection

Fonction

Restitue la liste de tous les objets de colonne de WinCC RulerControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetRulerColumnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS325
Dim ctrl
Dim coll
Dim rulercol
Set ctrl = ScreenItems("RulerControl")
Set coll = ctrl.GetRulerColumnCollection
HMIRuntime.Trace "Number of ruler columns:" & coll.Count & vbCrLf
For Each rulercol In coll
 HMIRuntime.Trace rulercol.Index & vbCrLf
 HMIRuntime.Trace rulercol.Name & vbCrLf
 HMIRuntime.Trace rulercol.Sort & vbCrLf
 HMIRuntime.Trace rulercol.SortIndex & vbCrLf
Next
```

Voir aussi

Objet RulerColumn (liste) (Page 234)

Méthode GetRulerData

Fonction

Restitue la valeur de la courbe appelée à la position de règle.

Syntaxe

```
Expression.GetRulerData(ByVal RulerIndex As Long, pvValue As
Variant, Optional pvTimeStamp As Variant, Optional pvFlags As
Varian) Long
```

Expression

Impératif. Une expression qui restitue un objet de type "Trend".

Paramètre

Paramètre	Description
RulerIndex	0 = Règle
pvValue	Valeur de l'axe des X
pvTimeStamp	Instant ou valeur de l'axe Y
pvFlags	Qualitycode

Exemple

```
'VBS326
Dim ctrl
Dim objTrend
Dim objIOField1
Dim objIOField2
Dim value
Dim time
Set ctrl = ScreenItems( "Control1" )
Set objTrend = ctrl.GetTrend( "Trend 1" )
Set objIOField1 = ScreenItems( "I/O Field1" )
Set objIOField2 = ScreenItems( "I/O Field2" )
objTrend.GetRulerData 0, value, time
objIOField1.OutputValue = value
objIOField2.OutputValue = time
```

Méthode GetSelectedRow

Fonction

Restitue l'objet de ligne sélectionné des Controls basés sur des tables en tant que type "ICCAxDataRow".

Syntaxe

```
Expression.GetSelectedRow()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Exemple

```
'VBS358
Dim ctrl
 Dim lCellIndex
 Dim lCellCount
 Dim headingRow
 Dim selectedRow
 Set ctrl = ScreenItems("TableControl")
 Set headingRow = ctrl.GetRow(0)
 Set selectedRow = ctrl.GetSelectedRow
 lCellCount = headingRow.CellCount
 'enumerate and trace out column titles and cell texts
 For lCellIndex = 1 To lCellCount
  HMIRuntime.trace headingRow.CellText(lCellIndex) & ":  "
  HMIRuntime.trace selectedRow.CellText(lCellIndex)
    HMIRuntime.trace vbNewLine
 Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "Row", vous écrivez p. ex. "objRow.CellCount" et non "objRow.RowCellCount".

Voir aussi

Objet Row (liste) (Page 233)

Méthode GetSelectedRows

Fonction

Restitue les objets de ligne sélectionnés par sélection multiple des Controls basés sur des tables en tant que type "ICCAxDataRow".

Syntaxe

```
Expression.GetSelectedRows()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Exemple

```
'VBS359
Dim ctrl
 Dim lCellIndex
 Dim lCellCount
 Dim lRowIndex
 Dim lRowCount
 Dim headingRow
 Dim selectedRow
 Dim selectedRows
 Set ctrl = ScreenItems("TableControl")
 Set headingRow = ctrl.GetRow(0)
 Set selectedRows = ctrl.GetSelectedRows
 lCellCount = headingRow.CellCount
 lRowCount = selectedRows.Count
 'enumerate selected rows
 For lRowIndex = 1 To lRowCount
  Set selectedRow = selectedRows(lRowIndex)
  HMIRuntime.Trace "Row number: " & CStr(lRowIndex) & vbNewLine
  'enumerate and trace out column titles and cell texts
  For lCellIndex = 1 To lCellCount
   HMIRuntime.trace headingRow.CellText(lCellIndex) & ":  "
   HMIRuntime.trace selectedRow.CellText(lCellIndex)
     HMIRuntime.trace vbNewLine
  Next
 Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "Row", vous écrivez p. ex. "objRow.CellCount" et non "objRow.RowCellCount".

Voir aussi

Objet Row (liste) (Page 233)

Méthode GetStatisticAreaColumn

Fonction

Restitue l'objet de colonne de la fenêtre de la plage de statistiques de WinCC RulerControl désigné par un nom ou un index en tant que type "ICCAxRulerColumn".

Syntaxe

```
Expression.GetStatisticAreaColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne de la fenêtre de la plage de statistiques.

Exemple

```
'VBS327
Dim ctrl
Dim objStatAreaCol
Set ctrl = ScreenItems("RulerControl")
Set objStatAreaCol = ctrl.GetStatisticAreaColumn("DatasourceY")
objStatAreaCol.Visible = FALSE
Set objStatAreaCol = ctrl.GetStatisticAreaColumn("ValueY(LL)")
objStatAreaCol.Sort = 1
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "StatisticAreaColumn", vous écrivez p. ex. "objStatAreaCol.Visible" et non "objStatAreaCol.ColumnVisible".

Voir aussi

Objet StatisticAreaColumn (liste) (Page 235)

Méthode GetStatisticAreaColumnCollection

Fonction

Restitue la liste de tous les objets de colonne de la fenêtre de la plage de statistiques de WinCC RulerControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetStatisticAreaColumnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS328
Dim ctrl
Dim coll
Dim statcol
Set ctrl = ScreenItems("RulerControl")
Set coll = ctrl.GetStatisticAreaColumnCollection
HMIRuntime.Trace "Number of statistic Area columns:" & coll.Count & vbCrLf
For Each statcol In coll
 HMIRuntime.Trace statcol.Index & vbCrLf
 HMIRuntime.Trace statcol.Name & vbCrLf
 HMIRuntime.Trace statcol.Sort & vbCrLf
 HMIRuntime.Trace statcol.SortIndex & vbCrLf
Next
```

Voir aussi

Objet StatisticAreaColumn (liste) (Page 235)

Méthode GetStatisticResultColumn

Fonction

Restitue l'objet de colonne de la fenêtre de statistiques de WinCC RulerControl désigné par un nom ou un index en tant que type "ICCAxRulerColumn".

Syntaxe

```
Expression.GetStatisticResultColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne dans la fenêtre de statistiques.

Exemple

```
'VBS329
```

```
Dim ctrl
Dim objStatResCol
Set ctrl = ScreenItems("RulerControl")
Set objStatResCol = ctrl.GetStatisticResultColumn("MaxValue")
objStatResCol.Visible = FALSE
Set objStatResCol = ctrl.GetStatisticResultColumn("Average")
objStatResCol.Sort = 2
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "StatisticResultColumn", vous écrivez p. ex. "objStatResCol.Visible" et non "objStatResCol.ColumnVisible".

Voir aussi

Objet StatisticResultColumn (liste) (Page 236)

Méthode GetStatisticResultColumnCollection

Fonction

Restitue la liste de tous les objets de colonne de la fenêtre statistiques de WinCC RulerControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetStatisticAreaColumnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS330
Dim ctrl
Dim coll
Dim statcol
Set ctrl = ScreenItems("RulerControl")
Set coll = ctrl.GetStatisticResultColumnCollection
HMIRuntime.Trace "Number of statistic result columns:" & coll.Count & vbCrLf
For Each statcol In coll
 HMIRuntime.Trace statcol.Index & vbCrLf
 HMIRuntime.Trace statcol.Name & vbCrLf
 HMIRuntime.Trace statcol.Sort & vbCrLf
 HMIRuntime.Trace statcol.SortIndex & vbCrLf
Next
```

Voir aussi

Objet StatisticResultColumn (liste) (Page 236)

Méthode GetStatusbarElement

Fonction

Restitue l'élément de la barre d'état du Control désigné par un nom ou un index en tant que type "ICCAxStatusbarElement".

Syntaxe

```
Expression.GetStatusbarElement(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de l'élément de la barre d'état.

Exemple

```
'VBS331
Dim ctrl
Dim objStatusBar
Set ctrl = ScreenItems( "Control1" )
Set objStatusBar = ctrl.GetStatusbarElement(1)
objStatusBar.Visible = FALSE
Set objStatusBar = ctrl.GetStatusbarElement(3)
objStatusBar.Width = 10
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "StatusbarElement", vous écrivez p. ex. "objStatusBar.Visible" et non "objStatusBar.StatusbarElementVisible".

Voir aussi

Objet StatusbarElement (liste) (Page 237)

Méthode GetStatusbarElementCollection

Fonction

Restitue la liste de tous les éléments de la barre d'état du Control en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetStatusbarElementCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS332
Dim ctrl
Dim coll
Dim statelement
Set ctrl = ScreenItems.Item("Control1")
Set coll = ctrl.GetStatusbarElementCollection
HMIRuntime.Trace "Number of statusbar elements:" & coll.Count & vbCrLf
For Each statelement In coll
 HMIRuntime.Trace statelement.Name & vbCrLf
 HMIRuntime.Trace statelement.Width & vbCrLf
 HMIRuntime.Trace statelement.Text & vbCrLf
Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "StatusbarElement", vous écrivez p. ex. "statelement.Name" et non "statelement.StatusbarElementName".

Voir aussi

Objet StatusbarElement (liste) (Page 237)

Méthode GetTimeAxis

Fonction

Restitue l'objet de l'axe des temps de WinCC OnlineTrendControl désigné par un nom ou un index en tant que type "ICCAxTimeAxis".

Syntaxe

```
Expression.GetTimeAxis(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de l'axe des temps.

Exemple

```
'VBS333
Dim ctrl
Dim objTimeAxis
Set ctrl = ScreenItems("OnlineTrendControl")
Set objTimeAxis = ctrl.GetTimeAxis(1)
objTimeAxis.Visible = FALSE
Set objTimeAxis = ctrl.GetTimeAxis("axis 2")
objTimeAxis.Label = "Time axis 2"
objTimeAxis.DateFormat = "dd.MM.yy"
objTimeAxis.TimeFormat = "HH:mm:ss.ms"
objTimeAxis.RangeType = 2
objTimeAxis.BeginTime = "06.04.2010 9:33:18"
objTimeAxis.MeasurePoints = 100
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "TimeAxis", vous écrivez p. ex. "objTimeAx.Visible" et non "objTimeAx.TimeAxisVisible".

Voir aussi

Objet TimeAxis (liste) (Page 237)

Méthode GetTimeAxisCollection

Fonction

Restitue la liste de tous les objets d'axes des temps de WinCC OnlineTrendControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetTimeAxisCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS334
Dim ctrl
Dim objTrendWnd
Dim objTimeAxis1
Dim objTimeAxis2
Dim objTrend
Set ctrl = ScreenItems("OnlineTrendControl")
Set objTrendWnd = ctrl.GetTrendWindowCollection.AddItem("myWindow")
Set objTimeAxis1 = ctrl.GetTimeAxisCollection.AddItem("TimeAxis2010")
Set objTimeAxis2 = ctrl.GetTimeAxisCollection.AddItem("TimeAxis2011")
objTimeAxis1.TrendWindow = objTrendWnd.Name
objTimeAxis1.Label = "2010"
objTimeAxis1.RangeType = 1
objTimeAxis1.BeginTime = "01.01.2010 0:00:00"
objTimeAxis1.EndTime = "31.12.2010 11:59:59"
objTimeAxis2.TrendWindow = objTrendWnd.Name
objTimeAxis2.Label = "2011"
```

```
objTimeAxis2.RangeType = 1
objTimeAxis2.BeginTime = "01.01.2011 0:00:00"
objTimeAxis2.EndTime = "31.12.2011 11:59:59"
Set objTrend = ctrl.GetTrendCollection.AddItem("myTrend1")
objTrend.TrendWindow = objTrendWnd.Name
objTrend.TimeAxis = objTimeAxis1.Name
Set objTrend = ctrl.GetTrendCollection.AddItem("myTrend2")
objTrend.TrendWindow = objTrendWnd.Name
objTrend.TimeAxis = objTimeAxis2.Name
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "TimeAxis", vous écrivez p. ex. "objTimeAxis1.Label" et non "objTimeAxis1.TimeAxisLabel".

Voir aussi

Objet TimeAxis (liste) (Page 237)

Méthode GetTimeColumn

Fonction

Restitue l'objet de colonne des heures de WinCC OnlineTableControl désigné par un nom ou un index en tant que type "ICCAxTimeColumn".

Syntaxe

```
Expression.GetTimeColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne des heures.

Exemple

```
'VBS335
Dim ctrl
Dim objTimeCol
Set ctrl = ScreenItems("TableControl")
Set objTimeCol = ctrl.GetTimeColumn("Timecolumn1")
objTimeCol.ShowDate = FALSE
Set objTimeCol = ctrl.GetTimeColumn("Timecolumn2")
objTimeCol.Visible = FALSE
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "TimeColumn", vous écrivez p. ex. "objTimeColumn.ShowDate" et non "objTimeColumn.TimeColumnShowDate".

Voir aussi

Objet TimeColumn (liste) (Page 238)

Méthode GetTimeColumnCollection

Fonction

Restitue la liste de tous les objets de colonne des heures de WinCC OnlineTableControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetTimeColumnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS336
Dim ctrl
Dim objTimeCol1
Dim objTimeCol2
Dim coll
Dim timecol
Set ctrl = ScreenItems("TableControl")
Set objTimeCol1 = ctrl.GetTimeColumnCollection.AddItem("TimeColumn2010")
Set objTimeCol2 = ctrl.GetTimeColumnCollection.AddItem("TimeColumn2011")
objTimeCol1.Caption = "2010"
objTimeCol1.RangeType = 1
objTimeCol1.BeginTime = "01.01.2010 0:00:00"
objTimeCol1.EndTime = "31.12.2010 11:59:59"
objTimeCol2.Caption = "2011"
objTimeCol2.RangeType = 0
objTimeCol2.BeginTime = "01.01.2011 0:00:00"
objTimeCol2.TimeRangeFactor = 1
objTimeCol2.TimeRangeBase = 3600000
Set coll = ctrl.GetTimeColumnCollection
For Each timecol In coll
 timecol.Align = 1
 timecol.Length = 12
 timecol.BackColor = RGB(240,240,0)
 timecol.ForeColor = RGB(130,160,255)
Next
```

Voir aussi

Objet TimeColumn (liste) (Page 238)

Méthode GetToolbarButton

Fonction

Renvoie la fonction de touche de la barre d'outils du contrôle désigné par un nom ou un index en tant que type "ICCAxToolbarButton".

Syntaxe

```
Expression.GetToolbarButton(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la fonction de touche de la barre d'outils.

Exemple

```
'VBS337
Dim ctrl
Set ctrl = ScreenItems( "Control1" )
Dim toolbu
Set toolbu = ctrl.GetToolbarButton ("ShowHelp")
HMIRuntime.Trace "Name: " & toolbu.Name & vbCrLf
HMIRuntime.Trace "Index: " & toolbu.Index & vbCrLf
HMIRuntime.Trace "Hotkey: " & toolbu.HotKey & vbCrLf
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "ToolbarButton", vous écrivez p. ex. "toolbu.Index" et non "toolbu.ToolbarButtonIndex".

Voir aussi

Objet ToolbarButton (liste) (Page 239)

Méthode GetToolbarButtonCollection

Fonction

Restitue la liste de toutes les fonction de touche de la barre d'outils du Control en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetToolbarButtonCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les méthodes suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS338
Dim ctrl
Dim coll
Dim toolbu
Set ctrl = ScreenItems( "Control1" )
Set coll = ctrl.GetToolbarButtonCollection
HMIRuntime.Trace "Number of toolbar buttons:" & coll.Count & vbCrLf
For Each toolbu In coll
 HMIRuntime.Trace toolbu.Name & vbCrLf
 HMIRuntime.Trace "Hotkey: " & toolbu.HotKey & vbCrLf
 HMIRuntime.Trace "Authorization: " & toolbu.PasswordLevel & vbCrLf
Next
```

Voir aussi

Objet ToolbarButton (liste) (Page 239)

Méthode GetTrend

Fonction

Restitue l'objet de courbe de WinCC OnlineTrendControl ou WinCC FunctionTrendControl désigné par un nom ou un index en tant que type "ICCAxTrend" ou "ICCAxFunctionTrend".

Syntaxe

```
Expression.GetTrend(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la courbe.

Exemple

```
'VBS339
Dim ctrl
Dim objTrend
Set ctrl = ScreenItems("OnlineTrendControl")
Set objTrend = ctrl.GetTrend( "Trend 1" )
objTrend.PointStyle = 1
objTrend.LineWidth = 4
Set objTrend = ctrl.GetTrend(2)
objTrend.Provider = 1
objTrend.TagName = "Archive\ArchiveTag2"
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "Trend", vous écrivez p. ex. "objTrend.PointStyle" et non "objTrend.TrendPointStyle".

Voir aussi

Objet Trend (liste) (Page 240)

Méthode GetTrendCollection

Fonction

Restitue la liste de tous les objets de courbe de WinCC OnlineTrendControl ou de WinCC FunctionTrendControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetTrendCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS340
Dim ctrl
Dim objTrendWnd
Dim objTimeAxis
Dim objValAxis
Dim objTrend
Set ctrl = ScreenItems("OnlineTrendControl")
Set objTrendWnd = ctrl.GetTrendWindowCollection.AddItem("myWindow")
Set objTimeAxis = ctrl.GetTimeAxisCollection.AddItem("myTimeAxis")
Set objValAxis = ctrl.GetValueAxisCollection.AddItem("myValueAxis")
objTimeAxis.TrendWindow = objTrendWnd.Name
objValAxis.TrendWindow = objTrendWnd.Name
Set objTrend = ctrl.GetTrendCollection.AddItem("myTrend1")
objTrend.Provider = 1
objTrend.TagName = "Archive\ArchiveTag1"
objTrend.TrendWindow = objTrendWnd.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValAxis.Name
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "Trend", vous écrivez p. ex. "objTrend.TagName" et non "objTrend.TrendTagName".

Voir aussi

Objet Trend (liste) (Page 240)

Méthode GetTrendWindow

Fonction

Restitue l'objet de fenêtre de courbe de WinCC OnlineTrendControl ou de WinCC FunctionTrendControl désigné par un nom ou un index en tant que type "ICCAxTrendWindow".

Syntaxe

```
Expression.GetTrendWindow(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la fenêtre de courbe.

Exemple

```
'VBS341
Dim ctrl
Dim objTrendWnd
Set ctrl = ScreenItems("OnlineTrendControl")
Set objTrendWnd = ctrl.GetTrendWindow(1)
objTrendWnd.Visible = FALSE
Set objTrendWnd = ctrl.GetTrendWindow("trend window 2")
objTrendWnd.VerticalGrid = TRUE
```

```
objTrendWnd.FineGrid = TRUE
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "TrendWindow", vous écrivez p. ex. "objTrendWnd.Visible" et non "objTrendWnd.TrendWindowVisible".

Voir aussi

Objet TrendWindow (liste) (Page 242)

Méthode GetTrendWindowCollection

Fonction

Restitue la liste de tous les objets de fenêtre courbe de WinCC OnlineTrendControl ou de WinCC FunctionTrendControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetTrendWindowCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS342
Dim ctrl
Dim objTrendWnd
Dim objTimeAxis
Dim objValAxis
Set ctrl = ScreenItems("OnlineTrendControl")
Set objTrendWnd = ctrl.GetTrendWindowCollection.AddItem("myWindow")
Set objTimeAxis = ctrl.GetTimeAxisCollection.AddItem("myTimeAxis")
Set objValAxis = ctrl.GetValueAxisCollection.AddItem("myValueAxis")
objTimeAxis.TrendWindow = objTrendWnd.Name
objValAxis.TrendWindow = objTrendWnd.Name
```

Voir aussi

Objet TrendWindow (liste) (Page 242)

Méthode GetValueAxis

Fonction

Restitue l'objet de l'axe des valeurs de WinCC OnlineTrendControl désigné par un nom ou un index en tant que type "ICCAxValueAxis".

Syntaxe

```
Expression.GetValueAxis(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de l'axe des valeurs.

Exemple

```
'VBS343
Dim ctrl
Dim objValAxis
Set ctrl = ScreenItems("OnlineTrendControl")
```

```
Set objValAxis = ctrl.GetValueAxis(1)
objValAxis.Visible = FALSE
Set objValAxis = ctrl.GetValueAxis("axis 2")
objValAxis.Label = "Value axis 2"
objValAxis.ScalingType = 0
objValAxis.Precisions = 2
objValAxis.AutoRange = TRUE
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "ValueAxis", vous écrivez p. ex. "objValueAx.Visible" et non "objValueAx.ValueAxisVisible".

Voir aussi

Objet ValueAxis (liste) (Page 243)

Méthode GetValueAxisCollection

Fonction

Restitue la liste de tous les objets de l'axe des valeurs de WinCC OnlineTrendControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetValueAxisCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS344
Dim ctrl
Dim objTrendWnd
Dim objValAxis1
Dim objValAxis2
Dim objTrend
Set ctrl = ScreenItems("OnlineTrendControl")
Set objTrendWnd = ctrl.GetTrendWindowCollection.AddItem("myWindow")
Set objValAxis1 = ctrl.GetValueAxisCollection.AddItem("myValueAxis1")
Set objValAxis2 = ctrl.GetValueAxisCollection.AddItem("myValueAxis2")
objValAxis1.TrendWindow = objTrendWnd.Name
objValAxis1.Label = "Value1"
objValAxis2.TrendWindow = objTrendWnd.Name
objValAxis2.inTrendColor = TRUE
Set objTrend = ctrl.GetTrendCollection.AddItem("myTrend1")
objTrend.TrendWindow = objTrendWnd.Name
objTrend.ValueAxis = objValAxis1.Name
Set objTrend = ctrl.GetTrendCollection.AddItem("myTrend2")
objTrend.TrendWindow = objTrendWnd.Name
objTrend.ValueAxis = objValAxis2.Name
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "ValueAxis", vous écrivez p. ex. "objValueAxis1.Label" et non "objValueAxis1.ValueAxisLabel".

Voir aussi

Objet ValueAxis (liste) (Page 243)

Méthode GetValueColumn

Fonction

Restitue l'objet de colonne des valeurs de WinCC OnlineTableControl désigné par un nom ou un index en tant que type "ICCAxValueColumn".

Syntaxe

```
Expression.GetValueColumn(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de la colonne des valeurs dans le OnlineTableControl.

Exemple

```
'VBS345
Dim ctrl
Dim objValueColumn
Set ctrl = ScreenItems("TableControl")
Set objValueColumn = ctrl.GetValueColumn("Valuecolumn1")
objValueColumn.Precisions = 4
Set objValueColumn = ctrl.GetValueColumn(2)
objValueColumn.ExponentialFormat = TRUE
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "ValueColumn", vous écrivez p. ex. "objValueColumn.Precisions" et non "objValueColumn.ValueColumnPrecisions".

Voir aussi

Objet ValueColumn (liste) (Page 244)

Méthode GetValueColumnCollection

Fonction

Restitue la liste de tous les objets de colonne des valeurs de WinCC OnlineTableControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetValueColulmnCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS346
Dim ctrl
Dim objValCol1
Dim objValCol2
Dim coll
Dim valcol
Set ctrl = ScreenItems("TableControl")
Set objValCol1 = ctrl.GetValueColumnCollection.AddItem("ValueColumn1")
Set objValCol2 = ctrl.GetValueColumnCollection.AddItem("ValueColumn2")
objValCol1.Caption = "Value Archive"
objValCol1.Provider = 1
objValCol1.TagName = "ProcessValueArchive\arch1"
objValCol1.TimeColumn = "TimeColumn1"
objValCol2.Caption = "Value Tag"
objValCol2.Provider = 2
objValCol2.TagName = "tagxx"
objValCol2.TimeColumn = "TimeColumn2"
Set coll = ctrl.GetValueColumnCollection
For Each valcol In coll
 valcol.Align = 2
 valcol.Length = 10
 valcol.AutoPrecisions = TRUE
Next
```

Voir aussi

Objet ValueColumn (liste) (Page 244)

Méthode GetXAxis

Fonction

Restitue l'objet de l'axe X de WinCC FunctionTrendControl désigné par un nom ou un index en tant que type "ICCAxValueAxis".

Syntaxe

```
Expression.GetXAxis(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de l'axe X.

Exemple

```
'VBS347
Dim ctrl
Dim objXAx
Set ctrl = ScreenItems("FunctionTrendControl")
Set objXAx = ctrl.GetXAxis(1)
objXAx.Visible = FALSE
Set objXAx = ctrl.GetXAxis("axis 2")
objXAx.Label = "X axis 2"
objXAx.ScalingType = 0
objXAx.Precisions = 2
objXAx.Color = RGB(109,109,109)
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "XAxis", vous écrivez p. ex. "objXAx.Visible" et non "objXAx.XAxisVisible".

Voir aussi

Objet XAxis (liste) (Page 245)

Méthode GetXAxisCollection

Fonction

Restitue la liste de tous les objets d'axe X de WinCC FunctionTrendControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetXAxisCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS348
Dim ctrl
Dim objXAxis1
Dim objXAxis2
Dim coll
Dim axes
Set ctrl = ScreenItems("FunctionTrendControl")
Set objXAxis1 = ctrl.GetXAxisCollection.AddItem("myXAxis1")
objXAxis1.Label = "temperature"
Set objXAxis2 = ctrl.GetXAxisCollection.AddItem("myXAxis2")
objXAxis2.Label = "pressure"
Set coll = ctrl.GetXAxisCollection
```

```
HMIRuntime.Trace "Number of XAxis:" & coll.Count & vbCrLf
For Each axes In coll
 HMIRuntime.Trace axes.Name & vbCrLf
 HMIRuntime.Trace axes.Label & vbCrLf
Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "XAxis", vous écrivez p. ex. "objXAxis1.Label" et non "objXAxis1.XAxisLabel".

Voir aussi

Objet XAxis (liste) (Page 245)

Méthode GetYAxis

Fonction

Restitue l'objet de l'axe Y de WinCC FunctionTrendControl désigné par un nom ou un index en tant que type "ICCAxValueAxis".

Syntaxe

```
Expression.GetYAxis(ByVal vIndex As Variant)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

VARIANT

Paramètre	Description
vIndex	Index ou nom de l'axe Y.

Exemple

```
'VBS349
Dim ctrl
Dim objYAx
```

```
Set ctrl = ScreenItems("FunctionTrendControl")
Set objYAx = ctrl.GetYAxis(1)
objYAx.Visible = FALSE
Set objYAx = ctrl.GetYAxis("axis 2")
objYAx.Label = "Y axis 2"
objYAx.Align = 0
objYAx.Precisions = 3
objYAx.EndValue = 90.000
objYAx.BeginValue = 10.000
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "YAxis", vous écrivez p. ex. "objYAx.Visible" et non "objYAx.YAxisVisible".

Voir aussi

Objet YAxis (liste) (Page 245)

Méthode GetYAxisCollection

Fonction

Restitue la liste de tous les objets d'axe Y de WinCC FunctionTrendControl en tant que type "ICCAxCollection".

Syntaxe

```
Expression.GetYAxisCollection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Propriétés et fonctions de l'ICCAxCollection

Les propriétés suivantes sont disponibles pour l'ICCAxCollection :

- Count
- Item

Les fonctions suivantes sont disponibles pour l'ICCAxCollection :

- AddItem(vName) As Object
- RemoveItem(vIndex)

Exemple

```
'VBS350
Dim ctrl
Dim objYAxis1
Dim objYAxis2
Dim coll
Dim axes
Set ctrl = ScreenItems("FunctionTrendControl")
Set objYAxis1 = ctrl.GetXAxisCollection.AddItem("myYAxis1")
objYAxis1.Label = "temperature"
Set objYAxis2 = ctrl.GetXAxisCollection.AddItem("myYAxis2")
objYAxis2.Label = "pressure"
Set coll = ctrl.GetYAxisCollection
HMIRuntime.Trace "Number of YAxis:" & coll.Count & vbCrLf
For Each axes In coll
 HMIRuntime.Trace axes.Name & vbCrLf
 HMIRuntime.Trace axes.Label & vbCrLf
Next
```

Remarque

Si vous accédez aux propriétés à l'aide de l'objet de liste, vous n'avez pas besoin de saisir le nom de la liste.

Pour la liste "YAxis", vous écrivez p. ex. "objYAxis1.Label" et non "objYAxis1.YAxisLabel".

Voir aussi

Objet YAxis (liste) (Page 245)

1.14.5.4 Méthodes H à M

Méthode HideAlarm

Fonction

Exécute la fonction de touche "Masquer l'alarme" de l'AlarmControl.

Syntaxe

```
Expression.HideAlarm()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode InsertData

Fonction

Ajoute des données à la courbe appelée.

Syntaxe

```
Expression.InsertData(dblAxisX As Variant, dblAxisY As Variant)
```

Expression

Impératif. Une expression qui restitue un objet de type "Trend".

Paramètre

Paramètre	Description
dblAxisX	Valeur de l'axe des X
dblAxisY	Valeur de l'axe des Y

Exemple

```
'VBS300
Dim lngFactor
Dim dblAxisX
Dim dblAxisY
Dim objTrendControl
Dim objTrend
Set objTrendControl = ScreenItems("Control1")
Set objTrend = objTrendControl.GetTrend("Trend 1")
For lngFactor = -100 To 100
dblAxisX = CDbl(lngFactor * 0.02)
dblAxisY = CDbl(dblAxisX * dblAxisX + 2 * dblAxisX + 1)
objTrend.InsertData dblAxisX, dblAxisY
Next
```

Méthode Item

Fonction

Extrait un objet d'une collection et permet d'y accéder via l'Index.

Description pour l'objet DataItem

L'accès s'effectue par le nom sous lequel la valeur a été ajoutée dans la liste. L'accès individuel par l'index n'est pas recommandé étant donné que l'index change à chaque fois que des valeurs sont ajoutées et supprimées.

Syntaxe

```
Expression.Item()
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "Screens", "Layers" (ou "Tags").

Remarque

L'étendue de la fonction est restreint pour "Tags"! Les méthodes standard get_Count et get_NewEnum font défaut, rendant ainsi impossible l'accès via Index et le comptage de toutes les variables.

Paramètres

VARIANT

Exemple

L'exemple suivant affiche le nom de tous les objets contenus dans la vue "NewPDL1" :

```
'VBS99
Dim objScreen
Dim objScrItem
Dim lngIndex
Dim lngAnswer
Dim strName
lngIndex = 1
Set objScreen = HMIRuntime.Screens("NewPDL1")
For lngIndex = 1 To objScreen.ScreenItems.Count
'
'The objects will be indicate by Item()
strName = objScreen.ScreenItems.Item(lngIndex).ObjectName
Set objScrItem = objScreen.ScreenItems(strName)
lngAnswer = MsgBox(objScrItem.ObjectName, vbOKCancel)
```

```
If vbCancel = lngAnswer Then Exit For
Next
```

Voir aussi

Objet ScreenItems (Liste) (Page 136)
Objet ScreenItem (Page 132)
Objet Tags (Liste) (Page 147)
Objet d'alarmes (liste) (Page 118)
Objet ProcessValues (liste) (Page 131)

Méthode LockAlarm

Fonction

Exécute la fonction de touche "Verrouiller l'alarme" de l'AlarmControl.

Syntaxe

```
Expression.LockAlarm()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode LoopInAlarm

Fonction

Exécute la fonction de touche "Loop in Alarm" de l'AlarmControl.

Syntaxe

```
Expression.LoopInAlarm()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode MoveAxis

Fonction

Exécute la fonction de touche "Déplacer la zone des axes" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.MoveAxis()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

MoveRuler

Fonction

Déplace la règle d'une distance définie à partir d'un point de référence indiqué.

Syntaxe

```
Expression.MoveRuler( RulerIndex As Long, RulerMoveRef As Long,
MoveDistance As Long, Optional vTrendWindow As Variant )
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

Paramètres	Description
RulerIndex	Indique la règle qui doit être déplacée : 0 = Règle 1 = Règle au début de la zone de statistiques 2 = Règle à la fin de la zone de statistiques
RulerMoveRef	Indique le point de référence sur lequel s'oriente le troisième paramètre "MoveDistance" : 0 = Position de début de l'axe des temps 1 = Position actuelle de la règle 2 = Position de fin de l'axe des temps
MoveDistance	Nombre de pixels dont la règle est déplacée à partir du point de référence "RulerMoveRef" indiqué.
vTrendWindow	Paramètre optionnel lors de l'utilisation de plusieurs fenêtres de courbes indépendantes. Détermine la fenêtre de courbes dans laquelle la règle sera déplacée. Si ce paramètre n'est pas précisé, la règle sera déplacée dans toutes les fenêtres de courbes.

Valeur de retour

La fonction renvoie la nouvelle position de la règle.

Exemple

Tableau 1-1 Déplacer la règle de 10 pixels vers la gauche

```
'VBS367
Sub OnClick(ByVal Item)
Dim ctrl
Set ctrl = ScreenItems.Item("Control1")
ctrl.MoveRuler (0, 1, -10)
End Sub
```

Dans cet exemple, la règle est déplacée de -10 pixels à partir du point de référence 1 (position actuelle de la règle). En résultat, la règle se trouve donc 10 pixels à gauche de sa position de départ.

Exemple

Tableau 1-2 Déplacer la règle de 10 pixels vers la droite

```
'VBS368
Sub OnClick(ByVal Item)
Dim ctrl
Set ctrl = ScreenItems.Item("Control1")
ctrl.MoveRuler (0, 1, 10)
End Sub
```

Dans cet exemple, la règle est déplacée de 10 pixels à partir du point de référence 1 (position actuelle de la règle). En résultat, la règle se trouve donc 10 pixels à droite de sa position de départ.

Exemple

Tableau 1-3 Placer la règle à la fin lors de l'ouverture de la vue

```
'VBS369
Sub OnOpen()
Dim ctrl
Set ctrl = ScreenItems.Item("Control1")
ctrl.MoveRuler (0, 2, 0)
End Sub
```

Dans cet exemple, la règle est déplacée de 0 pixel à partir du point de référence 2 (position de fin de l'axe des temps). En résultat, la règle se trouve donc sur la position de fin de l'axe des temps.

Exemple

Tableau 1-4 Déterminer la position actuelle de la règle

```
'VBS370
Sub OnClick(ByVal Item)
Dim ctrl
Set ctrl = ScreenItems.Item("Control1")
Dim pos
pos = ctrl.MoveRuler (0, 1, 0)
HmiRuntime.Trace "RulerPosition=" & pos & vbCrLf
End Sub
```

Dans cet exemple, la règle est déplacée de 0 pixel à partir du point de référence 1 (position actuelle de la règle). En résultat, la règle se trouve donc toujours sur sa position de départ. La position de la règle est renvoyée comme valeur de retour.

Méthode MoveToFirst

Fonction

Exécute la fonction de touche "Première ligne" du contrôle.

Syntaxe

```
Expression.MoveToFirst()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode MoveToFirstLine

Fonction

Exécute la fonction de touche "Première alarme" de l'AlarmControl.

Syntaxe

```
Expression.MoveToFirstLine()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToFirstPage

Fonction

Exécute la fonction de touche "Première page" de l'AlarmControl.

Syntaxe

```
Expression.MoveToFirstPage()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToLast

Fonction

Exécute la fonction de touche "Dernier enregistrement" du contrôle.

Syntaxe

```
Expression.MoveToLast()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToLastLine

Fonction

Exécute la fonction de touche "Dernière alarme" de l'AlarmControl.

Syntaxe

```
Expression.MoveToLastLine()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToLastPage

Fonction

Exécute la fonction de touche "Dernière page" de l'AlarmControl.

Syntaxe

```
Expression.MoveToLastPage()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToNext

Fonction

Exécute la fonction de touche "Enregistrement suivant" du contrôle.

Syntaxe

```
Expression.MoveToNext()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToNextLine

Fonction

Exécute la fonction de touche "Alarme suivante" de l'AlarmControl.

Syntaxe

```
Expression.MoveToNextLine()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToNextPage

Fonction

Exécute la fonction de touche "Page suivante" de l'AlarmControl.

Syntaxe

```
Expression.MoveToNextPage()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToPrevious

Fonction

Exécute la fonction de touche "Enregistrement précédent" du contrôle.

Syntaxe

```
Expression.MoveToPrevious()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToPreviousLine

Fonction

Exécute la fonction de touche "Alarme précédente" de l'AlarmControl.

Syntaxe

```
Expression.MoveToPreviousLine()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode MoveToPreviousPage

Fonction

Exécute la fonction de touche "Page précédente" de l'AlarmControl.

Syntaxe

```
Expression.MoveToPreviousPage()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

1.14.5.5 Méthodes N à R

Méthode NextColumn

Fonction

Exécute la fonction de touche "Colonne suivante" du OnlineTableControl.

Syntaxe

```
Expression.MoveToNextLine()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode NextTrend

Fonction

Exécute la fonction de touche "Courbe suivante" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.NextTrend()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode OneToOneView

Fonction

Exécute la fonction de touche "Vue d'origine" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.OneToOneView()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

VARIANT

Méthode PasteRows

Fonction

Exécute la fonction de touche "Insérer lignes" du UserArchiveControl.

Syntaxe

```
Expression.PasteRows()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode PreviousColumn

Fonction

Exécute la fonction de touche "Colonne précédente" du OnlineTableControl.

Syntaxe

```
Expression.PreviousColumn()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode PreviousTrend

Fonction

Exécute la fonction de touche "Courbe précédente" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.PreviousTrend()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode Print

Fonction

Exécute la fonction de touche "Imprimer" du contrôle.

Syntaxe

```
Expression.Print()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode QuitHorn

Fonction

Exécute la fonction de touche "Acquittement de l'avertisseur central" de l'AlarmControl.

Syntaxe

```
Expression.QuitHorn()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode QuitSelected

Fonction

Exécute la fonction de touche "Acquittement individuel" de l'AlarmControl.

Syntaxe

```
Expression.QuitSelected()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode QuitVisible

Fonction

Exécute la fonction de touche "Acquittement groupé" de l'AlarmControl.

Syntaxe

```
Expression.QuitVisible()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Methode Read

Description pour l'objet Tag

Lit l'état d'une variable (objet "Tag") immédiatement après appel. L'objet "Tag" est simultanément complété par les valeurs lues. Lors de la lecture des variables, la valeur, le code qualité et l'horodatage de la variable sont définis. Il est possible de déterminer via la propriété "LastError" si l'appel a correctement fonctionné.

Les propriétés "Name", "Serverprefix" et "Tagprefix" ne sont alors pas modifiées.

Si la valeur de la variable a correctement été lue, les propriétés de l'objet "Tag" reçoivent les valeurs suivantes :

Propriété	Affectation
value	Valeur de la variable
Nom	Nom de variable (reste inchangé)
QualityCode	Niveau de qualité
TimeStamp	Horodatage actuel de la variable
LastError	0
ErrorDescription	" "

Si la valeur de la variable n'a pas correctement été lue, les propriétés de l'objet "Tag" reçoivent les valeurs suivantes :

Propriété	Affectation
value	VT_Empty
Nom	Nom de variable (reste inchangé)
QualityCode	Bad Out of Service
TimeStamp	0
LastError	Code d'erreur pour l'opération de lecture
ErrorDescription	Description de l'erreur pour LastError

Remarque

Vous trouverez un récapitulatif des Codes qualité possibles en consultant WinCC Information System sous le mot repère "Communication" > "Diagnostic" ou "Communication" > "Codes qualité".

Syntaxe

```
Expression.Read([Readmode])
```

Expression

Impératif. Une expression qui restitue un objet Tag. La valeur de retour de la méthode Read est la valeur de la variable lue.

Paramètres

Le paramètre optionnel "Readmode" permet de distinguer deux types de lecture :

Paramètres	Description
0	La valeur des variables est lue dans la mémoire image du process (cache). 0 est la valeur par défaut.
1	La valeur des variables est directement lue dans l'AS ou le canal (direct).

Si le paramètre "Readmode" est omis, la lecture se fait en standard dans la mémoire image du process. La valeur de retour de la méthode Read est la valeur de variable lue en tant que VARIANT.

Lire dans la mémoire image du process

Lors de la lecture dans la mémoire image du process, la variable est connectée et, à partir de là, constamment demandée de façon cyclique dans le contrôleur. Le cycle de connexion dépend alors du Trigger configuré. La valeur est lue depuis l'image de variable de WinCC. En

cas de désélection de vue, les variables sont à nouveau déconnectées. L'appel se caractérise de la façon suivante :

- La valeur est lue à partir de la mémoire-image de WinCC.
- Comparé à la lecture directe, l'appel est plus rapide (sauf pour le premier appel : le premier appel dure en principe plus longtemps étant donné que la valeur doit être lue dans l'AS et puis connectée.)
- La durée de l'appel ne dépend ni de la charge du bus, ni de l'AS

Comportement dans les actions avec Trigger à variables

Toutes les variables contenues dans le déclencheur de variable sont déjà connues lors de la sélection de vues et sont connectées avec le temps de surveillance spécifié. Etant donné que toutes les variables sont exigées en une fois, l'optimisation la meilleure possible peut être obtenue du canal. Dans une action, si une variable est sélectionnée avec Read alors qu'elle se trouve dans le Trigger, la valeur est déjà disponible et est directement transmise à l'appel. Si une variable ne se trouvant pas dans le Trigger est sélectionnée, le système se comporte comme dans le cas d'un Trigger standard.

Comportement dans les actions avec Trigger cyclique

Lors du premier appel, la variable est connectée avec la moitié du temps de cycle. La valeur est alors présente à chaque nouvel appel.

Comportement dans les actions déclenchées par des événements

Lors du premier appel, la variable est connectée dans le mode "En cas de modification". Les variables de process qui sont connectées dans le mode "En cas de modification", correspondent à un contrat de lecture cyclique ayant un temps de cycle d'1s.

Lorsqu'un événement (clic de souris par exemple) est sélectionné de façon asynchrone avec une valeur, la variable est enregistrée dans la mémoire-image de variable. La variable est exigée cycliquement à partir de ce moment depuis l'AS, ce qui augmente la charge initiale. Pour éviter cette augmentation des sollicitations à la base, vous pouvez également lire la valeur de façon synchrone. L'appel synchrone occasionne certes une seule et unique fois une puissance charge de communication, mais la variable n'est pas enregistrée dans la mémoire image.

Lecture directe

Lors de la lecture directe, la valeur actuelle est restituée. La variable n'est pas connectée cycliquement, mais la valeur est exigée une seule fois depuis l'AS. Les propriétés de la lecture directe sont les suivantes :

- La valeur est lue explicitement dans l'AS.
- Elle prend plus de temps comparée à la lecture dans la mémoire image du process
- La durée de l'appel dépend entre autres de la charge du bus ainsi que de l'AS

Exemple

Lecture d'une variable directement dans l'AS ou dans le canal

```
'VBS100
Dim objTag
Dim vntValue
Set objTag = HMIRuntime.Tags("Tagname")
vntValue = objTag.Read(1)    'Read direct
MsgBox vntValue
```

Lecture d'une variable depuis la mémoire image du process

```
'VBS101
Dim objTag
Dim vntValue
Set objTag = HMIRuntime.Tags("Tagname")
vntValue = objTag.Read    'Read from cache
MsgBox vntValue
```

Description pour TagSet

L'objet TagSet offre la possibilité de lire plusieurs variables en un appel.

Le fonctionnement est pratiquement similaire à celui d'un objet Tag. Seules les divergences sont décrites ci-après.

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "TagSet".

Lire dans la mémoire image du process

L'objet TagSet offre l'avantage de pouvoir demander plusieurs variables avec un seul contrat de lecture. Les variables sont alors connectées ensemble dans la mémoire image du process, ce qui améliore les performances.

Lecture directe

Etant donné qu'il est possible de traiter plusieurs contrats de lecture en un appel, les performances sont meilleures que s'il s'agissait de plusieurs appels individuels.

Exemple

L'exemple suivant montre comment accepter puis lire des variables dans la liste TagSet.

```
'VBS174
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
group.Read
HMIRuntime.Trace "Motor1: " & group("Motor1").Value & vbNewLine
HMIRuntime.Trace "Motor2: " & group("Motor2").Value & vbNewLine
```

Si l'on positionne le paramètre optionnel "Readmode" sur 1, les variables du process ne sont pas connectées, mais directement lues dans l'AS ou dans le canal.

```
group.Read 1
```

Voir aussi

Exemple : Pour lire les valeurs variable (Page 799)

Exemple : Pour écrire les valeurs variable (Page 797)

Propriété LastError (Page 432)

Propriété ErrorDescription (Page 386)

Objet TagSet (Liste) (Page 149)

Objet Tag (Page 144)

Méthode Read Tags

Fonction

Exécute la fonction de touche "Lire variables" du UserArchiveControl.

Syntaxe

```
Expression.ReadTags()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode Refresh

Fonction

Redessine toutes les vues visibles.

Syntaxe

```
Expression.Refresh
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "Screens" ou "Screen".

Paramètres

--

Exemples

Le premier exemple impose que toutes les vues visibles soient immédiatement redessinées :

```
'VBS149
HMIRuntime.Screens.Refresh
```

Le deuxième exemple impose que la vue de base soit immédiatement redessinée :

```
'VBS150
HMIRuntime.Screens(1).Refresh
```

Voir aussi

Objet Screen (Page 138)
Objet Screens (Liste) (Page 141)
Objet HMIRuntime (Page 125)

Méthode Remove

Description pour TagSet

Supprime une variable de la liste TagSet. La variable peut être supprimée par le nom ou par une référence à un objet Tag.

Syntaxe

```
Expression.Remove [Tag]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "TagSet".

Paramètres

VARIANT

Paramètres	Description
Tag	Nom d'une variable WinCC ou référence à un objet Tag qu'il faut supprimer de la liste.

Exemple

L'exemple suivant montre comment plusieurs variables sont acceptées dans la liste TagSet et comment une variable en est ensuite supprimée.

```
'VBS175
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
group.Remove "Motor1"
```

Description pour l'objet DataSet

Supprime d'une liste l'élément spécifié dans le paramètre "Name".

Syntaxe

```
Expression.Remove [Name]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "DataSet".

Paramètres

VARIANT

Paramètres	Description
Nom	Nom de l'objet qui doit être supprimé de la liste.

Exemple

L'exemple montre comment l'objet "motor1" est supprimé de la liste.

```
'VBS166
HMIRuntime.DataSet.Remove("motor1")
```

Description pour les objets Logging, AlarmLogs, DataLogs

La méthode supprime du projet Runtime les segments d'archive auparavant transférés.

Les segments d'archive supprimés avec la méthode "Remove" sont supprimés du répertoire Common Archiving du projet.

L'appel peut durer un certain temps, selon les données archivées. Ce qui peut bloquer le traitement des scripts suivants. Vous pouvez éviter un blocage des actions dans la vue en lançant l'appel dans une action de Global Scripting, en lançant par exemple l'action par une variable de déclenchement.

La séparation et la suppression des archives créent une charge sur la CPU. Ce qui a des répercussions sur les performances.

Remarque

L'appel de la méthode "Remove" n'est actuellement possible qu'au niveau du serveur. Il existe cependant un exemple qui montre comment la méthode peut être lancée depuis le client, sur le serveur.

En cas de redondance : les archives à nouveau transférées ne sont supprimées avec la méthode "Remove" que sur l'ordinateur où la méthode a été appelée.

Syntaxe

Objets Logging, AlarmLogs

```
Expression.Remove [TimeFrom] [TimeTo] [TimeOut] [ServerPrefix]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "Logging" ou "AlarmLogs".

Objet DataLogs

```
Expression.Remove [TimeFrom] [TimeTo] [TimeOut] [Type] [ServerPrefix]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "DataLogs".

Paramètres

TimeFrom

Heure à partir de laquelle les archives doivent être supprimées.
Il est également possible de spécifier le format de l'heure sous forme abrégée, comme décrit au paragraphe "Format de l'heure".

TimeTo

Date/Heure jusqu'à laquelle les segments d'archive doivent être supprimés.
Le format de l'heure peut également être spécifié sous forme abrégée, comme décrit au paragraphe "Format de l'heure".

Timeout

Timeout en millisecondes.
Si vous tapez la valeur "-1", l'attente est sans fin (infinite). Si vous tapez "0", il n'y a aucun temps d'attente.

Type

Type d'archive.
Ce paramètre ne peut (en option) être utilisé que pour supprimer des segments d'archive de Tag Logging.
Les valeurs suivantes peuvent être saisies :

Valeur affectée	Type	Description
1	hmiDataLogFast	Tag Logging Fast data
2	hmiDataLogSlow	Tag Logging Slow data
3	hmiDataLogAll	Tag Logging Fast and Slow data

ServerPrefix

Réservé aux versions futures.

Valeur en retour

La méthode renvoie un message d'erreur si une erreur s'est produite en effaçant les segments d'archive. Vous trouverez plus d'informations à ce sujet dans la rubrique "Messages d'erreur dans la zone Base de données".

Format de l'heure

Le format de l'heure est définit comme suit : YYYY-MM-DD hh:mm:ss, où YYYY désigne l'année, MM le mois, DD le jour, hh l'heure, mm les minutes et ss les secondes. 11 heures passées de 2 minutes et d'une seconde le 26 juillet 2004 sera par exemple représenté comme suit : 2004-07-26 11:02:01.

La date et l'heure peuvent être abrégées pour les paramètres "TimeFrom" et "TimeTo". Il n'est pas nécessaire de couvrir tous les champs du format. On entend par abréviation qu'il est possible de supprimer un ou plusieurs paramètres pour indiquer la date et l'heure, en commençant par la valeur des secondes. L'indication peut alors p. ex. être donnée dans le format "YYYY-MM" ou "YYYY-MM-DD hh". Si vous indiquez "TimeFrom" = "2004-09" et

"TimeTo" = "2004-10-04", tous les segments d'archive sont transférés du mois de septembre 2004 jusqu'au 4 octobre inclu.

Exemple

Dans l'exemple suivant, les segments d'archive ultérieurement (re)transférés sont supprimés pour une période déterminée, et la valeur de retour est affichée en tant que Trace.

```
'VBS182
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Remove("2004-08-22","2004-09-22",-1) &
vbNewLine
```

Dans l'exemple suivant, tous les segments d'archive ultérieurement (re)transférés sont supprimés et la valeur de retour est affichée en tant que Trace.

```
'VBS183
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Remove("","",-1) & vbNewLine
```

Voir aussi

Messages d'erreur dans la zone Bases de données (Page 790)

Exemple : Pour démarrer une action sur le serveur (objet Logging) (Page 805)

Objet Logging (Page 129)

Objet DataSet (Liste) (Page 123)

Objet DataLogs (Page 122)

Objet AlarmLogs (Page 119)

Objet TagSet (Liste) (Page 149)

Méthode RemoveAll

Description pour TagSet

Supprime toutes les variables d'une liste TagSet.

Syntaxe

```
Expression.RemoveAll
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "TagSet".

Paramètres

--

Exemple

L'exemple suivant montre comment plusieurs variables sont acceptées dans la liste TagSet et comment toutes en sont ensuite supprimées.

```
'VBS176
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
group.RemoveAll
```

Description pour l'objet DataSet

Supprime d'une liste DataSet toutes les valeurs ou les références sur un objet.

Syntaxe

```
Expression.RemoveAll
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "DataSet".

Paramètres

--

Exemple

L'exemple montre comment tous les objets sont supprimés de la liste.

```
'VBS167
HMIRuntime.DataSet.RemoveAll
```

Voir aussi

Objet DataSet (Liste) (Page 123)

Objet TagSet (Liste) (Page 149)

Objet Tag (Page 144)

Méthode RemoveData

Fonction

Supprime des données à la courbe appelée.

Syntaxe

```
Expression.RemoveData
```

Expression

Impératif. Une expression qui restitue un objet de type "Trend".

Exemple

```
'VBS310
Dim objTrendControl
Dim objTrend
Set objTrendControl = ScreenItems("Control1")
Set objTrend = objTrendControl.GetTrend("Trend 1")
objTrend.RemoveData
```

Méthode Restore

Description pour les objets Logging, AlarmLogs, DataLogs

La méthode ajoute au projet Runtime des segments d'archive transférés.

Lorsque les segments d'archive sont transférés, ils sont copiés dans le répertoire Common Archiving du projet. Il est donc nécessaire d'avoir une capacité de mémoire suffisante.

L'appel peut durer un certain temps, selon les données archivées. Ce qui peut bloquer le traitement des scripts suivants. Vous pouvez éviter un blocage des actions dans la vue en lançant l'appel dans une action de Global Scripting, en lançant par exemple l'action par une variable de déclenchement.

Le fait de relier et de copier les archives provoque une charge sur la CPU étant donné que la sollicitation est particulièrement plus importante lorsque le contrôle de signature du serveur SQL est activé. L'accès au disque dur est ralenti par le fait de copier les segments d'archive.

Lorsque le contrôle de signature est activé, un message d'erreur est restitué si une archive non signée ou modifiée doit être transférée. Seul un message d'erreur est restitué, même si plusieurs erreurs se produisent lors de la procédure de transfert. Une alarme système WinCC est en outre générée pour chaque segment d'archive. Une inscription s'ajoute sous "Application", dans l'indicateur d'événements Windows. Il est ainsi possible de vérifier quels segments d'archive engendrent le défaut.

- Dans le cas d'une archive non signée, la valeur en retour "0x8004720F" est restituée. L'archive est stockée. L'entrée suivante est effectuée dans l'affichage d'événements : "Validation of database <db_name> failed! No signature found!"
- Dans le cas d'une archive modifiée, la valeur en retour "0x80047207" est restituée. Le message "Validation of database <db_name> failed !" s'inscrit dans l'indicateur d'événements.
  L'archive n'est pas stockée.

Remarque

L'appel de la méthode "Restore" est actuellement possible qu'au niveau du serveur. Il existe cependant un exemple qui montre comment la méthode peut être lancée depuis le client, sur le serveur.

En cas de redondance : En cas de retransfert d'archives avec la méthode "Restore", les segments d'archive ne sont ajoutés au projet Runtime que sur l'ordinateur où la méthode a été appelée.

Syntaxe

Objets Logging, AlarmLogs

```
Expression.Restore [SourcePath] [TimeFrom] [TimeTo] [TimeOut]
[ServerPrefix]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "Logging" ou "AlarmLogs".

Objet DataLogs

```
Expression.Restore [SourcePath] [TimeFrom] [TimeTo] [TimeOut] [Type]
[ServerPrefix]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "DataLogs".

Paramètres

SourcePath

Chemin vers les données archivées.

TimeFrom

Heure à partir de laquelle les archives doivent être stockées.
Il est également possible de spécifier le format de l'heure sous forme abrégée, comme décrit au paragraphe "Format de l'heure".

TimeTo

Date/Heure jusqu'à laquelle les segments d'archive doivent être transférées.
Le format de l'heure peut également être spécifié sous forme abrégée, comme décrit au paragraphe "Format de l'heure".

Timeout

Timeout en millisecondes.
Si vous tapez la valeur "-1", l'attente est sans fin (infinite). Si vous tapez "0", il n'y a aucun temps d'attente.

Type

Type d'archive.
Ce paramètre ne peut (en option) être utilisé que pour stocker des segments d'archive de Tag Logging.
Les valeurs suivantes peuvent être saisies :

Valeur affectée	Type	Description
1	hmiDataLogFast	Tag Logging Fast data
2	hmiDataLogSlow	Tag Logging Slow data
3	hmiDataLogAll	Tag Logging Fast and Slow data

ServerPrefix

Réservé aux versions futures.

Valeur en retour

La méthode renvoie un message d'erreur si une erreur s'est produite en transférant les segments d'archive. Vous trouverez plus d'informations à ce sujet dans la rubrique "Messages d'erreur dans la zone Base de données".

Format de l'heure

Le format de l'heure est définit comme suit : YYYY-MM-DD hh:mm:ss, où YYYY désigne l'année, MM le mois, DD le jour, hh l'heure, mm les minutes et ss les secondes. 11 heures passées de 2 minutes et d'une seconde le 26 juillet 2004 sera par exemple représenté comme suit : 2004-07-26 11:02:01.

La date et l'heure peuvent être abrégées pour les paramètres "TimeFrom" et "TimeTo". Il n'est pas nécessaire de couvrir tous les champs du format. On entend par abréviation qu'il est possible de supprimer un ou plusieurs paramètres pour indiquer la date et l'heure, en commençant par la valeur des secondes. L'indication peut alors p. ex. être donnée dans le format "YYYY-MM" ou "YYYY-MM-DD hh". Si vous indiquez "TimeFrom" = "2004-09" et "TimeTo" = "2004-10-04", tous les segments d'archive sont transférés du mois de septembre 2004 jusqu'au 4 octobre inclus.

Exemple

Dans l'exemple suivant, tous les segments d'archive sont retransférés à partir de la période spécifiée et la valeur en retour est affichée en tant que Trace.

```
'VBS184
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Restore("D:\Folder","2004-09-14","",-1) &
vbNewLine
```

Dans l'exemple suivant, les segments d'archive Tag Logging Slow de la période spécifiée sont retransférés et la valeur en retour est affichée en tant que Trace.

```
'VBS185
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.DataLogs.Restore("D:\Folder","2004-09-14
12:30:05","2004-09-20 18:30",-1,2) & vbNewLine
```

Dans l'exemple suivant, les segments d'archive Alarm Logging sont retransférés jusqu'à la période spécifiée et la valeur en retour est affichée en tant que Trace.

```
'VBS186
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.AlarmLogs.Restore("","2004-09-20",-1) &
vbNewLine
```

Voir aussi

Messages d'erreur dans la zone Bases de données (Page 790)

Exemple : Pour démarrer une action sur le serveur (objet Logging) (Page 805)

Objet Logging (Page 129)

Objet DataLogs (Page 122)

Objet AlarmLogs (Page 119)

1.14.5.6 Méthodes S à T

SelectAll

Fonction

Sélectionne toutes les lignes du Control basé sur une table.

Syntaxe

```
Expression.SelectAll()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Voir aussi

Objet Row (liste) (Page 233)

SelectRow

Fonction

Sélectionne une ligne du Control basé sur une table.

Syntaxe

```
Expression.SelectRow(ByVal IRow As Long, Optional bExtendSelection
As Boolean)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

Paramètre	Description
IRow	Numéro de la ligne à sélectionner.
bExtendSelection	Indique si la sélection doit être élargie (facultatif). Uniquement significatif en cas de sélection multiple.

Exemple

- Actuellement, la ligne 1 est sélectionnée. Si SelectRow(2, True) est appelé, les lignes 1 et 2 sont sélectionnées.
- Actuellement, la ligne 1 est sélectionnée. Si SelectRow(2, False) ou SelectRow(2) est appelé sans paramètre optionnel, seule la ligne 2 est sélectionnée.

Voir aussi

Objet Row (liste) (Page 233)

Méthode SelectedStatisticArea

Fonction

Exécute la fonction de touche "Déterminer la plage de statistiques" du OnlineTableControl.

Syntaxe

```
Expression.SelectedStatisticArea()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ServerExport

Fonction

Exécute la fonction de touche "Exporter l'archive" du UserArchiveControl.

Syntaxe

```
Expression.ServerExport()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ServerImport

Fonction

Exécute la fonction de touche "Importer l'archive" du UserArchiveControl.

Syntaxe

```
Expression.ServerImport()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowColumnSelection

Fonction

Exécute la fonction de touche "Sélectionner les colonnes" du OnlineTableControl.

Syntaxe

```
Expression.ShowColumnSelection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowComment

Fonction

Exécute la fonction de touche "Dialogue Commentaire" de l'AlarmControl.

Syntaxe

```
Expression.ShowComment()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowDisplayOptionsDialog

Fonction

Exécute la fonction de touche "Dialogue des options de visualisation" de l'AlarmControl.

Syntaxe

```
Expression.ShowDisplayOptionsDialog()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowEmergencyQuitDialog

Fonction

Exécute la fonction de touche "Acquittement d'urgence" de l'AlarmControl.

Syntaxe

```
Expression.ShowEmergencyQuitDialog()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowHelp

Fonction

Exécute la fonction de touche "Aide" du contrôle.

Syntaxe

```
Expression.ShowHelp()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

VARIANT

Méthode ShowHideList

Fonction

Exécute la fonction de touche "Liste des alarmes à masquer" de l'AlarmControl.

Syntaxe

```
Expression.ShowHideList()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowHitList

Fonction

Exécute la fonction de touche "Liste d'occurrences" de l'AlarmControl.

Syntaxe

```
Expression.ShowHitList()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowInfoText

Fonction

Exécute la fonction de touche "Dialogue Texte d'info" de l'AlarmControl.

Syntaxe

```
Expression.ShowInfoText()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowLockDialog

Fonction

Exécute la fonction de touche "Dialogue de verrouillage" de l'AlarmControl.

Syntaxe

```
Expression.ShowLockDialog()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowLockList

Fonction

Exécute la fonction de touche "Liste de verrouillage" de l'AlarmControl.

Syntaxe

```
Expression.ShowLockList()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowLongTermArchiveList

Fonction

Exécute la fonction de touche "Liste d'archives longue durée" de l'AlarmControl.

Syntaxe

```
Expression.ShowLongTermArchiveList()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowMessageList

Fonction

Exécute la fonction de touche "Liste d'alarmes" de l'AlarmControl.

Syntaxe

```
Expression.ShowMessageList()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowPercentageAxis

Fonction

Exécute la fonction de touche "Axe relatif" du OnlineTrendControl.

Syntaxe

```
Expression.ShowPercentageAxis()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowPropertyDialog

Fonction

Exécute la fonction de touche "Dialogue de configuration" du contrôle.

Syntaxe

```
Expression.ShowPropertyDialog()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

VARIANT

Méthode ShowSelectArchive

Fonction

Exécute la fonction de touche "Sélectionner la connexion des données" du UserArchiveControl.

Syntaxe

```
Expression.ShowSelectArchive()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowSelection

Fonction

Exécute la fonction de touche "Dialogue de sélection" du UserArchivControl.

Syntaxe

```
Expression.ShowSelection ()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowSelectTimeBase

Fonction

Exécute la fonction de touche "Dialogue Base de temps" du UserArchivControl.

Syntaxe

```
Expression.ShowSelectTimeBase()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowSelectionDialog

Fonction

Exécute la fonction de touche "Dialogue de sélection" de l'AlarmControl.

Syntaxe

```
Expression.ShowSelectionDialog()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowShortTermArchiveList

Fonction

Exécute la fonction de touche "Liste d'archives courte durée" de l'AlarmControl.

Syntaxe

```
Expression.ShowShortTermArchiveList()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowSort

Fonction

Exécute la fonction de touche "Dialogue Tri" du UserArchivControl.

Syntaxe

```
Expression.ShowSort()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowSortDialog

Fonction

Exécute la fonction de touche "Dialogue Tri" de l'AlarmControl.

Syntaxe

```
Expression.ShowSortDialog()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowTagSelection

Fonction

Exécute la fonction de touche "Sélectionner intégration des données" du contrôle.

Syntaxe

```
Expression.ShowTagSelection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowTimebaseDialog

Fonction

Exécute la fonction de touche "Dialogue Base de temps" de l'AlarmControl.

Syntaxe

```
Expression.ShowTimebaseDialog()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowTimeSelection

Fonction

Exécute la fonction de touche "Sélectionner période" du contrôle.

Syntaxe

```
Expression.ShowTimeSelection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ShowTrendSelection

Fonction

Exécute la fonction de touche "Sélectionner les courbes" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.ShowTrendSelection()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

-- -

Méthode StartStopUpdate

Fonction

Exécute la fonction de touche "Marche" ou "Arrêt" du contrôle.

Syntaxe

```
Expression.StartStopUpdate()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode Stop

Fonction

Quitter WinCC Runtime.

Syntaxe

```
HMIRuntime.Stop
```

Paramètres

Exemple

L'exemple suivant montre comment quitter WinCC Runtime:

```
'VBS124
HMIRuntime.Stop
```

Voir aussi

Objet HMIRuntime (Page 125)

Methode Trace

Description

Affiche des alarmes dans la fenêtre de diagnostic.

Syntaxe

```
HMIRuntime.Trace
```

Paramètres

String

Exemple

L'exemple suivant présente l'écriture d'un texte dans la fenêtre de diagnostic:

```
'VBS103
HMIRuntime.Trace "Customized error message"
```

Voir aussi

Objet HMIRuntime (Page 125)

1.14.5.7 Méthodes U à Z

Méthode UnhideAlarm

Fonction

Exécute la fonction de touche "Afficher l'alarme" de l'AlarmControl.

Syntaxe

```
Expression.UnhideAlarm()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode UnlockAlarm

Fonction

Exécute la fonction de touche "Valider l'alarme" de l'AlarmControl.

Syntaxe

```
Expression.UnlockAlarm()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

UnselectAll

Fonction

Supprime la sélection de toutes les lignes du Control basé sur une table.

Syntaxe

```
Expression.UnselectAll()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Voir aussi

Objet Row (liste) (Page 233)

UnselectRow

Fonction

Supprime la sélection d'une ligne du Control basé sur une table.

Syntaxe

```
Expression.UnselectRow(ByVal IRow As Long)
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

Long

Paramètre	Description
IRow	Numéro de la ligne à sélectionner.

Voir aussi

Objet Row (liste) (Page 233)

Methode Write

Description pour l'objet Tag

Ecrit une valeur de façon synchrone ou asynchrone dans une variable. Il est possible de déterminer via la propriété "LastError" si l'appel a correctement fonctionné.

Si la valeur de la variable a correctement été réglée, les propriétés de l'objet "Tag" reçoivent les valeurs suivantes :

Propriété	Affectation
value	Valeur de variable réglée par l'utilisateur (inchangée)
Nom	Nom de variable (reste inchangé)
QualityCode	Bad Out of Service
TimeStamp	0
LastError	0
ErrorDescription	" "

Si la valeur de la variable n'a pas correctement été réglée, les propriétés de l'objet "Tag" reçoivent les valeurs suivantes :

Propriété	Affectation
value	Valeur de variable réglée par l'utilisateur (inchangée)
Nom	Nom de variable (reste inchangé)
QualityCode	Bad Out of Service
TimeStamp	0
LastError	Code d'erreur de l'opération d'écriture
ErrorDescription	Description de l'erreur pour LastError

Syntaxe

```
Expression.Write [Value],[Writemode]
```

Expression

Impératif. Un travail d'impression qui retransmet un objet "Tags".

Paramètres

La valeur à écrire peut directement être transmise à la méthode en tant que paramètre. Si le paramètre n'est pas spécifié, c'est la valeur de la propriété "Value" est utilisée. Le paramètre optionnel "Writemode" vous permet de sélectionner si la valeur de variable doit être écrite sous forme synchrone ou asynchrone. Si le paramètre "Writemode" n'est pas utilisé, il est écrit comme valeur par défaut sous forme asynchrone.

Lors d'une opération d'écriture, aucune information concernant l'état des variables n'est fournie.

Dans la propriété "Value" se trouve la valeur qui a été positionnée avant ou lors de l'opération d'écriture, elle ne doit donc pas forcément correspondre à la valeur actuelle de la variable. Si vous voulez mettre à jour les informations concernant la variable, il vous faut utiliser la méthode Read.

Paramètres	Description
Value (optionnel)	La valeur de la variable est spécifiée. La valeur spécifiée décrit la valeur de la propriété "Value" dans l'objet "Tag". La valeur de la variable n'est pas spécifiée. La variable reçoit la valeur actuelle de la propriété "Value" de l'objet "Tag".
Writemode (en option)	0 ou vide : La valeur de la variable est écrite de façon asynchrone. 0 est la valeur par défaut. 1: La valeur de la variable est écrite de façon synchrone.

En cas d'écriture asynchrone, l'inscription se fait immédiatement dans l'image de variable. L'utilisateur n'est pas informé si la valeur a ou non été inscrite dans l'AS.

Lors de l'écriture synchrone (directement dans l'AS), l'opération d'écriture démarre seulement lorsque l'automate programmable est prêt. L'utilisateur reçoit un retour d'information si l'opération d'écriture n'a pas été correctement effectuée.

Exemple

Ecriture asynchrone

```
'VBS104
Dim objTag
Set objTag = HMIRuntime.Tags("Var1")
objTag.Value = 5
objTag.Write
MsgBox objTag.Value
```

ou

```
'VBS105
Dim objTag
Set objTag = HMIRuntime.Tags("Var1")
objTag.Write 5
MsgBox objTag.Value
```

écriture synchrone

```
'VBS106
Dim objTag
Set objTag = HMIRuntime.Tags("Var1")
objTag.Value = 5
objTag.Write ,1
MsgBox objTag.Value
```

ou

```
'VBS107
Dim objTag
Set objTag = HMIRuntime.Tags("Var1")
objTag.Write 5, 1
MsgBox objTag.Value
```

Description pour TagSet

L'objet TagSet offre la possibilité d'écrire plusieurs variables en un appel.

Le fonctionnement est pratiquement similaire à celui d'un objet Tag. Seules les divergences sont décrites ci-après.

Expression

Impératif. Un travail d'impression qui retransmet un objet de type "TagSet".

Paramètres

Pour écrire des valeurs différentes, il vous faut valider la propriété "Value" de chacun des objets Tag puis sélectionner Write sans paramètre "Value". Etant donné que les contrats d'écriture sont regroupés en un appel, les performances sont améliorées comparé aux appels individuels.

Pour l'objet TagSet, il n'est pas possible d'ajouter une valeur avec la méthode "Write". Chaque valeur doit être positionnée via la propriété "Value" des divers objets Tag.

Exemple

L'exemple suivant montre comment l'on accepte des variables dans la liste TagSet, comment l'on active puis écrit les valeurs variable.

```
'VBS173
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Valeur1"
group.Add "Valeur2"
group("Valeur1").Value = 3
group("Valeur2").Value = 9
group.Write
```

Si l'on positionne le paramètre optionnel "Writemode" sur 1, les variables de process sont écrites de façon synchronisée (directement dans l'AS).

```
group.Write 1
```

Voir aussi

Propriété LastError (Page 432)

Propriété ErrorDescription (Page 386)

Objet TagSet (Liste) (Page 149)

Objet Tag (Page 144)

Méthode WriteTags

Fonction

Exécute la fonction de touche "Ecrire variables" du UserArchiveControl.

Syntaxe

```
Expression.WriteTags()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètre

- -

Méthode ZoomArea

Fonction

Exécute la fonction de touche "Zone de zoom" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.ZoomArea()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ZoomInOut

Fonction

Exécute la fonction de touche "Zoom +/-" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.ZoomInOut()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ZoomInOutTime

Fonction

Exécute la fonction de touche "Zoomer l'axe des temps +/-" du OnlineTrendControl.

Syntaxe

```
Expression.ZoomInOutTime()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ZoomInOutValues

Fonction

Exécute la fonction de touche "Zoomer l'axe des valeurs +/-" du OnlineTrendControl.

Syntaxe

```
Expression.ZoomInOutValues()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ZoomInOutX

Fonction

Exécute la fonction de touche "Zoom +/- de l'axe X" du FunctionTrendControl.

Syntaxe

```
Expression.ZoomInOutX()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ZoomInOutY

Fonction

Exécute la fonction de touche "Zoom +/- de l'axe Y" du FunctionTrendControl.

Syntaxe

```
Expression.ZoomInOutY()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

Méthode ZoomMove

Fonction

Exécute la fonction de touche "Déplacer la zone de courbes" du OnlineTrendControl et du FunctionTrendControl.

Syntaxe

```
Expression.ZoomMove()
```

Expression

Impératif. Une expression qui renvoie un objet de type "ScreenItem".

Paramètres

- -

1.14.6 Annexe

1.14.6.1 Messages d'erreur dans la zone Bases de données

Introduction

Pendant les accès aux bases de données, une valeur est restituée après exécution. Les valeurs de la plage "0x8..." représentent un message d'erreur. Les valeurs autres que "0x8..." représentent un message d'état.

Messages d'état

Les messages d'état suivants sont définis :

0x0	OK
0x1	La fonction n'a constaté aucune erreur dans l'affectation paramètres et n'a identifié aucune erreur interne. Cette valeur de retour peut avoir les causes suivantes. En reliant des bases de données : - Aucune archive trouvée dans la fenêtre de temps spécifiée. - Des archives ont certes été trouvées dans la fenêtre de temps spécifiée, mais elles étaient déjà liées. En dissociant des bases de données : - Aucune archive liée trouvée dans la fenêtre de temps spécifiée. Il n'est pas vérifié s'il existe ou non des archives reliées.

Messages d'erreur

Les messages d'erreur suivants sont définis (en anglais uniquement) :

Code d'erreur	Erreur
0x80047200	WinCC is not activated
0x80047201	Invalid archive type
0x80047202	Invalid lower boundary
0x80047203	Invalid upper boundary
0x80047204	Path 'CommonArchiving' could not be created in the project path
0x80047205	Timeout, please retry
0x80047206	WinCC was deactivated
0x80047207	Wrong signification At least one database had a invalid signature and has not been attached.
0x80047208	Database could not be attached
0x80047209	Copy to 'CommonArchiving' is not possible.
0x8004720A	Invalide syntax for database filename.
0x8004720B	No list of databases.
0x8004720C	Database already detached.
0x8004720D	Database could not be detached.

Code d'erreur	Erreur
0x8004720F	Unsigned database attached. At least one database without signature has been attached.
0x80047210	Path error : - Path invalid, - no *.MDF files found in specified path or - no permission to specified path.

Voir aussi

Méthode Remove (Page 758)

Methode Write (Page 783)

Methode Read (Page 753)

Méthode Restore (Page 764)

Objet Logging (Page 129)

Objet DataLogs (Page 122)

Objet AlarmLogs (Page 119)

1.15 Exemples pour VBScript

1.15.1 Exemples pour VBScript

Introduction

Vous trouverez ci-après des exemples d'application pour VBS dans WinCC. La section "Exemples dans WinCC" présente des exemples de code vous permettant de dynamiser votre environnement Runtime sous WinCC. Ces exemples sont conçus de façon à ce que vous puissiez les appliquer 1:1 dans votre configuration.

La section "Exemples, généralités" présente des exemples avec lesquels vous pouvez influencer votre environnement Microsoft. Aucune garantie, aucune assistance technique ne peuvent être données en ce qui concerne l'exécutabilité de ces exemples

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2 Exemples dans WinCC

1.15.2.1 Exemples dans WinCC

Introduction

Vous trouverez ici des exemples concernant les applications de VBScript dans WinCC pour les rubriques suivantes:

- Accès aux objets dans le Graphics Designer (par ex. modification de la couleur ou du texte)
- Régler la couleur des objets par les couleurs RJB
- Configurer le changement de la langue
- Désactiver Runtime
- Démarrer un programme externe
- Configurer le changement de vue global (depuis Global Script)
- Configurer un changement de vue via Property
- Utiliser Trace pour la sortie du diagnostic
- Paramétrer la valeur d'une variable
- Lire la valeur d'une variable
- Vérifier l'accomplissement d'une action en lecture/écriture dans une variable.
- Paramétrer la valeur d'une variable de façon asynchrone

Voir aussi

Exemple : Pour démarrer une application tierce (Page 824)
Exemple : Comment écrire les propriétés de l'objet (Page 802)
Exemple : Pour lire les valeurs variable (Page 799)
Exemple : Pour écrire les valeurs variable (Page 797)
Exemple : Comment configurer l'affichage d'un diagnostic via Trace (Page 796)
Exemple : Comment configurer un changement de vue via Property (Page 796)
Exemple : Comment configurer globalement un changement de vue (Page 795)
Exemple : Comment désactiver Runtime (Page 795)
Exemple : Pour configurer un changement de langue (Page 794)
Exemple : Comment déterminer la couleur des objets (Page 794)
Exemple : Pour accéder à des objets dans Graphics Designer (Page 793)

1.15.2.2 Exemple : Pour accéder à des objets dans Graphics Designer

Introduction

Dans WinCC, VBS vous permet d'accéder à tous les objets du Graphics Designer pour dynamiser l'environnement graphique Runtime. Vous pouvez ainsi dynamiser sur commande les objets graphiques (en cliquant sur un bouton avec la souris par exemple), en fonction d'une variable ou de façon cyclique (clignotement par exemple).

Les exemples suivants vous indiquent comment modifier un objet graphique en cliquant sur la souris.

Marche à suivre

Dans l'exemple suivant, vous réglez en Runtime le rayon d'un cercle sur 20 en cliquant sur la souris:

```
'VBS121
Dim objCircle
Set objCircle= ScreenItems("Circle1")
objCircle.Radius = 20
```

Remarque

La forme d'impression utilisée dans l'exemple ne peut être exécutée que dans le Graphics Designer. Pour les actions analogues dans Global Script, adressez les objets via l'objet HMIRuntime.

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2.3 Exemple : Comment déterminer la couleur des objets

Introduction

Les couleurs des objets graphiques sont définies avec les valeurs RGB (Rouge/Vert/Bleu). Vous pouvez définir ou lire des valeurs de couleur pour les objets graphiques.

Marche à suivre

L'exemple suivant positionne sur Bleu la couleur de remplissage de la vue "ScreenWindow1" :

```
'VBS122
Dim objScreen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
objScreen.FillStyle = 131075
objScreen.FillColor = RGB(0, 0, 255)
```

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2.4 Exemple : Pour configurer un changement de langue

Introduction

Vous pouvez permuter le langue Runtime de WinCC avec VBS. L'application la plus courante sont les Buttons portant l'indicatif des pays correspondants que vous placez sur la page d'accueil de votre projet.

Spécifiez la langue Runtime en VBS avec un indicatif de pays, p, ex. 1031 pour l'allemand - standard, 1033 pour l'anglais - USA, etc. Vous trouverez dans les principes de base de VBScript sous la rubrique "Diagramme du schéma région-ID (LCID)" un récapitulatif de tous les indicatifs de pays.

Marche à suivre

Créez une action VBS sur l'événement "Clic de souris" pour un Button et tapez le code d'action suivant pour faire permuter sur l'allemand la langue Runtime:

```
'VBS123
HMIRuntime.Language = 1031
```

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2.5 Exemple : Comment désactiver Runtime

Introduction

VBS vous permet de quitter WinCC-Runtime, en cliquant par exemple sur la souris ou en fonction des valeurs variable ou en fonction d'autres événements, comme par exemple la saisie d'un mot de passe erroné plusieurs fois lors du démarrage du Runtime.

Marche à suivre

L'exemple suivant montre comment quitter WinCC Runtime:

```
'VBS124
HMIRuntime.Stop
```

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2.6 Exemple : Comment configurer globalement un changement de vue

Introduction

Vous pouvez changer de vue globalement avec VBS, et afficher ainsi par exemple la vue d'un serveur sur un poste client dans un système réparti. Le Serverprefix du serveur doit à cet effet précéder la vue cible.

Marche à suivre

Pour le changement de vue, configurez par exemple le code suivant sur un Button:

```
'VBS125
HMIRuntime.BaseScreenName = "Serverprefix::New screen"
```

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2.7 Exemple : Comment configurer un changement de vue via Property

Introduction

Si vous utilisez dans votre configuration des vues partagées, par exemple dans une vue de base le titre et la barre de commande du panneau de configuration plus une fenêtre de vue incorporée pour l'affichage réel de la vue, configurez le changement de vue via les propriétés de la fenêtre de vue.

La fonction Property de la fenêtre de vue "ScreenName" doit être changée pour que l'autre vue apparaisse. L'action et la fenêtre de vue doivent être configurées dans la même vue.

Marche à suivre

Dans l'exemple suivant, vous affichez dans une fenêtre de vue "ScreenWindow" la vue "test.pdl" lors de l'exécution de l'action:

```
'VBS126
Dim objScrWindow
Set objScrWindow = ScreenItems("ScreenWindow")
objScrWindow.ScreenName = "test"
```

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2.8 Exemple : Comment configurer l'affichage d'un diagnostic via Trace

Introduction

Si vous avez inséré dans votre vue une fenêtre de diagnostic GSC, vous pouvez afficher en Runtime les diagnostics avec la commande Trace.

La fonction Diagnostic GSC affiche, dans l'ordre chronologique de leur appel, les méthodes Trace contenues dans les actions. Il affiche également les instructions Trace codées dans les procédures appelées dans des actions. L'utilisation ciblée d'instructions Trace, p. ex. pour afficher des valeurs de variable, permet de suivre l'exécution des actions et des procédures appelées dans celles-ci. Spécifiez les instructions Trace sous la forme "HMIRuntime.Trace(<Sortie>)".

Les affichages Trace de C et VBS sont édités dans la fenêtre de diagnostic GSC.

Marche à suivre

L'exemple suivant présente l'écriture d'un texte dans la fenêtre de diagnostic:

```
'VBS127
```

```
HMIRuntime.Trace "Customized error message"
```

Voir aussi

Exemples dans WinCC (Page 792)

1.15.2.9 Exemple : Pour écrire les valeurs variable

Introduction

VBS vous permet d'écrire la valeur d'une variable dans l'automate, par exemple en cliquant avec la souris sur un Button pour définir la valeur de consigne, ou attribuer des valeurs variable internes pour déclencher d'autres actions.

Diverses variantes d'écriture sont présentées et expliquées ci-après.

Ecriture simple

Dans l'exemple suivant, une valeur est écrite dans la variable "Tag1":

```
'VBS128
HMIRuntime.Tags("Tag1").Write 6
```

Il s'agit de la forme d'écriture la plus simple étant donné qu'aucune référence d'objet n'est créée.

Ecrire avec référence d'objet

Dans l'exemple suivant, on génère une copie locale de l'objet "Tag" et on écrit une valeur dans la variable "Tag1":

```
'VBS129
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Write 7
```

Le référençage donne l'avantage de pouvoir travailler avec l'objet "Tag" avant d'écrire. Vous pouvez lire la valeur variable, effectuer des calculs et à nouveau écrire:

```
'VBS130
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
objTag.Value = objTag.Value + 1
objTag.Write
```

écriture synchrone

La valeur à écrire est en standard transmise à la gestion des variables et le traitement de l'action se poursuit. Dans certains cas, il est cependant nécessaire de garantir que la valeur a effectivement été écrite avant que l'action ne puisse se poursuivre.

Réalisez ce type d'écriture en spécifiant la valeur 1 pour le paramètre supplémentaire optionnel:

```
'VBS131
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Write 8,1
```

ou

```
'VBS132
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Value = 8
objTag.Write ,1
```

Remarque

Notez que l'appel prend plus de temps comparé à un appel standard. La durée dépend entre autres du canal et de l'AS.

Ce type d'écriture correspond à l'appel de SetTagXXXWait() dans C-Scripting.

Ecrire avec traitement d'état

Pour être certain qu'une valeur a correctement été écrite, il est nécessaire d'effectuer un contrôle d'erreur après écriture ou de définir l'état de la variable.

Cela est possible en contrôlant la propriété LastError après écriture de la valeur. Si le contrôle a un résultat correct, c'est-à dire que le contrat a correctement pu être exécuté, l'état de la variable est contrôlé.

Lors d'une opération d'écriture, l'état actuel n'est pas défini dans le processus. Pour le définir, il faut que la variable soit lue. La valeur contenue dans la propriété QualityCode après la procédure de lecture informe sur l'état de la variable et informe au besoin qu'une liaison AS a échoué.

Dans l'exemple suivant, la variable "Tag1" est écrite: Si une erreur survient lors de l'écriture, la valeur et la description de cette erreur s'affichent dans la fenêtre de diagnostic Global Script. C'est le QualityCode qui est ensuite contrôlé. Si le QualityCode n'est pas correct (0x80), il s'affiche dans la fenêtre de diagnostic.

```
'VBS133
```

```
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Write 9
If 0 <> objTag.LastError Then
HMIRuntime.Trace "Error: " & objTag.LastError & vbCrLf & "ErrorDescription: " &
objTag.ErrorDescription & vbCrLf
    Else
objTag.Read
If &H80 <> objTag.QualityCode Then
HMIRuntime.Trace "QualityCode: 0x" & Hex(objTag.QualityCode) & vbCrLf
    End If
    End If
```

Remarque

Après l'écriture d'une variable, la propriété QualityCode de l'objet local "Tag" est activée sur "BAD Out of Service", étant donné qu'on ne sait pas quel QualityCode la variable possède dans le processus.

Le QualityCode ne peut pas être écrit à partir de VBS.

Voir aussi

Methode Write (Page 783)

Exemples dans WinCC (Page 792)

1.15.2.10 Exemple : Pour lire les valeurs variable

Introduction

VBS vous permet de lire la valeur d'une variable et la traiter. Vous pouvez ainsi par exemple en cliquant la souris sur un Button vous informer sur l'état de l'installation ou effectuer des calculs.

Diverses variantes de lecture sont présentées et expliquées ci-après.

Lecture simple

Dans l'exemple suivant, la valeur de la variable "Tag1" est lue et affichée dans la fenêtre de diagnostic Global Script :

```
'VBS134
HMIRuntime.Trace "Value: " & HMIRuntime.Tags("Tag1").Read & vbCrLf
```

Il s'agit de la forme de lecture la plus simple étant donné qu'aucune référence d'objet n'est créée.

Lecture avec référence d'objet

Dans l'exemple suivant, on crée une copie locale de l'objet "Tag", la valeur variable est lue puis affichée dans la fenêtre de diagnostic Global Script :

```
'VBS135
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
HMIRuntime.Trace "Value: " & objTag.Read & vbCrLf
```

Le référençage donne l'avantage de pouvoir travailler avec l'objet "Tag". Vous pouvez lire la valeur variable, effectuer des calculs et à nouveau écrire :

```
'VBS136
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
objTag.Value = objTag.Value + 1
objTag.Write
```

Les variables de process lues avec la méthode Read sont ajoutées à la mémoire image, et sont à partir de ce moment là cycliquement sélectionnées dans l'automate programmable (AS). Si la variable se trouve déjà dans la mémoire image, c'est la valeur s'y trouvant qui est donnée.

En cas de désélection de vue, les variables sont à nouveau déconnectées.

Remarque

Si une variable est demandée dans une action Global Script, elle reste connectée pendant toute la durée d'exécution Runtime de WinCC.

Lecture directe

Les valeurs variable sont en standard lues dans la mémoire image variable. Dans certains cas, il peut toutefois s'avérer nécessaire de lire directement la valeur dans l'AS, pour par exemple synchroniser des opérations rapides.

Si l'on positionne sur 1 le paramètre optionnel lors de la lecture, la variable ne sera pas connectée de façon cyclique mais la valeur sera sélectionnée une seule et unique fois depuis l'AS.

```
'VBS137
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
HMIRuntime.Trace "Value: " & objTag.Read(1) & vbCrLf
```

Remarque

Notez que l'appel prend plus de temps comparé à un appel standard. La durée dépend entre autres du canal et de l'AS.

Dans les actions C cycliques, évitez ce type d'appel étant donné qu'il s'agit là de la source principale des problèmes de performance.

Ce type d'écriture correspond à l'appel de GetTagXXXWait() dans C-Scripting

Lecture avec traitement d'état

Pour être certain qu'une valeur est valable, il est recommandé d'effectuer un contrôle après la lecture. Cela s'effectue en contrôlant le QualityCode.

Dans l'exemple suivant, la variable "myWord" est lue, puis le QualityCode est contrôlé. Si le QualityCode n'est pas correct (0x80), les propriétés LastError, ErrorDescription et QualityCode s'affichent dans la fenêtre de diagnostic Global Script.

```
'VBS138
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
If &H80 <> objTag.QualityCode Then
HMIRuntime.Trace "Error: " & objTag.LastError & vbCrLf & "ErrorDescription: " &
objTag.ErrorDescription & vbCrLf & "QualityCode: 0x" & Hex(objTag.QualityCode) & vbCrLf
    Else
HMIRuntime.Trace "Value: " & objTag.Value & vbCrLf
    End If
```

Remarque

Si une erreur survient lors de la lecture, le QualityCode est positionné sur "BAD Out of Service". C'est pourquoi il suffit de contrôler le QualityCode lors de la lecture.

Voir aussi

Methode Read (Page 753)

Exemples dans WinCC (Page 792)

1.15.2.11 Exemple : Comment écrire les propriétés de l'objet

Introduction

VBS permet d'accéder aux propriétés de tous les objets d'image du Graphics Designer. Vous pouvez lire les propriétés ou les modifier en Runtime.

Les exemples suivants vous présentent diverses formes d'accès.

Attribution simple d'une propriété

Dans l'exemple suivant, la couleur d'arrière-plan de l'objet "Rectangle1" contenu dans la vue est positionnée sur Rouge :

```
'VBS139
ScreenItems("Rectangle1").BackColor = RGB(255,0,0)
Il s'agit de la forme d'écriture la plus simple étant donné qu'aucune référence d'objet
n'est créée.
```

Remarque

Si vous travaillez sans référence d'objet, seules les propriétés standard sont exécutées sous Intellisense.

La forme d'impression utilisée dans cet exemple ne peut être exécutée que dans le Graphics Designer. Pour les actions analogues dans Global Script, adressez les objets via l'objet HMIRuntime.

Attribution d'une propriété avec référence d'objet

Dans l'exemple suivant, une référence est créée pour l'objet "Rectangle1" contenu dans la vue et la couleur d'arrière-plan est positionnée sur Rouge à l'aide de la fonction standard RJB() de VBS :

```
'VBS140
Dim objRectangle
Set objRectangle = ScreenItems("Rectangle1")
objRectangle.BackColor = RGB(255,0,0)
```

Le référençage est judicieux lorsque vous voulez modifier plusieurs propriétés d'un même objet. Vous obtenez de cette façon sous Intellisense une liste de toutes les propriétés de l'objet.

Remarque

La forme d'impression utilisée dans cet exemple ne peut être exécutée que dans le Graphics Designer. Pour les actions analogues dans Global Script, adressez les objets via l'objet HMIRuntime.

Imposer des propriétés via la fenêtre des vues

VBS dans Graphics Designer offre deux possibilités d'adressage pour l'ensemble des vues :

- par l'objet Screen d'une fenêtre de vue avec "ScreenItems"
- depuis la vue de base avec "HMIRuntime.Screens"

Référençage par la fenêtre de vue

Dans l'exemple suivant, la couleur d'un rectangle est modifiée dans une fenêtre de vue sous-jacente. Le Script est exécuté dans la vue "BaseScreen" dans laquelle se trouve la fenêtre "ScreenWindow1". Dans la fenêtre de vue, vous voyez s'afficher une vue contenant un objet de type "Rectangle" appelé "Rectangle1".

```
'VBS199
Sub OnLButtonUp(ByVal Item, ByVal Flags, ByVal x, ByVal y)
Dim objRectangle
Set objRectangle = ScreenItems("ScreenWindow1").Screen.ScreenItems("Rectangle1")
objRectangle.BackColor = RGB(255,0,0)
End Sub
```

Référençage depuis la vue de base

HMIRuntime.Screens vous permet de référencer la vue avec l'objet à modifier. La spécification de la vue se fait alors en fonction de la vue de base, via le code d'accès suivant :

[<Nom de la vue de base>.]<Nom de la fenêtre de vue>[:<Nom de la vue>]... .<Nom de la fenêtre de vue>[:<Nom de la vue>]

Dans l'exemple suivant, une référence est créée pour l'objet "Rectangle1" contenu dans la vue "Screen2" et la couleur d'arrière-plan est positionnée sur Rouge.

La vue "Screen2" se trouve alors dans "Screen1". "Screen1" s'affiche dans la vue de base "BaseScreen".

```
'VBS141
Dim objRectangle
Set objRectangle =
HMIRuntime.Screens("BaseScreen.ScreenWindow1:Screen1.ScreenWindow1:Screen2").ScreenItems("
Rectangle1")
```

```
objRectangle.BackColor = RGB(255,0,0)
```

Il n'est pas nécessaire de spécifier le nom de la vue. Le nom de la fenêtre de vue permet d'adresser une vue de façon univoque. Comme présenté dans l'exemple suivant, il suffit de spécifier le nom de la fenêtre de vue :

```
'VBS142
Dim objRectangle
Set objRectangle =
HMIRuntime.Screens("ScreenWindow1.ScreenWindow2").ScreenItems("Rectangle1")
objRectangle.BackColor = RGB(255,0,0)
```

Cette méthode d'adressage permet d'accéder à des objets dans les fenêtres lorsque les vues changent. Cela s'avère être particulièrement intéressant pour la technique à vues modulables.

Dynamisation de la propriété par une valeur de retour

Vous pouvez non seulement déclencher des actions pour des propriétés avec un Trigger d'événement ou un Trigger cyclique, mais dynamiser également des propriétés directement par le biais d'une action.

Dans l'exemple suivant, la couleur d'arrière-plan d'un objet est dynamisée par une valeur de retour. La valeur transmise peut par exemple provenir de l'interprétation d'événements dans l'automate et être utilisée pour la représentation graphique d'un état de service :

```
'VBS146
Function BackColor_Trigger(ByVal Item)
    BackColor_Trigger = RGB(125,0,0)
End Function
```

Remarque

Si vous dynamisez une propriété d'objet avec une action VBS par le biais de la valeur retour d'un script, la valeur de la propriété d'objet ne sera inscrite que si elle a changé par rapport au dernier passage du script. Peu importe alors si la valeur a été modifiée depuis un autre endroit.

C'est pourquoi les propriétés qui ont été dynamisées avec une action VBS par le biais de la valeur retour, ne sont pas modifiées d'un autre endroit (autres scripts C ou VBS par exemple).

Si vous ne prenez pas ce fait en considération, il se peut que les valeurs affichées soient erronées.

Voir aussi

Référence VBS (Page 110)

Exemples dans WinCC (Page 792)

1.15.2.12 Exemple : Pour démarrer une action sur le serveur (objet Logging)

Introduction

Dans les projets multipostes, l'objet Logging ne fonctionne actuellement que sur le serveur. Dans l'exemple suivant, il est montré comment démarrer depuis le client une action sur le serveur, et comment ainsi transférer ou supprimer des segments d'archive sur le client.

L'exemple montre une action globale démarrée avec une variable de forçage. Le contenu de la variable de forçage détermine si la méthode "Restore" ou "Remove" sera appelée. A la fin de l'action, la variable de forçage est positionnée sur "0".

Une interrogation empêche de démarrer l'action sur les ordinateurs client.

Le chemin et la période sont transmis par des variables internes.

Il est également possible d'indiquer une validation réseau pour spécifier le chemin. Les segments d'archive à transférer ne doivent donc pas impérativement se trouver localement sur le serveur. Il faut toutefois pouvoir garantir que le serveur puisse accéder correctement au chemin.

Remarque

L'exemple montre une solution possible que vous pouvez au besoin adapter.

Marche à suivre

1. Créez sous WinCC Explorer les variables internes suivantes avec rafraîchissement dans tout le projet :
   - StartLogging (Valeur 8 bits non signée)
   - SourcePath (variable de texte 8 bits jeu de caractères
   - TimeFrom (variable de texte 8 bits jeu de caractères)
   - TimeTo (variable de texte 8 bits jeu de caractères)
   - RetVal (Valeur 32 bits signée)
2. Créez une action globale VBS et inscrivez comme déclenchement de variable, la variable 'StartLogging' accompagnée du cycle Sur modification.
3. Copiez le script suivant dans l'action

```
'VBS180
Dim StartLogging
Dim SourcePath
Dim TimeFrom
Dim TimeTo
Dim RetVal
```

```
'Exit when running on client
If (Left(HMIRuntime.ActiveProject.Path, 1) = "\") Then
Exit Function
    End If
'read parameters
StartLogging = HMIRuntime.Tags("StartLogging").Read
SourcePath = HMIRuntime.Tags("SourcePath").Read(1)
TimeFrom = HMIRuntime.Tags("TimeFrom").Read(1)
TimeTo = HMIRuntime.Tags("TimeTo").Read(1)
'restore or remove depends on the parameter
If (StartLogging = 1) Then
RetVal = HMIRuntime.Logging.Restore(SourcePath, TimeFrom, TimeTo, -1)
HMIRuntime.Tags("RetVal").Write RetVal, 1
HMIRuntime.Tags("StartLogging").Write 0,1
Elseif (StartLogging = 2) Then
RetVal = HMIRuntime.Logging.Remove(TimeFrom, TimeTo, -1)
HMIRuntime.Tags("RetVal").Write RetVal, 1
HMIRuntime.Tags("StartLogging").Write 0,1
    End If
```

L'action peut démarrer sur un client, p. ex. avec l'action suivante. Notez que les paramètres doivent être écrits avant de valider les variables de forçage.

```
'VBS181
'set parameters
HMIRuntime.Tags("SourcePath").Write "\\client_pc\temp",1
HMIRuntime.Tags("TimeFrom").Write "2004",1
HMIRuntime.Tags("TimeTo").Write "2004",1
'start action
HMIRuntime.Tags("StartLogging").Write 1,1
```

Remarque

Les variables sont la plupart du temps écrites et lues en mode "direct". Les opérations sont ainsi synchronisées. Ce mode peut être utilisé sans problème étant donné qu'il s'agit de variables internes.

1.15.2.13 Dynamisation des contrôles

Exemple : Pour appeler les méthodes d'un ActiveX-Control

Introduction

Les exemples suivants vous montrent comment appeler des méthodes et des propriétés d'un ActiveX Control incorporé dans une vue WinCC.

Exemple 1 : WinCC FunctionTrendControl

Cet exemple remplit la courbe "Trend 1" du FunctionTrendControl "Control1" avec des valeurs décrivant une parabole.

Pour dynamiser une courbe avec VBS, vous paramétrez dans le dialogue de configuration du contrôle dans l'onglet "Connexion de données", sous "Sélection de données", "0 - aucune".

```
'VBS300
Dim lngFactor
Dim dblAxisX
Dim dblAxisY
Dim objTrendControl
Dim objTrend

Set objTrendControl = ScreenItems("Control1")
Set objTrend = objTrendControl.GetTrend("Trend 1")

For lngFactor = -100 To 100
dblAxisX = CDbl(lngFactor * 0.02)
dblAxisY = CDbl(dblAxisX * dblAxisX + 2 * dblAxisX + 1)
objTrend.InsertData dblAxisX, dblAxisY
Next
```

Exemple 2 : WinCC FunctionTrendControl avec sélection des valeurs par tableau

Dans cet exemple, la courbe "Trend 1" du FunctionTrendControl "Control1" est alimentée en valeurs qui sont enregistrées dans des tableaux.

Pour dynamiser une courbe avec VBS, vous paramétrez dans le dialogue de configuration du contrôle dans l'onglet "Connexion de données", sous "Sélection de données", "0 - aucune".

```
'VBS301
Dim lngIndex
Dim dblAxisX(100)
Dim dblAxisX(100)
Dim objTrendControl
Dim objTrend
Set objTrendControl = ScreenItems("Control1")
Set objTrend = objTrendControl.GetTrend("Trend 1")
```

```
For lngIndex = 0 To 100
dblAxisX(lngIndex) = CDbl(lngIndex * 0.8)
dblAxisY(lngIndex) = CDbl(lngIndex)
Next
objTrend.InsertData dblAxisX, dblAxisY
```

Exemple 3 : WinCC FunctionTrendControl (avant WinCC V7)

Dans cet exemple, la FunctionTrendControl appelée "Control1" est complétée par des valeurs qui décrivent une parabole.

```
'VBS111
Dim lngFactor
Dim dblAxisX
Dim dblAxisY
Dim objTrendControl
Set objTrendControl = ScreenItems("Control1")
For lngFactor = -100 To 100
dblAxisX = CDbl(lngFactor * 0.02)
dblAxisY = CDbl(dblAxisX * dblAxisX + 2 * dblAxisX + 1)
objTrendControl.DataX = dblAxisX
objTrendControl.DataY = dblAxisY
objTrendControl.InsertData = True
Next
```

Exemple 4 : WinCC Function TrendControl avec sélection des valeurs par tableau (avant WinCC V7)

Dans cet exemple, 100 paires de valeurs alimentent une FunctionTrendControl appelée "Control1". Pour que ces paires de valeurs soient correctement transférées, le transfert ne doit pas être réalisé directement par ex. dans "dblAxisXY" mais à l'aide d'une variable intermédiaire, par ex. "varTemp".

```
'VBS152
Dim lngIndex
Dim dblXY(1)
Dim dblAxisXY(100)
Dim varTemp
Dim objTrendControl
Set objTrendControl = ScreenItems("Control1")
For lngIndex = 0 To 100
 dblXY(0) = CDbl(lngIndex * 0.8)
 dblXY(1) = CDbl(lngIndex)
 dblAxisXY(lngIndex) = dblXY
Next
varTemp = (dblAxisXY)
objTrendControl.DataXY = varTemp
objTrendControl.InsertData = True
```

Exemple 5 : Microsoft Web Browser

Cet exemple montre comment est commandé MS Web Browser.

```
'VBS112
Dim objWebBrowser
Set objWebBrowser = ScreenItems("WebControl")
objWebBrowser.Navigate "http://www.siemens.de"
...
objWebBrowser.GoBack
...
objWebBrowser.GoForward
...
objWebBrowser.Refresh
...
objWebBrowser.GoHome
...
objWebBrowser.GoSearch
...
objWebBrowser.Stop
...
```

Remarque

Installez les instructions séparées par des points dans vos propres procédures. Déclaration et assignations doivent toujours se trouver devant.

Voir aussi

Exemples d'ordre général pour VBScript (Page 819)

Exemple : Vous configurez de la sorte un bouton de barre d'outils personnalisé avec un dialogue de sélection créé en propre.

Introduction

Dans l'exemple suivant, vous créez un bouton de barre d'outils personnalisé d'un Online TrendControl. Vous configurez sur ce bouton de barre d'outils un dialogue de sélection créé en propre, avec lequel vous pouvez au choix paramétrer une période de temps sur deux du OnlineTrendControl.

Condition préalable

- Le Graphics Designer est ouvert.
- Une archive est créée dans l'éditeur Tag Logging.

Insérer et configurer WinCC OnlineTrendControl

1. Créez une nouvelle vue de process dans le Graphics Designer.
2. Enregistrez la vue de process sous "OnlineTrend.pdl".
3. Insérez un contrôle "WinCC OnlineTrendControl" dans la vue de process.
4. Sélectionnez dans le menu contextuel du contrôle la commande "Dialogue de configuration...".
   Le dialogue "Propriétés de WinCC OnlineTrendControl" s'ouvre.
5. Dans l'onglet "Courbe" sous "Connexion de données", reliez la courbe avec une variable d'archive.
6. Créez dans l'onglet "Barre d'outils" sous "Fonctions de touche" un nouveau bouton de barre d'outils personnalisé avec l'ID d'objet "1001" pour le OnlineTrendControl.
7. Cliquez sur "Appliquer" pour enregistrer les modifications.
8. Fermez le dialogue en cliquant sur "OK".
9. Sélectionnez dans le menu contextuel du contrôle la commande "Propriétés".
   La boîte de dialogue "Propriétés de l'objet" s'ouvre.
10. Entrez "Control1" comme nom d'objet pour le contrôle.
11. Sélectionnez dans les propriétés d'objet du contrôle "Control1" l'onglet "Événement".
12. Configurez sur l'événement d'objet "OnToolbarButtonClicked" le script VB "Générer l'action VBS sur l'événement "OnToolbarButtonClicked" du bouton de barre d'outils personnalisé (VBS302)".
13. Fermez la boîte de dialogue "Propriétés de l'objet".

Créer une vue de process pour le dialogue de sélection

1. Créez une nouvelle vue de process dans le Graphics Designer.
2. Enregistrez la vue de process sous "Selectiondialog.pdl".
3. Cliquez dans le menu contextuel de la vue de process sur "Propriétés".
   La boîte de dialogue "Propriétés de l'objet" s'ouvre.
4. Paramétrez sous "Géometrie" la valeur "200" pour les attributs "Largeur de vue" et "Hauteur de vue".
5. Fermez la boîte de dialogue "Propriétés de l'objet".
6. Insérez dans la vue de process deux objets "Button".
7. Entrez comme texte pour le bouton "Morning" ou "Afternoon".

Dynamiser le bouton pour le dialogue de sélection

1. Sélectionnez dans les propriétés d'objet du bouton "Morning" l'onglet "Événement".
2. Configurez sur l'événement "Clic de souris" le script VB "Générer l'action VBS sur l'événement "Clic de souris" du bouton "Morning" (VBS303)".
3. Fermez la boîte de dialogue "Propriétés de l'objet".

4. Sélectionnez dans les propriétés d'objet du bouton "Afternoon" l'onglet "Événement".
5. Configurez sur l'événement "Clic de souris" le script VB "Générer l'action VBS sur l'événement "Clic de souris" du bouton "Afternoon" (VBS304)".
6. Fermez la boîte de dialogue "Propriétés de l'objet".

Insérer et configurer une fenêtre de vue

1. Insérez dans la vue de process "OnlineTrend.pdl" un objet "Fenêtre de vue".
2. Sélectionnez dans le menu contextuel de la fenêtre de vue la commande "Propriétés". La boîte de dialogue "Propriétés de l'objet" s'ouvre.
3. Entrez "PictureWindow1" comme nom d'objet pour la fenêtre de vue.
4. Paramérez sous "Autres" l'attribut "Affichage" sur "non".
5. Sélectionnez sous "Autres" pour l'attribut "Nom de vue" la vue de process "Selectiondialog.pdl".
6. Fermez la boîte de dialogue "Propriétés de l'objet".

Générer l'action VBS sur l'événement "OnToolbarButtonClicked" du bouton de barre d'outils personnalisé (VBS302)

```
'VBS302
'Open selection window if Toolbarbutton with ID 1001 is pressed
If lId = 1001 Then
ScreenItems("PictureWindow1").Visible = True
End If
```

Générer l'action VBS sur l'événement "Clic de souris" du bouton "Morning" (VBS303)

```
'VBS303
Dim obj
Set obj = Parent.Parent.ScreenItems("Control1")

'choose time axis, stop update, set begin time and time range
obj.TimeAxisName = "Time axis 1"
obj.TimeAxisActualize = False
obj.TimeAxisBeginTime = CStr(Date & " 4:00:00")
obj.TimeAxisTimeRangeBase = 3600000
obj.TimeAxisTimeRangeFactor = 8

'close the selection window
Parent.Visible = False
```

Générer l'action VBS sur l'événement "Clic de souris" du bouton "Afternoon" (VBS304)

```
'VBS304
Dim obj
Set obj = Parent.Parent.ScreenItems("Control1")

'choose time axis, stop update, set begin time and time range
obj.TimeAxisName = "Time axis 1"
obj.TimeAxisActualize = False
obj.TimeAxisBeginTime = CStr(Date & " 12:00:00")
obj.TimeAxisTimeRangeBase = 3600000
obj.TimeAxisTimeRangeFactor = 8

'close the selection window
Parent.Visible = False
```

Exemple : Vous ajoutez de la sorte des éléments à un WinCC OnlineTrendControl vide

Introduction

Dans l'exemple suivant, sous insérez les éléments Fenêtre de courbe, Axe des valeurs, Axe des temps et Courbes dans un WinCC OnlineTrendControl vide.

Condition préalable

- Le Graphics Designer est ouvert.
- Une archive est créée dans l'éditeur Tag Logging avec trois variables d'archive.

Insérer et configurer WinCC OnlineTrendControl

1. Créez une nouvelle vue de process dans le Graphics Designer.
2. Insérez un contrôle "WinCC OnlineTrendControl" dans la vue de process.
3. Sélectionnez dans le menu contextuel du contrôle la commande "Dialogue de configuration...".
   Le dialogue "Propriétés de WinCC OnlineTrendControl" s'ouvre.
4. Supprimez dans l'onglet "Courbes" dans la zone "Courbes" la fenêtre de courbe par défaut "Courbe 1".
5. Cliquez sur "Appliquer" pour enregistrer les modifications.
6. Fermez le dialogue en cliquant sur "OK".
7. Sélectionnez dans le menu contextuel du contrôle la commande "Propriétés".
   La boîte de dialogue "Propriétés de l'objet" s'ouvre.
8. Entrez "Control1" comme nom d'objet pour le contrôle.
9. Fermez la boîte de dialogue "Propriétés de l'objet".

Insérer et configurer un bouton

1. Insérez dans la vue de process un objet "Button".
2. Entrez comme texte pour le bouton "Insérer éléments".
3. Sélectionnez dans le menu contextuel du bouton la commande "Propriétés".
   La boîte de dialogue "Propriétés de l'objet" s'ouvre.
4. Sélectionnez dans les propriétés d'objet du bouton l'onglet "Événement".
5. Configurez sur l'événement "Clic de souris" le script VB "Générer l'action VBS sur l'événement "Clic de souris" du bouton (VBS305))".
6. Fermez la boîte de dialogue "Propriétés de l'objet".

Générer l'action VBS sur l'événement "Clic de souris" du bouton (VBS305)

```
'VBS305
Dim objTrendControl
Dim objTrendWindow
Dim objTimeAxis
Dim objValueAxis
Dim objTrend

'create reference to TrendControl
Set objTrendControl = ScreenItems("Control1")

'create reference to new window, time and value axis
Set objTrendWindow =
objTrendControl.GetTrendWindowCollection.AddItem("myWindow")
Set objTimeAxis =
objTrendControl.GetTimeAxisCollection.AddItem("myTimeAxis")
Set objValueAxis =
objTrendControl.GetValueAxisCollection.AddItem("myValueAxis")

'assign time and value axis to the window
objTimeAxis.TrendWindow = objTrendWindow.Name
objValueAxis.TrendWindow = objTrendWindow.Name

'add new trend and assign propertys
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myTrend1")
objTrend.Provider = 1
objTrend.TagName = "TestArchive\ArchivTag1"
objTrend.Color = RGB(255,0,0)
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValueAxis.Name

'add new trend and assign propertys
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myTrend2")
objTrend.Provider = 1
objTrend.TagName = "TestArchive\ArchivTag2"
objTrend.Color = RGB(0,255,0)
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
```

```
objTrend.ValueAxis = objValueAxis.Name

'add new trend and assign propertys
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myTrend3")
objTrend.Provider = 1
objTrend.TagName = "TestArchive\ArchivTag3"
objTrend.Color = RGB(0,0,255)
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValueAxis.Name
```

Remarque

Remplacez dans le script VB l'archive utilisée et les variables d'archive "Archive \ArchiveTagX" par les noms de l'archive et des variables d'archive que vous avez créées.

Exemple : Comment ajouter une courbe et une courbe de consigne dans un OnlineTrendControl vide

Introduction

Dans l'exemple suivant, vous ajoutez une courbe et une courbe de consigne dans un WinCC OnlineTrendControl vide. Pour les courbes, un axe de temps et un axe de valeurs est ajouté dans une fenêtre de courbes.

Condition

- Un "WinCC OnlineTrendControl" est ajouté dans le synoptique au niveau du Graphics Designer avec le nom "Control2".
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré pour le bouton l'événement "Clic de souris" avec une action VBS et le script suivant.

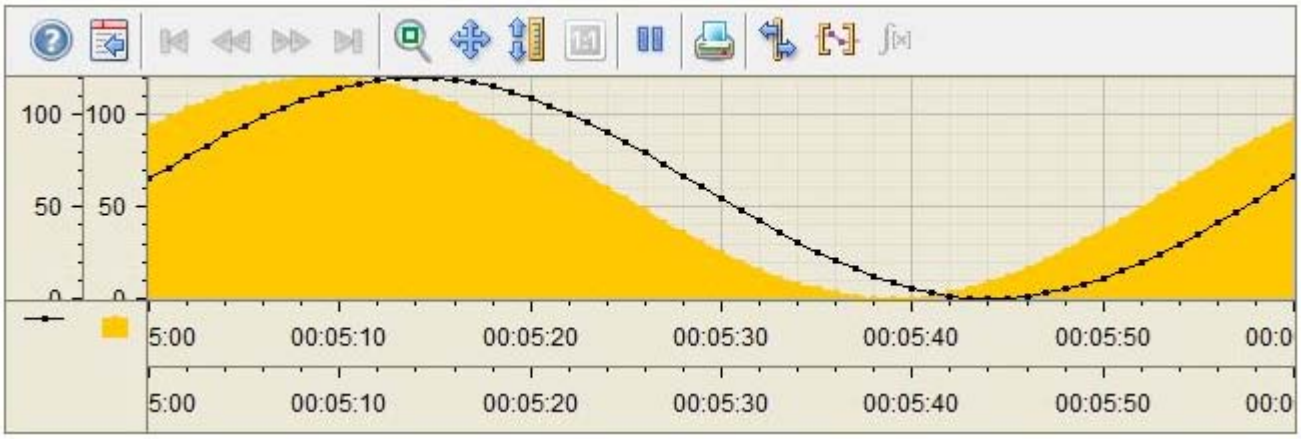
Exemple

```
'VBS352
Dim objTrendControl
Dim objTrendWindow
Dim objTimeAxis
Dim objValueAxis
Dim objTrend
'tags used to generate trend data
Dim dtCurrent
Dim dblCurrent
Dim lIndex
Dim vValues(360)
Dim vTimeStamps(360)

'create reference to TrendControl
```

```
Set objTrendControl = ScreenItems("Control2")

'---- reference trend ----
'create reference to new window, time and value axis
Set objTrendWindow = objTrendControl.GetTrendWindowCollection.AddItem("myWindow")
Set objTimeAxis = objTrendControl.GetTimeAxisCollection.AddItem("myRefTimeAxis")
Set objValueAxis = objTrendControl.GetValueAxisCollection.AddItem("myRefValueAxis")

'assign time and value axis to the window
objTimeAxis.TrendWindow = objTrendWindow.Name
objTimeAxis.ShowDate = False
objValueAxis.TrendWindow = objTrendWindow.Name

'add trend and assign propertys
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myRefTrend")
objTrend.Provider = 0
objTrend.Color = RGB(0,0,0)
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValueAxis.Name

'generate values for reference trend
dtCurrent = CDate("23.11.2006 00:00:00")
For lIndex = 0 To 360
 vValues(lIndex) = ( Sin(dblCurrent) * 60 ) + 60
 vTimeStamps(lIndex) = dtCurrent
 dblCurrent = dblCurrent + 0.105
 dtCurrent = dtCurrent + CDate ("00:00:01")
Next

'insert data to the reference trend
objTrend.RemoveData
objTrend.InsertData vValues, vTimeStamps

'---- data trend ----
'add time and value axis to the existing window
Set objTimeAxis = objTrendControl.GetTimeAxisCollection.AddItem("myTimeAxis")
Set objValueAxis = objTrendControl.GetValueAxisCollection.AddItem("myValueAxis")

'assign time and value axis to the window
objTimeAxis.TrendWindow = objTrendWindow.Name
objTimeAxis.ShowDate = False
objValueAxis.TrendWindow = objTrendWindow.Name

'add new trend and assign propertys
Set objTrend = objTrendControl.GetTrendCollection.AddItem("myTrend")
objTrend.Provider = 0
objTrend.Color = RGB(255,200,0)
objTrend.Fill = True
objTrend.TrendWindow = objTrendWindow.Name
objTrend.TimeAxis = objTimeAxis.Name
objTrend.ValueAxis = objValueAxis.Name

'generate values for data trend
dtCurrent = CDate("23.11.2006 00:00:00")
For lIndex = 0 To 360
```

```
 vValues(lIndex) = ( Sin(dblCurrent) * 60 ) + 60
 vTimeStamps(lIndex) = dtCurrent
 dblCurrent = dblCurrent + 0.106
 dtCurrent = dtCurrent + CDate ("00:00:01")
Next

'insert values to the data trend
objTrend.RemoveData
objTrend.InsertData vValues, vTimeStamps
```

Résultat



Exemple : Comment ajouter des éléments à un WinCC OnlineTableControl

Introduction

L'exemple suivant vous permet d'ajouter des colonnes de variables avec des propriétés dans un WinCC OnlineTableControl vide et de relier les colonnes aux variables d'archive.

Condition

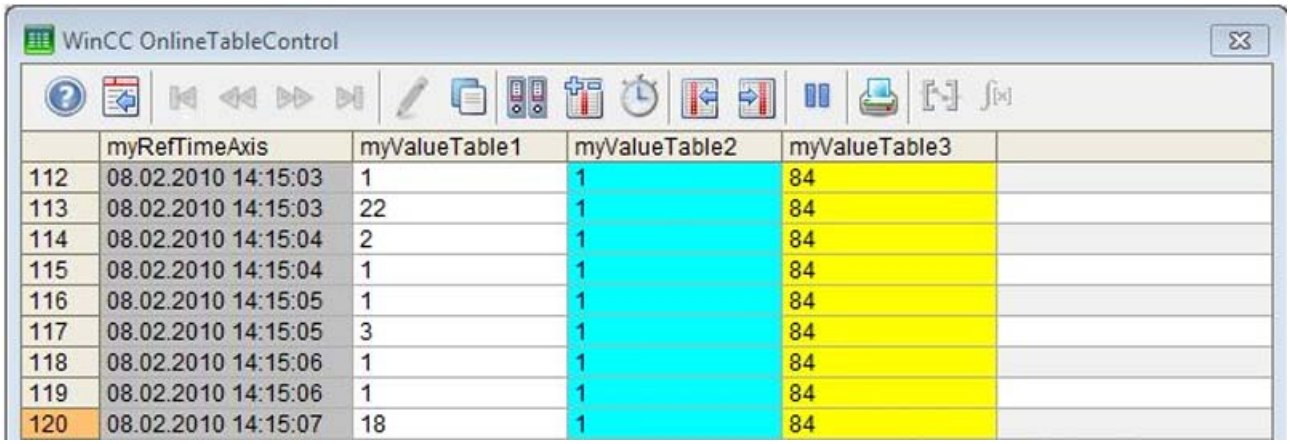
- Une archive est créée dans l'éditeur Tag Logging avec trois variables d'archive.
- Un "WinCC OnlineTableControl" est ajouté dans le synoptique au niveau du Graphics Designer avec le nom "Control2".
- Un bouton est ajouté dans le Graphics Designer. Vous avez configuré pour le bouton l'événement "Clic de souris" avec une action VBS et le script suivant.

Exemple

```
'VBS351
```

```
Dim objTableControl
Dim objTimeColumn
Dim objValueColumn
Dim objTrend

'create reference to TableControl and enable BackColor
Set objTableControl = ScreenItems("Control2")
objTableControl.UseColumnBackColor = True

'create reference to new TimeColumn and assign column length
Set objTimeColumn = objTableControl.GetTimeColumnCollection.AddItem("myRefTimeAxis")
objTimeColumn.Length = 20

'add new ValueColumn and assign propertys
Set objValueColumn = objTableControl.GetValueColumnCollection.AddItem("myValueTable1")
objValueColumn.Provider = 1
objValueColumn.TagName = "Process value archive\PDL_ZT_1"
objValueColumn.BackColor = RGB(255,255,255)
objValueColumn.TimeColumn = objTimeColumn.Name

'add new ValueColumn and assign propertys
Set objValueColumn = objTableControl.GetValueColumnCollection.AddItem("myValueTable2")
objValueColumn.Provider = 1
objValueColumn.TagName = "Process value archive\PDL_ZT_2"
objValueColumn.BackColor = RGB(0,255,255)
objValueColumn.TimeColumn = objTimeColumn.Name

'add new ValueColumn and assign propertys
Set objValueColumn = objTableControl.GetValueColumnCollection.AddItem("myValueTable3")
objValueColumn.Provider = 1
objValueColumn.TagName = "Process value archive\PDL_ZT_3"
objValueColumn.BackColor = RGB(255,255,0)
objValueColumn.TimeColumn = objTimeColumn.Name
```

Résultat



	myRefTimeAxis	myValueTable1	myValueTable2	myValueTable3
112	08.02.2010 14:15:03	1	1	84
113	08.02.2010 14:15:03	22	1	84
114	08.02.2010 14:15:04	2	1	84
115	08.02.2010 14:15:04	1	1	84
116	08.02.2010 14:15:05	1	1	84
117	08.02.2010 14:15:05	3	1	84
118	08.02.2010 14:15:06	1	1	84
119	08.02.2010 14:15:06	1	1	84
120	08.02.2010 14:15:07	18	1	84

Exemple : Scripts pour WinCC AlarmControl

Introduction

Les exemples suivants montrent l'utilisation de scripts pour WinCC AlarmControl.

Condition

- Vous avez déjà configuré des alarmes dans l'éditeur "Alarm Logging".

Exemple1 : Définir un filtre

Un filtre avec le numéro d'alarme "2" est mis à 1 ou à 0 s'il est déjà à 1. L'état est également indiqué dans la fenêtre de diagnostic.

```
'VBS353
Dim objAlarmControl
'create reference to AlarmControl
Set objAlarmControl = ScreenItems("Control1")
'set / reset the filter and create a trace
If (objAlarmControl.MsgFilterSQL = "") Then
 objAlarmControl.MsgFilterSQL = "MSGNR = 2"
 HMIRuntime.Trace "MsgFilterSQL set to MSGNR = 2" & vbNewLine
    Else
 objAlarmControl.MsgFilterSQL = ""
 HMIRuntime.Trace "no filter" & vbNewLine
    End If
```

Exemple2 : ajouter la colonne dans WinCC AlarmContol

La colonne "Texte d'alarme" est ajoutée ou supprimée si la colonne existe déjà. L'état est également indiqué dans la fenêtre de diagnostic. Le bloc d'alarmes de la colonne "Texte d'alarme" porte le nom d'objet "Text1".

```
'VBS354
'add this function to the declaration section
Function IsExistingMsgColumn( objAlarmControl, strName )
'this function checks if the MessageColumn exists
on error resume next
objAlarmControl.GetMessageColumn( strName )
If err.number = 0 Then
 IsExistingMsgColumn = True
else
 err.Clear
 IsExistingMsgColumn = False
end if
End Function

'example code
```

```
Dim objAlarmControl
Dim colMsgColumn
'create reference to the alarm control
Set objAlarmControl = ScreenItems("Control1")
Set colMsgColumn = objAlarmControl.GetMessageColumnCollection
'add or remove the MsgColumn
If ( IsExistingMsgColumn(objAlarmControl, "Text1") ) Then
 HMIRuntime.Trace "Remove MsgColumn" & vbNewLine
 colMsgColumn.RemoveItem("Text1")
    Else
 HMIRuntime.Trace "Add MsgColumn" & vbNewLine
 colMsgColumn.AddItem("Text1")
    End If
```

Exemple3 : Editer le contenu de la fenêtre d'alarme dans la fenêtre de diagnostic

```
'VBS355
Dim objAlarmControl
Dim lIndex
Dim lCellIndex
'create reference to the alarm control
Set objAlarmControl = ScreenItems("Control1")
'enumerate and trace out row numbers
For lIndex = 1 To objAlarmControl.GetRowCollection.Count
 HMIRuntime.trace "Row: " & (objAlarmControl.GetRow(lIndex).RowNumber) & "  "
 'enumerate and trace out column titles and cell texts
 For lCellIndex = 1 To objAlarmControl.GetRow(lIndex).CellCount
  HMIRuntime.trace objAlarmControl.GetMessageColumn(lCellIndex -1).Name & " "
  HMIRuntime.trace objAlarmControl.GetRow(lIndex).CellText(lCellIndex) & "  "
 Next
 HMIRuntime.trace vbNewLine
Next
```

1.15.3 Exemples, généralités

1.15.3.1 Exemples d'ordre général pour VBScript

Introduction

Vous trouverez ici des exemples concernant l'utilisation générale de VBScript pour les rubriques suivantes:

- Programmer l'intégration des données avec VBS
- Appeler des méthodes
- Utiliser l'interface MS-Automation
- Démarrer une application tierce

Remarque

Vous pouvez intégrer dans votre environnement tous les objets qui sont fournis avec Windows Script Host (WSH) de Microsoft, et ce par la méthode standard VBS "CreateObject". De WinCC, VBS ne vous permet toutefois pas d'accéder directement à l'objet WSH.

Exemple 1 : "FileSystemObject"-Objet pour travailler avec le système de fichiers

```
Dim fso, MyFile
Set fso = CreateObject("Scripting.FileSystemObject")
Set MyFile = fso.CreateTextFile("c:\testfile.txt", True)
MyFile.WriteLine("This is a test.")
MyFile.Close
```

Exemple 2 : Objet " WScript.Shell" pour travailler en environnement Windows

Voir aussi

Exemple : Comment configurer une intégration de la base de données avec VBS (Page 820)

Exemple : Pour démarrer une application tierce (Page 824)

Exemple : Comment utiliser l'interface MS Automation (Page 822)

Exemple : Pour appeler les méthodes d'un ActiveX-Control (Page 807)

1.15.3.2 Exemple : Comment configurer une intégration de la base de données avec VBS

Introduction

Les exemples suivants décrivent la configuration d'une liaison à la base de données Access via un pilote ODBC.

- L'exemple 1 décrit une valeur de variable de WinCC dans une base de données Access.
- L'exemple 2 lit une valeur dans la base de données et inscrit cette valeur dans une variable WinCC.

Les exemples n'abordent pas le traitement des erreurs.

Marche à suivre Exemple 1

1. Créer une base de données Access avec tableau WINCC_DATA et colonnes (ID, TagValue) avec ID comme valeur automatique.
2. Configurez la source de données ODBC appelée "SampleDSN", renvoi à la base de données Access mentionnée plus haut.
3. Programmation.

Exemple 1

```
'VBS108
Dim objConnection
Dim strConnectionString
    Dim lngValue
Dim strSQL
Dim objCommand
strConnectionString = "Provider=MSDASQL;DSN=SampleDSN;UID=;PWD=;"
lngValue = HMIRuntime.Tags("Tag1").Read
strSQL = "INSERT INTO WINCC_DATA (TagValue) VALUES (" & lngValue & ");"
Set objConnection = CreateObject("ADODB.Connection")
objConnection.ConnectionString = strConnectionString
objConnection.Open
Set objCommand = CreateObject("ADODB.Command")
With objCommand
    .ActiveConnection = objConnection
    .CommandText = strSQL
End With
objCommand.Execute
Set objCommand = Nothing
objConnection.Close
Set objConnection = Nothing
```

Marche à suivre Exemple 2

1. Créer une variable WinCC portant le nom dbValue.
2. Base de données Access avec tableau WINCC_DATA et colonnes : Créer ID, TagValue (ID comme valeur automatique).
3. Configurez la source de données ODBC appelée "SampleDSN", renvoi à la base de données Access mentionnée plus haut.
4. Programmation.

Exemple 2

```
'VBS108a
Dim objConnection
Dim objCommand
Dim objRecordset
Dim strConnectionString
Dim strSQL
    Dim lngValue
Dim lngCount
strConnectionString = "Provider=MSDASQL;DSN=SampleDSN;UID=;PWD=;"
strSQL = "select TagValue from WINCC_DATA where ID = 1"
Set objConnection = CreateObject("ADODB.Connection")
objConnection.ConnectionString = strConnectionString
objConnection.Open
Set objRecordset = CreateObject("ADODB.Recordset")
```

```
Set objCommand = CreateObject("ADODB.Command")
objCommand.ActiveConnection = objConnection
objCommand.CommandText = strSQL
Set objRecordset = objCommand.Execute
lngCount = objRecordset.Fields.Count
If (lngCount>0) Then
objRecordset.movefirst
lngValue = objRecordset.Fields(0).Value
HMIRuntime.Tags("dbValue").Write lngValue
    Else
HMIRuntime.Trace "Selection returned no fields" & vbNewLine
    End If
Set objCommand = Nothing
objConnection.Close
Set objRecordset = Nothing
Set objConnection = Nothing
```

Il existe diverses possibilités de définir la ConnectionString pour la liaison, selon la source de données utilisée :

Microsoft OLE DB provider for ODBC

Permet les liaisons à une source de données ODBC quelconque. La syntaxe correspondante est la suivante :

```
"[Provider=MSDASQL;]{DSN=name|FileDSN=filename};
[DATABASE=database;]UID=user; PWD=password"
```

Autres Microsoft OLE DB Provider (MS Jet, MS SQL Server p. ex.)

Ils peuvent fonctionner sans DSN. La syntaxe correspondante est la suivante :

```
"[Provider=provider;]DRIVER=driver; SERVER=server;
DATABASE=database; UID=user; PWD=password"
```

Voir aussi

Exemples d'ordre général pour VBScript (Page 819)

1.15.3.3 Exemple : Comment utiliser l'interface MS Automation

Introduction

Les trois exemples suivants vous indiquent comment utiliser l'interface MS Automation.

Exemple 1: MS Excel

Cet exemple écrit la valeur de sortie d'un champ de saisie dans un tableau Excel.

```
'VBS113
Dim objExcelApp
Set objExcelApp = CreateObject("Excel.Application")
objExcelApp.Visible = True
'
'ExcelExample.xls is to create before executing this procedure.
'Replace <path> with the real path of the file ExcelExample.xls.
objExcelApp.Workbooks.Open "<path>\ExcelExample.xls"
objExcelApp.Cells(4, 3).Value = ScreenItems("IOField1").OutputValue
objExcelApp.ActiveWorkbook.Save
objExcelApp.Workbooks.Close
objExcelApp.Quit
Set objExcelApp = Nothing
```

Exemple 2: MS Access

Cet exemple ouvre un Report de MS Access.

```
'VBS114
Dim objAccessApp
Set objAccessApp = CreateObject("Access.Application")
objAccessApp.Visible = True
'
'DbSample.mdb and RPT_WINCC_DATA have to create before executing
'this procedure.
'Replace <path> with the real path of the database DbSample.mdb.
objAccessApp.OpenCurrentDatabase "<path>\DbSample.mdb", False
objAccessApp.DoCmd.OpenReport "RPT_WINCC_DATA", 2
objAccessApp.CloseCurrentDatabase
Set objAccessApp = Nothing
```

Exemple 3: MS Internet Explorer

Cet exemple ouvre MS IE.

```
'VBS115
Dim objIE
Set objIE = CreateObject("InternetExplorer.Application")
objIE.Navigate "http://www.siemens.de"
Do
Loop While objIE.Busy
objIE.Resizable = True
objIE.Width = 500
objIE.Height = 500
objIE.Left = 0
```

```
objIE.Top = 0
objIE.Visible = True
```

Voir aussi

Exemples d'ordre général pour VBScript (Page 819)

1.15.3.4 Exemple : Pour démarrer une application tierce

Introduction

Les deux exemples suivants vous indiquent comment démarrer une application tierce.

Exemple

```
'VBS117
Dim objWshShell
Set objWshShell = CreateObject("Wscript.Shell")
objWshShell.Run "Notepad Example.txt", 1
```

Voir aussi

Exemples d'ordre général pour VBScript (Page 819)

1.16 Principes de base de VBScript

1.16.1 Principes de base de VBScript

Introduction

Les rubriques les plus importantes de Microsoft-VBScript Reference sont les suivantes:

- VBScript Répertoire des langues
- VBScript Tutorium et les principes de base essentiels
- Scripting Référence d'exécution

Si vous avez besoin d'une version intégrale de VBScript Reference, consultez Internet sous

http://msdn2.microsoft.com/en-us/library/t0aew7h6 (http://msdn2.microsoft.com/en-us/library/t0aew7h6)

Voir aussi

Microsoft VBScript Reference (http://msdn2.microsoft.com/en-us/library/t0aew7h6)

1.16.2 VBScript Principes de base

ANSI-C pour la création de fonctions et actions 2

2.1 Création de fonctions et actions avec ANSI-C

Contenu

Les activités d'arrière-plan, telles que par ex. l'impression quotidienne d'un journal, la surveillance de variables ou l'exécution de calculs ne concernant pas des vues, sont exécutées au Runtime par des actions.

Ces actions sont commandées par des déclenchements.

On peut, dans des actions, appeler des fonctions. WinCC propose un grand nombre de fonctions pouvant être modifiées par l'utilisateur. Celui-ci peut également développer ses s.

réer et à éditer des fonctions et des actions.

t"

ic pour l'analyse des problèmes au Runtime.

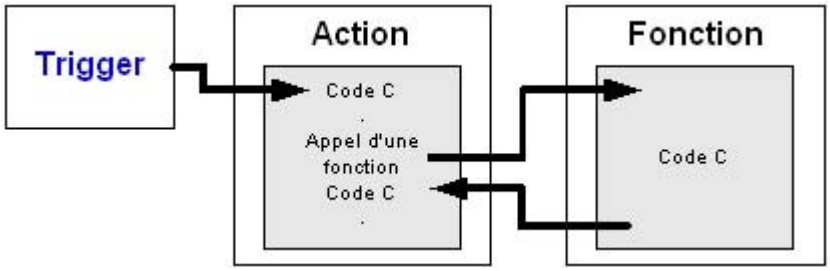
2.2 Création de fonctions et d'actions

Introduction

WinCC offre la possibilité d'utiliser des fonctions et des actions pour dynamiser les processus dans votre projet WinCC. Ces fonctions et actions sont écrites dans la langue ANSI-C.

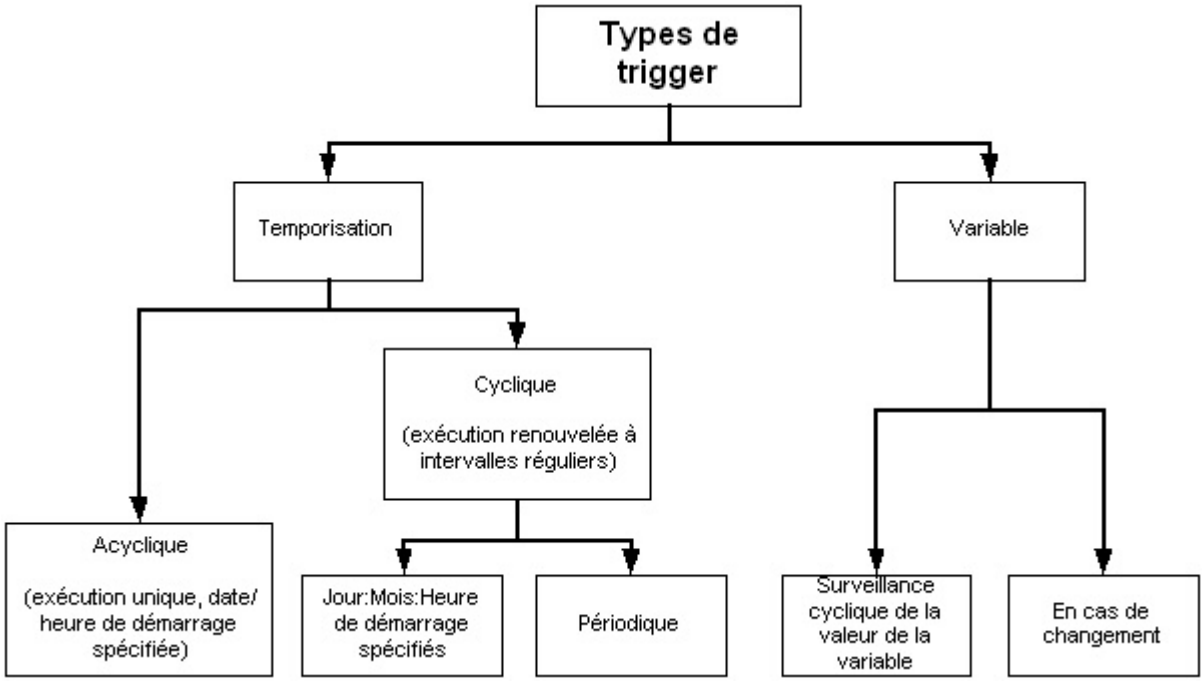
Distinction Fonction - Action

L'exécution des actions est déclenchée par un trigger (événement initiant le processus). Les fonctions ne possèdent pas de déclenchement et sont utilisées comme éléments d'actions, dans les dialogues de dynamisation, dans Tag Logging et dans Alarm Logging.



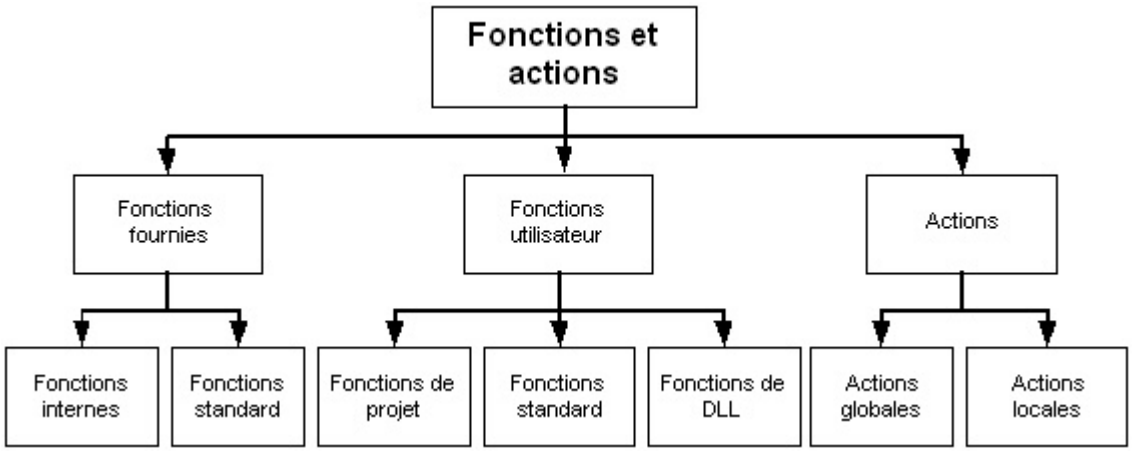
Types de déclenchement

Les types de déclenchement suivants sont disponibles:



Classement des fonctions et des actions

Le diagramme présente un panorama des fonctions et des actions:



Les actions sont utilisées pour des activités ne concernant pas des vues, telles que par ex. l'impression quotidienne d'un journal, la surveillance de variables ou l'exécution de calculs.

Les fonctions sont des petits programmes utilisés en plusieurs endroits mais définis une seule fois. WinCC propose un grand nombre de fonctions. L'utilisateur peut en outre écrire ses propres fonctions et actions.

Il peut aussi modifier les fonctions standard livrées avec WinCC. Lors d'une réinstallation ou d'une mise à jour de WinCC, les fonctions standard modifiées sont supprimées ou remplacées par les fonctions standard livrées. Il faut donc les sauvegarder avant la réinstallation ou la mise à jour.

Outil de développement

Le développement et l'édition des fonctions et actions se font dans l'éditeur Global Script de WinCC. Gobal Script se démarre depuis la fenêtre de navigation de l'explorateur WinCC.

Voir aussi

Comportement des actions au Runtime (Page 890)
Création et édition d'actions (Page 870)
Créer et éditer fonctions (Page 857)
L'éditeur Global Script (Page 842)
Utilisation de DLL dans des fonctions et actions (Page 840)
Utilisation de variables C globales (Page 838)
Pour ajouter Global Script Runtime à la liste de démarrage du projet (Page 837)
Pour générer l'en-tête (Page 852)
Caractéristiques d'actions globales (Page 836)
Caractéristiques d'actions locales (Page 835)
Caractéristiques de fonctions internes (Page 834)
Caractéristiques de fonctions standard (Page 832)
Caractéristiques de fonctions de projet (Page 831)

2.3 Caractéristiques de fonctions de projet

Caractéristiques de fonctions de projet

Les fonctions de projet...

- vous les créez vous-même
- vous pouvez les modifier
- peuvent être protégées par un mot de passe contre les modifications et l'affichage
- ne possèdent pas de déclenchement
- ne sont connues qu'au sein du projet
- ont l'extension "*.fct"

Les fonctions de projet sont rangées dans le sous-répertoire "\library" du projet WinCC.

Utilisation des fonctions de projet

On peut utiliser des fonctions de projet...

- dans d'autres fonctions de projet
- dans des actions Global Script
- (dans Graphics Designer) dans des actions C et dans le dialogue de dynamisation
- (dans Alarm Logging) dans la fonctionnalité Loop in Alarm
- (dans Tag Logging) pour le lancement et l'autorisation d'archives et pour le transfert d'archives cycliques

Voir aussi

Pour protéger une fonction contre les modifications et l'affichage (Page 864)

Créer et éditer fonctions (Page 857)

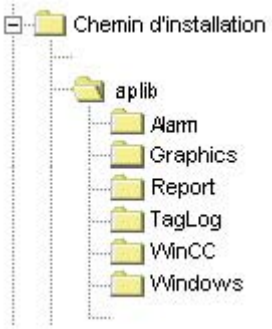
2.4 Caractéristiques de fonctions standard

Caractéristiques de fonctions standard

Les fonctions standard ...

- sont fournies par WinCC
- vous pouvez aussi les créer vous-même
- vous pouvez les modifier
- peuvent être protégées par un mot de passe contre les modifications et l'affichage
- ne possèdent pas de déclenchement
- sont connues de tous les projets
- ont l'extension "*.fct"

Les fonctions standard sont rangées dans les sous-répertoires de "\aplib", dans le répertoire d'installation de WinCC.



Utilisation des fonctions standard

On peut utiliser les fonctions standard...

- dans des fonctions de projet
- dans d'autres fonctions standard
- dans des actions Global Script
- (dans Graphics Designer) dans des actions C et dans le dialogue de dynamisation
- (dans Alarm Logging) dans la fonctionnalité Loop in Alarm
- (dans Tag Logging) pour le lancement et l'autorisation d'archives et pour le transfert d'archives cycliques

Remarque

Les fonctions standard livrées avec WinCC peuvent être modifiées par l'utilisateur. Lors d'une réinstallation ou d'une mise à jour de WinCC, les fonctions standard modifiées sont supprimées ou remplacées par les fonctions standard livrées. Il faut donc les sauvegarder avant la réinstallation ou la mise à jour.

Voir aussi

Pour utiliser des fonctions standard et des fonctions de projet (Page 863)

Créer et éditer fonctions (Page 857)

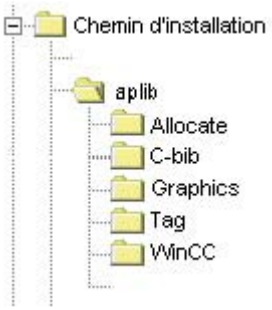
2.5 Caractéristiques de fonctions internes

Caractéristiques de fonctions internes

Les fonctions internes ...

- sont fournies par WinCC
- vous ne pouvez **pas** les créer vous-même
- vous ne pouvez **pas** les modifier
- vous ne pouvez **pas** les renommer
- ne possèdent pas de déclenchement
- sont connues de tous les projets
- ont l'extension "*.icf"

Les fonctions internes sont rangées dans les sous-répertoires de "\aplib", dans le répertoire d'installation de WinCC.



Utilisation des fonctions internes

On peut utiliser des fonctions internes...

- dans des fonctions de projet
- dans des fonctions standard
- dans des actions
- (dans Graphics Designer) dans des actions C et dans le dialogue de dynamisation

2.6 Caractéristiques d'actions locales

Caractéristiques d'actions locales

Les actions locales...

- vous les créez vous-même
- vous pouvez les modifier
- peuvent être protégées par un mot de passe contre les modifications et l'affichage
- possèdent au moins un déclenchement
- ne sont exécutées que sur l'ordinateur affecté
- ont l'extension "*.pas"

Les actions locales sont rangées dans le sous-répertoire "\<Nom de l'ordinateur>\Pas" du répertoire de projet.

Utilisation d'actions locales

Les actions sont utilisées pour des activités ne concernant pas des vues, telles que par ex. l'impression quotidienne d'un journal, la surveillance de variables ou l'exécution de calculs. L'exécution de l'action est déclenchée par le trigger que vous avez configuré. Pour que l'action puisse être exécutée, vous devez ajouter Global Script Runtime à la liste de démarrage.

A la différence des actions globales, les actions locales peuvent être affectées à un ordinateur. Ceci permet de faire en sorte, par exemple, qu'un journal soit imprimé uniquement sur le serveur.

Voir aussi

Pour protéger une action contre les modifications et l'affichage (Page 876)

Déclenchement (Page 879)

Création et édition d'actions (Page 870)

Pour ajouter Global Script Runtime à la liste de démarrage du projet (Page 837)

2.7 Caractéristiques d'actions globales

Caractéristiques d'actions globales

Les actions globales...

- vous les créez vous-même
- vous pouvez les modifier
- peuvent être protégées par un mot de passe contre les modifications et l'affichage
- possèdent au moins un trigger qui déclenche l'exécution
- sont, dans un projet Client-Serveur, exécutées sur tous les ordinateurs du projet
- ont l'extension "*.pas"

Les actions globales sont rangées dans le sous-répertoire "\Pas" du projet WinCC.

Utilisation des actions globales

Les actions sont utilisées pour des activités ne concernant pas des vues, telles que par ex. l'impression quotidienne d'un journal, la surveillance de variables ou l'exécution de calculs. L'exécution de l'action est déclenchée par le trigger que vous avez configuré. Pour que l'action puisse être exécutée, vous devez ajouter Global Script Runtime à la liste de démarrage.

A la différence des actions locales, dans un projet Client-Serveur, les actions globales sont exécutées sur tous les ordinateurs du projet. Dans un projet monoposte il n'y a aucune différence entre actions globales et actions locales.

Voir aussi

Pour protéger une action contre les modifications et l'affichage (Page 876)

Déclenchement (Page 879)

Création et édition d'actions (Page 870)

Pour ajouter Global Script Runtime à la liste de démarrage du projet (Page 837)

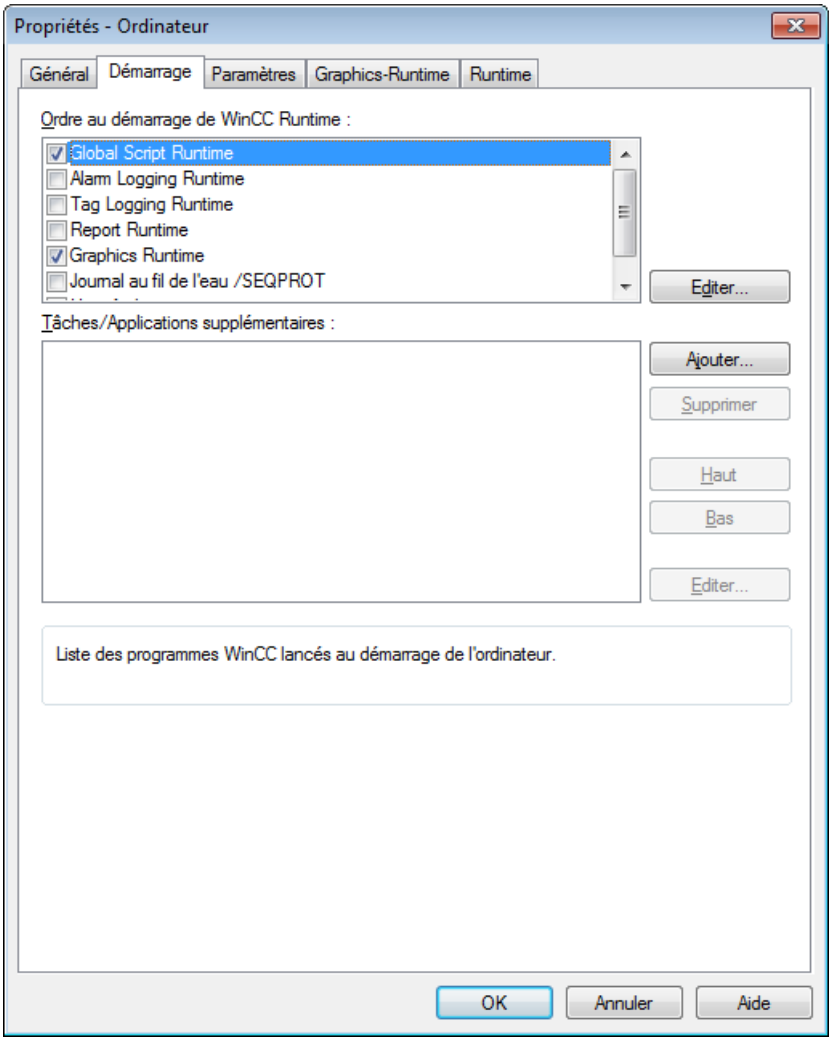
2.8 Pour ajouter Global Script Runtime à la liste de démarrage du projet

Introduction

Pour que les actions Global Script puissent être exécutées au Runtime, vous devez ajouter Global Script Runtime à la liste de démarrage. Ceci n'a aucune influence sur l'exécutabilité des fonctions.

Marche à suivre

1. Dans l'explorateur WinCC, sélectionnez dans le menu contextuel de l'ordinateur, la commande "Propriétés". Le dialogue "Propriétés - Liste des ordinateurs" s'ouvre.
2. Cliquez sur le bouton "Propriétés". Le dialogue "Propriétés - Ordinateur" s'ouvre.
3. Sélectionnez l'onglet "Démarrage"
4. Activez Global Script Runtime.



5. Validez votre saisie par "OK"

2.9 Utilisation de variables C globales

Définition de variables globales C

Pour définir une variable globale C, vous devez placer la ligne de définition devant le nom d'une fonction :

```
int a; //La variable a est définie comme nombre entier
void dummy() //Nom_fonction
{
. //Code de la fonction
}
```

Domaine de validité

Toutes les variables ainsi définies sont connues de toutes les fonctions et actions au Runtime. Elles sont créées au lancement du Runtime même si la fonction elle-même n'a pas été appelée.

Remarque

Lorsque vous utilisez WinCC ServiceMode, il n'existe pas de plage commune de données pour C-Scripting. Il n'est donc pas possible d'échanger de variables globales C entre "Global Script" et "Graphics Designer".

Utilisation de variables C globales

Pour utiliser des variables globales C dans des fonctions ou des actions, vous devez les déclarer comme variables externes dans la fonction ou l'action:

```
void dummy() //Nom_fonction
{
extern int a; //déclaration externe de la variable a
. //Code de la fonction
}
```

Vous signalez ainsi au compilateur qu'il n'a pas à créer la variable "a", mais qu'elle sera créée ailleurs au Runtime.

Lorsque la valeur de la variable "a" est modifiée, cette modification est prise en compte dans toutes les fonctions et actions.

Toute variable C ne doit être définie qu'en un seul endroit. Il est donc recommandé, pour des raisons de clarté et pour éviter toute double définition, de ne définir les variables globales C qu'en un seul endroit.

Remarque

Une fonction et les variables C globales définies avec elle ne disposent au maximum que de 64 Ko.

2.10 Utilisation de DLL dans des fonctions et actions

Adaptation des DLL

WinCC vous permet d'utiliser vos propres DLL (Dynamic Link Libraries).

Pour exploiter les DLL fournies avec les fonctions et actions, vous devez compléter la fonction ou l'action concernée.

Pour cela vous devez ajouter le code suivant au début de la fonction ou de l'action:

```
#pragma code("<Nom>.dll")
<Type de la valeur retournée> <Nom_fonction 1>(...);
<Type de la valeur retournée> <Nom_fonction 2>(...);
  .
  .
  .
<Type de la valeur retournée> <Nom_fonction n>(...);
#pragma code()
```

Les fonctions <Nom_fonction 1> ... <Nom_fonction n> de <Nom.dll> sont ainsi déclarées et peuvent être appelées dans la fonction ou l'action concernée.

Exemple :

```
#pragma code("kernel32.dll")
VOID GetLocalTime(LPSYSTEMTIME lpSystemTime);
#pragma code()

SYSTEMTIME st;

GetLocalTime(&st);
```

Au lieu d'utiliser la méthode qui vient d'être décrite, vous pouvez aussi compléter le fichier d'en-tête (header) Apdefap.h.

Pour utiliser vos propres DLL dans WinCC, vous devez disposer de la version validée. WinCC est livré en version validée et utilise donc aussi la version validée des DLL système. Lorsque vous générez une DLL que vous avez créée vous-même dans la version de test, il peut arriver que les DLL soient chargées en version validée et en version de test. Ceci nécessite alors un espace mémoire plus grand.

Les structures de la DLL doivent être créées avec alignement sur un octet.

Remarque

La DLL doit se trouver dans le répertoire "bin" ou dans le chemin défini avec la variable système "PATH". Cette variable se définit dans les Propriétés système du système d'exploitation.

2.11 L'éditeur Global Script

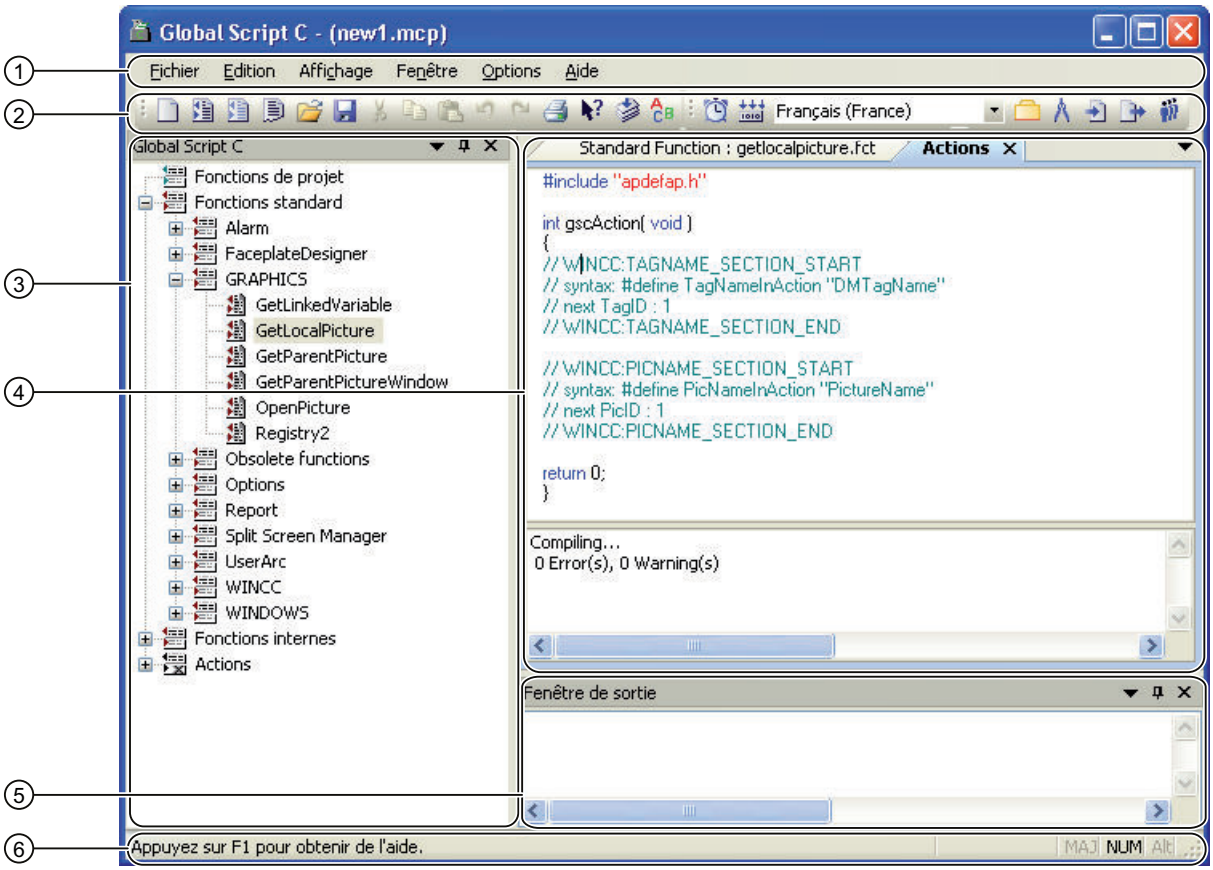
2.11.1 L'éditeur Global Script

Introduction

La création et l'édition de fonctions et d'actions dans WinCC se font dans l'éditeur Global Script. Global Script se lance dans la fenêtre de projet de l'explorateur WinCC.

Conception de l'éditeur Global Script

L'éditeur Global Script est conçu selon les standards Windows. Il possède des barres d'outils, une barre de menu et une barre d'état. Il contient plusieurs fenêtres avec menus déroulants.



Barre de menus (1)
Le contenu de la barre de menus dépend de la situation. Elle est toujours affichée.

Barres d'outils (2)
Global Script comporte deux barres d'outils. Elles peuvent être activées en fonction des

besoins et placées avec la souris en n'importe quel endroit de l'écran.
Les barres d'outils peuvent être affichées et masquées via la commande de menu "Affichage" > "Barres d'outils" et être déplacées dans l'éditeur à l'endroit voulu.

Fenêtre de navigation (3)
La fenêtre de navigation sert à sélectionner des fonctions et actions pour les éditer ou les insérer à la position du curseur dans une fenêtre d'édition. Les fonctions et les actions sont classées par groupes hiérarchisés. Les fonctions sont affichées avec leur nom de fonction, les actions avec leur nom de fichier.

Fenêtre d'édition (4)
La fenêtre d'édition sert à écrire et à éditer fonctions et actions. Elle n'est affichée que lorsqu'une fonction ou une action est ouverte pour l'édition. Chaque fonction ou action est ouverte dans une fenêtre d'édition qui lui est propre. Il est possible d'ouvrir simultanément plusieurs onglets d'édition.

Fenêtre de sortie (5)
Cette fenêtre affiche le résultat des fonctions "Rechercher dans fichiers" ou "Compiler toutes les fonctions". Elle est active par défaut et peut être désactivée en fonction des besoins.

- Recherche dans fichiers :
  La recherche renvoie en résultat dans la fenêtre de sortie une ligne par terme recherché trouvé, cette ligne contenant le numéro de ligne, le chemin et le nom du fichier, ainsi que le texte de la ligne dans laquelle la chaîne recherchée a été trouvée, avec le numéro de cette ligne. Vous pouvez ouvrir directement le fichier correspondant en double-cliquant sur une indication dans la fenêtre de sortie. Le curseur est alors placé dans la ligne dans laquelle se trouve la chaîne recherchée.
- Compilation de toutes les fonctions:
  Tous les avertissements (warnings) et messages d'erreur du compilateur sont affichés pour chaque fonction compilée. Le chemin et le nom du fichier de la fonction compilée sont affichés dans la ligne suivante avec le résumé de la compilation.

Barre d'état (6)
La barre d'état est ancrée en bas de la fenêtre Global Script et peut être masquée ou affichée. Elle contient les informations sur la position du curseur dans la fenêtre d'édition et sur le réglage du clavier. Une brève information sur la fonctionnalité Global Script sélectionnée ou une étiquette est également affichée.

Ancrage des fenêtres

L'ancrage des fenêtres permet de disposer les différentes fenêtres de manière flexible. Vous avez la possibilité de déplacer les fenêtres, de les positionner librement en tant que fenêtres autonomes ou de les regrouper dans des groupes d'onglets. Vous pouvez, par exemple, positionner vos actions les unes à côté des autres, les unes sous les autres ou sous forme d'onglets. Vous pouvez masquer automatiquement des fenêtres et les réafficher en cas de besoin.
Pour plus d'informations, reportez-vous au chapitre "Créer des vues de process".

Voir aussi

Imprimer des fonctions et actions (Page 855)
Pour rechercher un texte dans des fichiers (Page 854)
Pour compiler toutes les fonctions (Page 853)
Pour générer l'en-tête (Page 852)
Pour supprimer des actions ou des fonctions de projet ou standard (Page 851)
Pour utiliser "Enregistrer sous..." (Page 850)
Pour paramétrer la police (Page 850)
Pour paramétrer plusieurs vues (Page 849)
Utiliser les barres d'outils (Page 847)
Utiliser la fenêtre d'édition (Page 844)

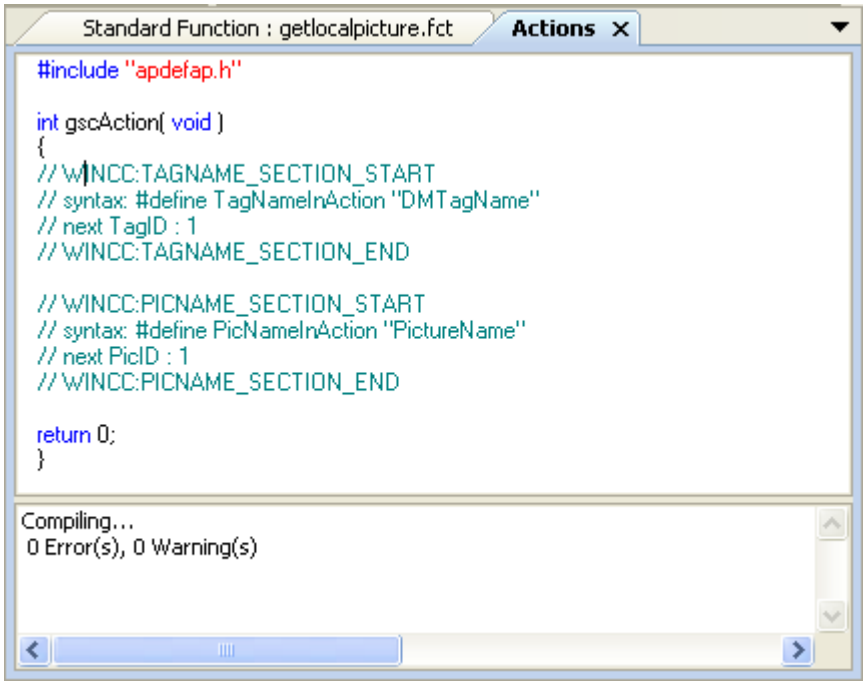
2.11.2 Utiliser la fenêtre d'édition

2.11.2.1 Utiliser la fenêtre d'édition

Introduction

La fenêtre d'édition propose des fonctions d'édition qui s'exécutent au clavier ou avec la souris.

La fenêtre d'édition sert à éditer des fonctions et des actions.



Contenu

La fenêtre est divisée en deux parties. La partie supérieure de la fenêtre d'édition affiche le code de la fonction ou de l'action. La partie inférieure affiche les messages du compilateur pendant la compilation de la fonction ou de l'action.

Propriétés

Lorsque vous ouvrez la fenêtre pour la première fois, la partie inférieure est réduite. Elle s'agrandit au lancement d'une compilation pour pouvoir afficher les messages du compilateur. Vous pouvez ajuster la division de la fenêtre avec la souris. Lorsque vous double-cliquez sur un message d'erreur, vous basculez dans la ligne correspondante du code.

Code couleurs

Le code couleurs pour l'affichage du code C est le suivant:

Couleur	Signification	Exemple
bleu	Mots clés	define, double, if
vert	Commentaires	/??/ Ceci est un commentaire??
rouge	Chaînes de caractères	"Rectangle3"
noir	Autre élément de code C	level=100*newvalue/255;

Remarque

Une fonction ou action ne peut contenir plus de 32767 caractères, les blancs compris.

Voir aussi

Fonctions d'édition avec la souris (Page 846)

Fonctions d'édition avec le clavier (Page 845)

2.11.2.2 Fonctions d'édition avec le clavier

Le clavier permet d'exécuter les fonctions d'édition suivantes :

Actions d'édition	Commande clavier
Insertion/Ecrasement	<Inser>
insérer une ligne supplémentaire	<ENTREE>
Supprimer un caractère à droite	<SUPPR>
Supprimer un caractère à gauche	<RET.ARR>
Supprimer texte marqué	<Suppr> ou <RET.ARR>
Saut en début de ligne	<Pos1>
Saut en fin de ligne	<Fin>

Actions d'édition	Commande clavier
Saut au début du texte	<Ctrl+Pos1>
Saut en fin de texte	<Ctrl+Fin>
Déplacement du curseur	<Touches CURSEUR>
Déplacement du curseur d'une hauteur de fenêtre vers le début du texte	<Pg Up>
Déplacement du curseur d'une hauteur de fenêtre vers la fin du texte	<Pg Dn>
Placement du curseur sur la première ligne de la fenêtre	<Ctrl+Pg Up>
Placement du curseur sur la dernière ligne de la fenêtre	<Ctrl+Pg Dn>
Saut à la position de tabulateur suivante vers la gauche	<TAB>
Couper le texte marqué et le copier dans le presse-papiers	<Ctrl+X>
Copier le texte marqué dans le presse-papiers	<Ctrl+C>
Coller le texte du presse-papiers	<Ctrl+V>

2.11.2.3 Fonctions d'édition avec la souris

La souris permet d'exécuter les fonctions d'édition suivantes :

Actions d'édition	Action avec la souris (touche gauche de la souris)
Sélectionner du texte	Faire glisser le curseur sur le texte avec la touche de la souris enfoncée
Sélectionner un mot	Double-clic sur le mot
Sélectionner une ligne	Triple-clic sur la ligne
Sélection étendue	<MAJ> + clic
Placer le curseur	Clic
Déplacer le texte sélectionné	Glisser-déplacer avec la touche de la souris enfoncée
Dupliquer le texte sélectionné	<Ctrl> + glisser avec la touche de la souris enfoncée

Autres fonctions d'édition :

- Avec un double-clic sur un message d'erreur du compilateur, la ligne correspondante est montrée en format balisé.
- Un clic avec la touche de droite de la souris ouvre le menu contextuel.

Lors des actions suivantes, le texte sélectionné est remplacé par le résultat de l'action :

- Saisie de caractères avec le clavier
- Insertion du contenu du presse-papiers
- Appel d'une fonction via l'affectation de paramètres

2.11.3 Utiliser les barres d'outils

Fonction

La position standard des barres d'outils est sous la barre de menu, en haut de la fenêtre Global Script. Les boutons des barres d'outils permettent d'accéder rapidement et confortablement aux fonctionnalités de Global Script.

Les deux barres d'outils ci-dessous sont disponibles :

Barre d'outils "Standard"

Barre d'outils "Edition"

Sommaire

La barre d'outils "Standard" contient des boutons pour les fonctions suivantes :

Bouton	Fonction	Combinaison de touches
	Créer une nouvelle action.	<Alt+A> ou <Ctrl +N>
	Créer une nouvelle fonction standard.	<Alt+S> ou <Ctrl +N>
	Créer une nouvelle fonction de projet.	<Alt+P> ou <Ctrl +N>
	Créer un nouveau fichier d'en-tête.	<Ctrl+H>
	Ouvrir une action ou une fonction existante.	<CTRL+O>
	Enregistrer le contenu de la fenêtre d'édition active. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte.	<CTRL+S>
	Coupe le texte marqué et le copie dans le presse-papiers. La fonction n'est disponible que lorsqu'un texte est marqué.	<CTRL+X>
	Copie dans le presse-papiers le texte marqué. La fonction n'est disponible que lorsqu'un texte est marqué.	<CTRL+C>

Bouton	Fonction	Combinaison de touches
	Insère le contenu du presse-papiers à la position du curseur. Cette fonction n'est disponible que si le presse-papiers n'est pas vide.	<CTRL+V>
	Annuler la dernière de 30 actions d'édition (maxi.). La fonction n'est disponible que si une action d'édition a été exécutée.	<Ctrl+Z>
	Restaurer la dernière action d'édition annulée. La fonction n'est disponible que lorsqu'une action d'édition a été annulée.	<Ctrl+A>
	Imprimer le contenu de la fenêtre d'édition active comme documentation de projet. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte.	<Ctrl+P>
	Activer l'Aide directe.	<MAJ+F1>
	Générer l'en-tête.	<Ctrl+G>
	Paramètre la police.	<Ctrl+F>

Remarque

La combinaison de touches <Ctrl+N> n'est disponible que si une fenêtre d'édition au moins est ouverte. Lorsque le contenu de la fenêtre d'édition active est une fonction, <Ctrl+N> crée une nouvelle fonction de projet. Lorsque le contenu de la fenêtre d'édition active est une action, une nouvelle action globale est créée.

La barre d'outils "Edition" contient des boutons pour les fonctions suivantes :

Bouton	Fonction	Combinaison de touches
	Insérer des informations sur la fonction et, pour les actions, créer en plus un déclenchement. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte.	<Ctrl+I>
	Compiler le code dans la fenêtre d'édition active. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte.	<MAJ+F8>
Français (France)	Règle la page de code appropriée. Veillez à ce que le choix de la page de code soit compatible avec le texte source. On ne peut utiliser qu'une langue à la fois dans le texte source.	-

Bouton	Fonction	Combinaison de touches
	Ouvrir le dialogue de variable. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte.	<Ctrl+R>
	Ouvrir un dialogue de sélection de vue. Le nom de la vue sélectionnée est placé, dans la fenêtre d'édition, à la position du curseur dans le code. La fonction n'est disponible que lorsqu'une fenêtre d'édition est ouverte.	<Ctrl+W>
	Importer une action. La fonction n'est disponible que lorsque la fenêtre active contient une action.	<Ctrl+M>
	Exporter l'action dans la fenêtre d'édition active. La fonction n'est disponible que lorsque la fenêtre active contient une action.	<Ctrl+T>
	Régler l'autorisation de conduite pour l'action. La fonction n'est disponible que lorsque la fenêtre active contient une action.	<Ctrl+E>

Attributs

Les deux barres d'outils peuvent être affichées et masquées.

Elles peuvent être accostées sous la barre de menu.

Désaccostées, elles peuvent être positionnées avec la souris en n'importe quel endroit de l'écran.

Voir aussi

Pour paramétrer plusieurs vues (Page 849)

2.11.4 Pour paramétrer plusieurs vues

Introduction

"Vues" signifie ici les diverses combinaisons dans l'éditeur Global Script des éléments, activés, suivants : Fenêtre d'affichage, Barre d'état et Barres d'outils. L'affichage de ces éléments peut être activé ou désactivé individuellement.

Dans le paramétrage par défaut tous les éléments sont affichés.

Marche à suivre

1. Ouvrez le menu "Affichage" dans la barre de menu de Global Script
2. Activez ou désactivez l'affichage de l'élément souhaité, par ex. de la barre d'outils. Lorsque l'affichage est activé, ceci est signalé par une coche devant le nom.

Remarque

Lorsque vous redémarrez Global Script, le paramétrage par défaut est réactivé et tous les éléments sont affichés.

2.11.5 Pour paramétrer la police

Introduction

La police se définit en paramétrant la "police", le "style" et la "taille".

La police paramétrée est active dans toutes les fenêtres d'édition.

Marche à suivre

1. Cliquez sur le bouton de la barre d'outils "Standard" et ouvrez le dialogue de la police.
2. Effectuez les paramétrages souhaités.
3. Validez les paramétrages par "OK"

Autre manipulation possible

Vous pouvez également ouvrir le dialogue de paramétrage de la police de la manière suivante:

Ouvrez le menu "Options" dans la barre de menu de Global Script et sélectionnez l'option "Police" ou utilisez la combinaison de touches correspondante.

Remarque

Les paramétrages effectués sont automatiquement enregistrés et sont disponibles après un redémarrage de WinCC.

2.11.6 Pour utiliser "Enregistrer sous..."

Introduction

Lorsque vous créez une fonction ou une action, Global Script crée un fichier correspondant, dans un chemin prescrit, sous un nom de fichier par défaut, par ex. "nouvelle_fonction_1.fct"

pour les fonctions et "gsccs1.pas" pour les actions. Ce nom de fichier par défaut étant peu évocateur, vous pouvez enregistrer la fonction ou l'action sous un autre nom avec "Enregistrer sous...".

"Enregistrer sous..." ne modifie que le nom du fichier, le nom de la fonction ou de l'action restant inchangé.

Global Script s'attend à ce que le répertoire d'enregistrement de la fonction ou de l'action à enregistrer soit un répertoire de projet. Si ce n'est pas le cas, ceci est signalé mais le fichier est quand même enregistré.

Condition

"Enregistrer sous.." n'est disponible que si une fenêtre d'édition au moins est ouverte, le contenu de la fenêtre d'édition active étant alors enregistré.

Marche à suivre

1. Ouvrez le menu "Fichier" dans la barre de menu de Global Script
2. Sélectionnez la commande "Enregistrer sous..."
3. Saisissez le nouveau nom de fichier
4. Fermez la fenêtre de dialogue par le bouton "Enregistrer"

Voir aussi

Pour supprimer des actions ou des fonctions de projet ou standard (Page 851)

2.11.7 Pour supprimer des actions ou des fonctions de projet ou standard

Introduction

Vous pouvez supprimer les actions ou les fonctions de projet et standard pendant la configuration ou au Runtime. Global Script supprime l'entrée dans la fenêtre de navigation et le fichier correspondant.

Si une fonction supprimée est appelée par une action, l'action est interrompue à l'endroit de l'appel de la fonction.

Un message est affiché si la fenêtre de diagnostic de Global Script est ouverte. En outre l'interruption de l'action est consignée dans le fichier de diagnostic WinCC_Sys_xx.log (xx = n ° consécutif). Ce fichier de diagnostic se trouve dans le sous-répertoire "Diagnostic" du répertoire d'installation de WinCC.

Marche à suivre

1. Dans la fenêtre de navigation de Global Script, ouvrez le menu contextuel de la fonction ou de l'action à supprimer.
2. Sélectionnez la commande "Supprimer"
3. Confirmez la question concernant la sécurité par "Oui"

Autre manipulation possible

Au lieu d'utiliser le menu contextuel, vous pouvez supprimer avec la touche <Suppr> la fonction ou l'action marquée.

Remarque

Lorsqu'une fonction est supprimée, l'entrée dans le fichier d'en-tête correspondant est également supprimée.

2.11.8 Pour générer l'en-tête

Introduction

l'en-tête doit être générée dans les cas suivants :

- Vous avez copié des fonctions de projet d'un autre projet dans votre chemin de projet, dans le répertoire "library".
- Vous avez copié des fonctions standard d'un autre PC dans le répertoire "aplib" (ou sous-répertoires).

La génération de l'en-tête déclare les fonctions copiées dans les fichiers d'en-tête correspondants. Après cela, vous pouvez utiliser les fonctions dans le projet.

Marche à suivre

1. Activez le bouton de la barre d'outils "Standard".

Autre manipulation possible

Vous pouvez aussi générer l'en-tête de la manière suivante:

Ouvrez le menu "Options" et sélectionnez la commande "Générer en-tête" ou utilisez la combinaison de touches correspondante.

Remarque

Le contenu de la fenêtre de navigation est rafraîchi à l'issue de la génération.

Si WinCC se trouve en mode Runtime, le système Runtime n'est pas affecté par la nouvelle génération de l'en-tête.

2.11.9 Pour compiler toutes les fonctions

Introduction

Si des modifications manuelles ont été apportées aux fichiers d'en-tête, toutes les fonctions doivent être de nouveau compilées. Avec la commande de menu "Compiler toutes les fonctions", toutes les fonctions de projet, standard et internes sont compilées automatiquement.

Si des fonctions sont appelées dans d'autres fonctions, des messages d'erreur peuvent apparaître. La raison de cette erreur est que les fonctions appelées ne sont pas encore compilées à ce moment. Ces fonctions doivent ensuite être compilées individuellement.

Condition

La fonction n'est disponible que lorsque toutes les fenêtres d'édition sont fermées.

Marche à suivre

1. Ouvrir le menu "Options"
2. Sélectionner l'entrée "Compiler toutes les fonctions"

Autre manipulation possible

Il est également possible de lancer la compilation de toutes les fonctions avec la combinaison de touches <Alt+U>.

Résultat

Les résultats des différentes opérations de compilation et les messages d'erreur sont affichées dans la fenêtre de sortie du compilateur. De plus, le chemin et le nom du fichier de la fonction compilée sont affichés dans la ligne suivante avec le résumé de la compilation.

Remarque

Dans un projet multiposte, la fonction "Compiler toutes les fonctions" n'est pas disponible. L'affectation des fonctions n'est plus possible pour ce projet.

Sur un ordinateur WinCC, les fonctions compilées ne sont effectives qu'au prochain démarrage de Runtime.

2.11.10 Pour rechercher un texte dans des fichiers

Introduction

La chaîne de caractères saisie est recherchée, dans la fenêtre de navigation, dans tous les fichiers du groupe sélectionné.

Le résultat de la recherche est présenté dans la fenêtre d'affichage:

Une ligne est créée dans la fenêtre d'affichage pour chaque chaîne trouvée. Elle comprend le numéro de la ligne de code dans laquelle la chaîne a été trouvée, le chemin, le nom du fichier et la ligne de code elle-même.

Vous pouvez double-cliquer sur le résultat de la recherche pour ouvrir les fonctions standard ou les fonctions de projet ainsi que les actions. Le curseur est positionné au début de la ligne qui contient la chaîne recherchée. Dans les fonctions internes, la fonction contenant la chaîne de caractères est présentée dans la fenêtre de navigation puis marquée.

Marche à suivre

1. Dans la fenêtre de navigation de Global Script, ouvrez le menu contextuel du groupe dans lequel doit se faire la recherche.
2. Sélectionnez "Rechercher dans fichiers"
3. Saisissez dans la fenêtre de dialogue le texte à rechercher
4. Lancez la recherche en confirmant avec "Rechercher". Le résultat de la recherche est présenté dans la fenêtre d'affichage

2.11.11 Imprimer des fonctions et actions

2.11.11.1 Imprimer des fonctions et actions

Introduction

Vous pouvez imprimer les fonctions de projet, les fonctions standard et les actions en utilisant des modèles d'impression système.

Pour cela la fonction ou l'action à imprimer doit se trouver dans une fenêtre d'édition. Le contenu de la fenêtre d'édition active est imprimé.

Un aperçu avant impression peut être affiché.

Des paramètres d'impression permettent de piloter l'impression.

Les modèles d'impression utilisés sont:

- @gsc_pfc.rpl pour les fonctions de projet
- @gsc_pfc.rpl pour les fonctions standard
- @gsc_act.rpl pour les actions

Voir aussi

Pour imprimer une documentation de projet (Page 856)

Pour ouvrir l'aperçu avant impression (Page 856)

Pour définir les paramètres d'impression (Page 855)

2.11.11.2 Pour définir les paramètres d'impression

Introduction

Vous pouvez utiliser les paramètres suivants pour l'impression:

- Paramétrage d'un modèle d'impression autre que le modèle par défaut
- Pages
- Sélection de l'imprimante
- Impression dans un fichier

Condition

Une fenêtre d'édition au moins doit être ouverte.

Marche à suivre

1. Ouvrez le menu "Fichier" dans la barre de menu de Global Script
2. Sélectionnez la commande "Configurer documentation de projet..."
3. Effectuez les paramétrages souhaités dans la fenêtre de dialogue ouverte.
4. Validez les paramétrages par "OK"

Remarque

Les paramétrages effectués sont automatiquement enregistrés et sont disponibles après un redémarrage de WinCC.

2.11.11.3 Pour ouvrir l'aperçu avant impression

Introduction

Il peut être bon, avant d'imprimer une fonction ou une action, de l'afficher avec la fonction d'aperçu avant impression.

Le contenu de la fenêtre d'édition active est affiché dans la vue d'aperçu avant impression.

Marche à suivre

1. Ouvrez le menu "Fichier" dans la barre de menu
2. Sélectionnez la commande "Afficher documentation de projet"

2.11.11.4 Pour imprimer une documentation de projet

Introduction

Vous pouvez imprimer le contenu de la fenêtre d'édition active sur l'imprimante ou dans un fichier. Les paramètres d'impression définis sont alors utilisés.

Marche à suivre

1. Ouvrez le menu "Fichier" dans la barre de menu de Global Script
2. Sélectionnez la commande " Imprimer documentation de projet "

2.12 Créer et éditer fonctions

2.12.1 Créer et éditer fonctions

Introduction

On distingue les fonctions de projet, les fonctions standard et les fonctions internes. Un grand nombre de fonctions standard et de fonctions internes prêtes à l'emploi sont fournies avec WinCC. Vous pouvez en outre écrire vos propres fonctions de projet ou fonctions standard ou modifier des fonctions standard existantes. Les fonctions standard fournies avec WinCC seront cependant écrasées à la prochaine installation de WinCC.

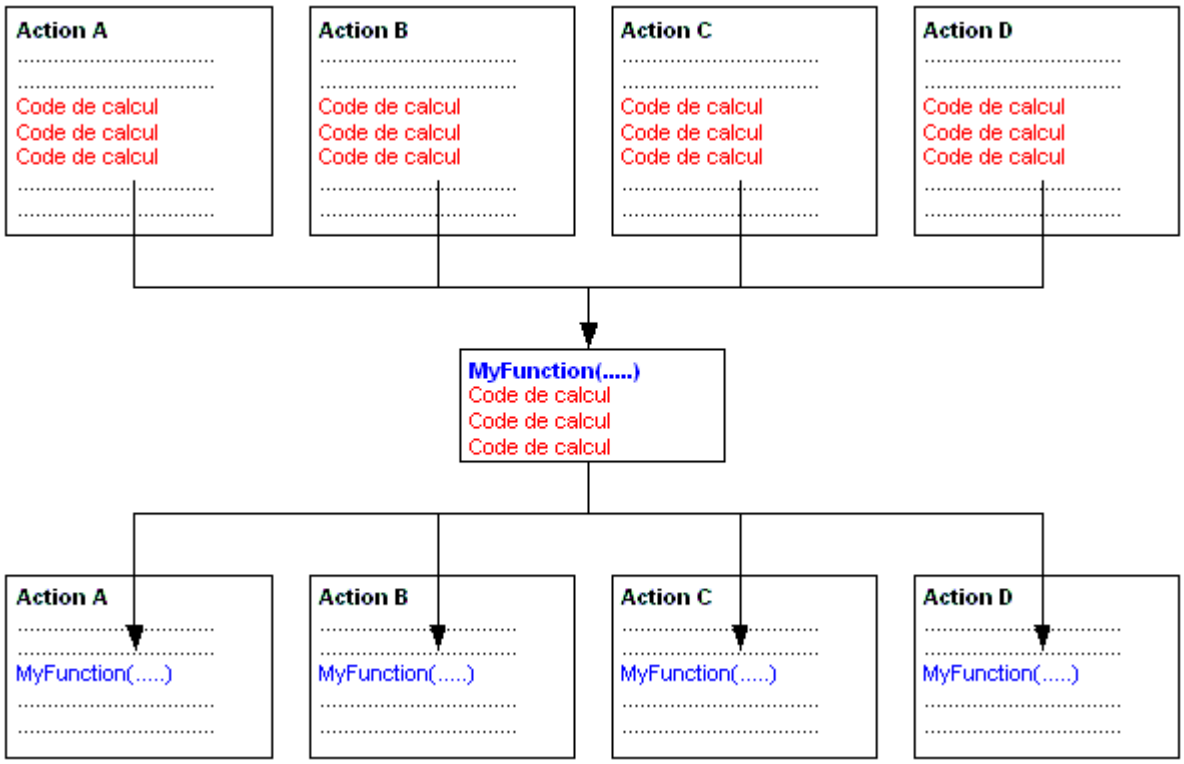
Les fonctions internes ne peuvent être ni créées ni modifiées.

Utilisation de fonctions

Lorsqu'un calcul, par ex., doit être effectué avec des valeurs de départ différentes dans plusieurs actions, il est judicieux de programmer ce calcul dans une fonction. La fonction n'est alors plus appelée qu'avec ses paramètres actuels dans les différentes actions.

Ceci présente les avantages suivants:

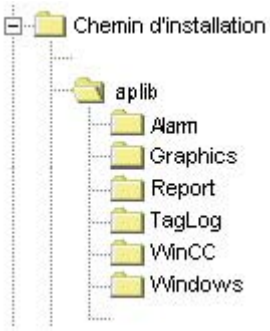
- Le calcul n'est programmé qu'une seule fois
- Les modifications sont faites en un seul endroit, dans la fonction et pas dans chaque action
- Le code des actions est plus court et plus clair



Accès aux fonctions

Vous pouvez accéder aux fonctions existantes ou créer de nouvelles fonctions par la fenêtre de navigation de Global Script, par le menu "Fichier" ou par le bouton de la barre d'outils.

Les fonctions sont rangées de la manière suivante dans le système de fichiers:



Edition et compilation de fonctions

Une fonction s'édite et se compile dans sa propre fenêtre d'édition. Une fois la compilation exécutée, la fenêtre d'édition affiche les messages du compilateur. Il peut s'agir de messages d'avertissement ou d'erreur. Dans tous les cas, le nombre des avertissements et des erreurs est indiqué.

Que se passe-t-il si je modifie le nom de la fonction?

Les fonctions sont toujours affichées dans la fenêtre de navigation avec leur nom de fonction, pas avec le nom de fichier Windows. Si vous modifiez le nom d'une fonction dans la fenêtre d'édition et exécutez une compilation, le nom affiché dans la fenêtre de navigation ne correspond plus au nom de la fonction. Ceci est indiqué par un signe "*" devant le nom. Dès que vous enregistrez la fonction, le nom actuel de la fonction est affiché dans la fenêtre de navigation.

Il faut s'en souvenir lors de l'enregistrement de fonctions.

Lorsque vous enregistrez une fonction pour laquelle une erreur de compilation est apparue ou une fonction pas compilée du tout, elle est signalée par le symbole dans la fenêtre de navigation.

Lorsque vous enregistrez une fonction compilée sans erreur, elle est signalée par le symbole dans la fenêtre de navigation.

Voir aussi

Utiliser la fenêtre d'édition (Page 844)

Pour utiliser des fonctions empruntées à d'autres sources (Page 868)

Pour renommer une fonction (Page 867)

Pour compiler et enregistrer une fonction (Page 865)

Pour protéger une fonction contre les modifications et l'affichage (Page 864)

Pour ajouter des informations supplémentaires à la fonction (Page 864)

Pour utiliser des fonctions standard et des fonctions de projet (Page 863)

Pour utiliser des fonctions internes (Page 862)

Pour écrire le code d'une fonction (Page 861)

Pour créer une nouvelle fonction (Page 860)

Caractéristiques de fonctions standard (Page 832)

Caractéristiques de fonctions de projet (Page 831)

2.12.2 Pour créer une nouvelle fonction

Introduction

La procédure est la même pour les fonctions de projet et les fonctions standard. Dans la fenêtre de navigation, créez le type de fonction (de projet ou standard) et, pour les fonctions standard, le groupe, par ex. "graphics". Le lieu de stockage dans le système de fichiers est ainsi également déterminé.

Global Script propose pour la nouvelle fonction un nom par défaut, par ex. "nouvelle_fonction_3". Ce nom est également le nom du fichier. Pour que les noms de fonction soient uniques, les noms proposés comprennent un numéro de série.

L'utilisateur remplacera généralement ce nom par défaut par un nom de fonction plus évocateur. Vous pouvez changer le nom de fichier lors du premier enregistrement de la fonction renommée.

Lors de la création de la fonction, Global Script lui ajoute les informations de date de création, de date de modification et de version. Ces informations peuvent être affichées dans le dialogue "Propriétés". Ce dialogue permet en outre de protéger la fonction par un mot de passe contre les modifications et l'affichage. Le dialogue s'ouvre avec le bouton .

Remarque

Les caractères aussi pris en charge par ANSI-C sont acceptés dans les noms de fonctions :

- Lettres, à l'exception des caractères spéciaux nationaux
- Chiffres
- Trait de soulignement

Marche à suivre

1. Dans la fenêtre de navigation, ouvrez le menu contextuel du groupe souhaité
2. Sélectionnez "Nouveau"
   Si une nouvelle fonction a été créée, la fenêtre d'édition correspondante contient comme première ligne de code le type de la valeur retournée et le nom par défaut de la nouvelle fonction. La parenthèse qui suit peut contenir le cas échéant des paramètres de transfert. Le code de la fonction est placé entre accolades.

Autre manipulation possible

Vous pouvez également créer une nouvelle fonction avec l'icône de la barre d'outils, le menu "Fichier" ou la combinaison de touches correspondantes.

Voir aussi

Pour ajouter des informations supplémentaires à la fonction (Page 864)

Pour utiliser "Enregistrer sous..." (Page 850)

2.12.3 Pour écrire le code d'une fonction

Introduction

Le code de la fonction s'écrit dans la fenêtre d'édition. Le langage de programmation est ANSI-C.

Chaque fonction de projet et fonction standard peut appeler d'autres fonctions dans son code. Il peut s'agir de fonctions de projet, de fonctions standard, de fonctions internes ou de fonctions DLL. Pour que la fonction appelée soit connue de la fonction appelante, vous devez ajouter la ligne #include "apdefap.h" comme première ligne du code de la fonction appelante.

La fenêtre de navigation contient la bibliothèque de fonctions C sous "Fonctions internes", "c_bib" .

La première ligne de code contient le type de la valeur retournée et le nom par défaut de la nouvelle fonction. La parenthèse qui suit peut contenir le cas échéant des paramètres de transfert.

Le code de la fonction est placé entre accolades.

Marche à suivre

1. Dans la fenêtre de navigation, double-cliquez sur la fonction correspondante pour ouvrir la fenêtre d'édition.
2. Placez le curseur à l'endroit où vous voulez écrire le code.
3. Saisissez le code.

Autre manipulation possible

Les possibilités ci-dessous permettent également d'ouvrir une fonction:

Ouvrez dans la fenêtre de navigation le menu contextuel de l'action voulue et sélectionnez "Ouvrir" ou utilisez le menu "Fichier\Ouvrir..." Vous pouvez aussi cliquer sur le bouton dans la barre d'outils "Standard" ou utiliser la combinaison de touches correspondante.

Remarque

L'espace mémoire disponible pour les variables locales (variables définies entre les accolades de la fonction) n'est que de 32 KO.

Voir aussi

Pour utiliser des fonctions standard et des fonctions de projet (Page 863)

Pour utiliser des fonctions internes (Page 862)

Fonctions d'édition avec la souris (Page 846)

Fonctions d'édition avec le clavier (Page 845)

Utiliser la fenêtre d'édition (Page 844)

2.12.4 Pour utiliser des fonctions internes

Introduction

Vous pouvez utiliser toutes les fonctions internes comme parties du code d"une fonction. Les fonctions internes se trouvent dans la fenêtre de navigation, dans le groupe "Fonctions internes".

Une fois que vous avez ajouté une fonction avec le dialogue de paramétrage, le commentaire de la fonction indique le type de la valeur retournée.

Marche à suivre

1. Placez le curseur à l'endroit où vous voulez ajouter la fonction interne
2. Dans la fenêtre de navigation, ouvrez le menu contextuel de la fonction interne à ajouter
3. Sélectionnez "Paramétrage". Le dialogue de paramétrage s'ouvre
   Le dialogue de paramétrage contient une ligne pour chaque paramètre. Vous devez saisir dans la colonne "Valeur" le paramètre actuel correspondant.
4. Dans la colonne "Valeur", saisissez la valeur actuelle des paramètres nécessaires.
   La saisie peut se faire au clavier ou en ouvrant le menu dans la colonne "Valeur" (simple clic puis clic sur le bouton affiché). Ce menu permet d'ouvrir le dialogue de sélection de variables, de vues ou d'objets graphiques.
5. Validez vos entrées avec "OK". La fonction paramétrée est ajoutée dans la fenêtre d'édition à la position du curseur.

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue de paramétrage de la fonction interne en double-cliquant sur la fonction à ajouter.

Remarque

Si vous fermez le dialogue de paramétrage (par "OK") sans avoir saisi les paramètres actuels, la fonction interne est ajoutée avec ses paramètres formels. Vous pouvez alors effectuer le paramétrage plus tard, dans la fenêtre d'édition.

Au lieu d'utiliser le dialogue de paramétrage, vous pouvez aussi écrire la fonction au clavier.

2.12.5 Pour utiliser des fonctions standard et des fonctions de projet

Introduction

Vous pouvez utiliser toutes les fonctions de projet et les fonctions standard comme parties d'une fonction à condition d'avoir inclus avant le fichier d'en-tête avec l'instruction #include "apdefap.h". Les fonctions de projet se trouvent dans la fenêtre de navigation, dans le groupe "Fonctions de projet". Les fonctions standard se trouvent dans la fenêtre de navigation, dans le groupe "Fonctions standard".

Les fonctions de projet sont déclarées dans le fichier d'en-tête Ap_pbib.h, les fonctions standard dans le fichier d'en-tête Ap_glob.h. Ces déclarations sont effectuées par le système. Le fichier d'en-tête Ap_glob.h est inséré dans le fichier d'en-tête Ap_pbib.h. Le fichier d'en-tête Ap_pbib.h est lui inséré dans le fichier d'en-tête Apdefap.h. Ainsi toutes les fonctions de projet et les fonctions standard sont-elles déclarées dans le fichier d'en-tête Apdefap.h.

Pour que les fonctions de projet et les fonctions standard insérées soient connues du compilateur, la ligne #include "apdefap.h" doit être placée devant la première ligne de code de la fonction.

Une fois que vous avez ajouté une fonction avec le dialogue de paramétrage, le commentaire de la fonction indique le type de la valeur retournée.

Marche à suivre

1. Placez le curseur à l'endroit où vous voulez ajouter la fonction de projet ou la fonction standard
2. Dans la fenêtre de navigation, ouvrez le menu contextuel de la fonction à ajouter
3. Sélectionnez "Paramétrage". Le dialogue de paramétrage s'ouvre
   Le dialogue de paramétrage contient une ligne pour chaque paramètre. Vous devez saisir dans la colonne "Valeur" le paramètre actuel correspondant.
4. Dans la colonne "Valeur", saisissez la valeur actuelle des paramètres nécessaires.
   La saisie peut se faire au clavier ou en ouvrant le menu dans la colonne "Valeur" (simple clic puis clic sur le bouton affiché). Ce menu permet d'ouvrir le dialogue de sélection de variables, de vues ou d'objets graphiques.
5. Validez votre saisie par "OK"

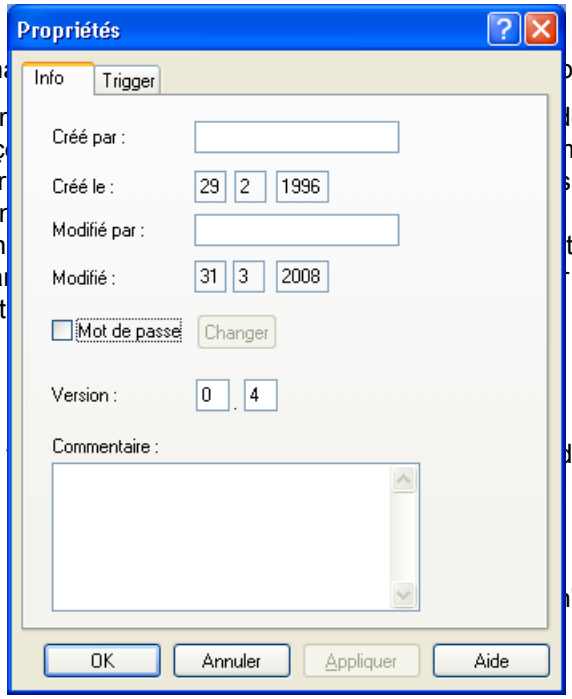
Remarque

Les fonctions qui ne nécessitent pas de paramètres sont ajoutées immédiatement au code de la fonction sans que la fenêtre de paramétrage s'ouvre.

Si vous fermez le dialogue de paramétrage (par "OK") sans avoir saisi les paramètres actuels, la fonction interne est ajoutée avec ses paramètres formels. Vous pouvez alors effectuer le paramétrage plus tard, dans la fenêtre d'édition.

2.12.6 Pour ajouter des informations supplémentaires à la fonction

Introduction



Ch plémentaires.

Lor le création s'inscrit automatiquement et de faç mentaires de la fonction. Le numéro de ver s pouvez attribuer individuellement les nur Lorsque vous modifiez et enregistrez une fon tiquement inscrite de façon non modifiable. Da un mot de passe contre toute modification et t

Condition

La doit être ouverte dans la fenêtre d'édition.

Marche à suivre

1. ". Le dialogue "Propriétés" s'ouvre.
2.
3. Validez votre saisie par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

Saisissez dans le menu "Edition" la commande "Info" ou utilisez la combinaison de touches correspondante.

Voir aussi

Pour protéger une fonction contre les modifications et l'affichage (Page 864)

Pour paramétrer plusieurs vues (Page 849)

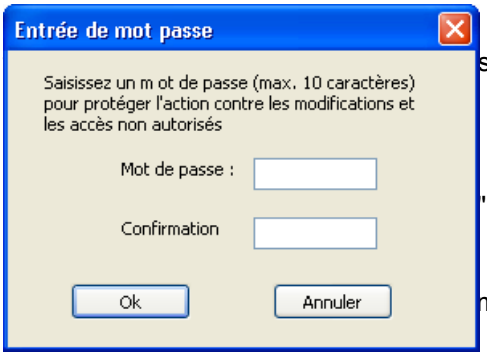
Utiliser les barres d'outils (Page 847)

2.12.7 Pour protéger une fonction contre les modifications et l'affichage

Introduction

Vous pouvez protéger les fonctions par un mot de passe contre les modifications et l'affichage. Le mot de passe est un élément des informations supplémentaires.

Condition

La 

s la fenêtre d'édition.

Marche à suivre

1. 'Edition". Le dialogue "Propriétés" s'ouvre.
2.
3. maintenant activé
4. Saisissez le mot de passe dans la ligne "Mot de passe"
5. Saisissez le mot de passe une seconde fois dans la ligne "Confirmation"
6. Validez votre saisie par "OK"
7. Validez votre saisie par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

Saisissez dans le menu "Edition" la commande "Info" ou utilisez la combinaison de touches correspondante.

Remarque

Une fonction protégée par mot de passe ne peut être ouverte dans la fenêtre d'édition qu'en entrant le mot de passe.

Pour annuler la protection par mot de passe, désactivez la case "Mot de passe".

Voir aussi

Pour paramétrer plusieurs vues (Page 849)

Utiliser les barres d'outils (Page 847)

2.12.8 Pour compiler et enregistrer une fonction

Introduction

Pour pouvoir utiliser une fonction, vous devez l'avoir compilée. Seule la fonction affichée dans la fenêtre d'édition est compilée.

Les erreurs signalées par le compilateur apparaissent au bas de la fenêtre d'édition. Chaque message est affiché sur forme d'une chaîne de caractères distincte. Cette chaîne contient le numéro de ligne du code source dans laquelle l'erreur apparaît, un code d'erreur et un message décrivant en clair l'erreur survenue.

Un double-clic sur le message d'erreur permet de marquer la ligne de code où est apparue l'erreur.

Il est très utile de vérifier le message d'erreur apparaissant le premier car les messages suivants peuvent être simplement la conséquence du premier. Dans ce cas, en éliminant la première erreur, les erreurs suivantes disparaissent également à la prochaine compilation.

Ces modifications de fonction doivent être enregistrées pour pouvoir être conservées.

Lorsque vous enregistrez une fonction pour laquelle une erreur de compilation est apparue ou une fonction qui n'est pas compilée du tout, elle est signalée par le symbole dans la fenêtre de navigation.

Lorsque vous enregistrez une fonction compilée sans erreur, elle est signalée par le symbole dans la fenêtre de navigation.

Condition

La fonction à compiler doit être ouverte dans la fenêtre d'édition.

Marche à suivre

1. Dans la barre d'outils, réglez la langue dans laquelle la fonction C doit être compilée.
2. Activez le bouton de la barre d'outils "Edition"
3. Contrôlez les messages du compilateur au bas de la fenêtre d'édition
4. Si le compilateur a émis des messages d'erreur, le code de la fonction doit être corrigé. Il faut dans ce cas recommencer à l'étape 1 de ce tableau.
5. Si le compilateur a émis des avertissements (warnings), le code de la fonction peut être corrigé. Si vous avez corrigé le code de la fonction, reprenez à l'étape 1 de ce tableau, sinon passez à l'étape 6
6. Activez le bouton de la barre d'outils "Standard"

Autre manipulation possible

Il existe d'autres possibilités de lancer la procédure de compilation :

Ouvrez le menu "Éditions" et choisissez la rubrique "Compilation" ou bien choisissez la rubrique "Compilation" du menu contextuel de la fenêtre d'édition, ou encore, utilisez la combinaison de touches correspondante.

Il existe d'autres possibilités de lancer l'enregistrement :

Ouvrez le menu "Fichier" et choisissez la rubrique "Enregistrer" ou utilisez la combinaison de touches correspondante.

Remarque

Si on utilise les noms de variable d'une fonction C en plusieurs cas, le compilateur n'émet pas de message d'erreur. C'est également le cas si un nom de variable est utilisé non seulement comme paramètre de transmission mais également comme définition locale de variable.

Par exemple, le script suivant défectueux ne provoque pas de message d'erreur dans le compilateur :

```
void nouvelle_fonction(DWORD dwMyVar)
{
DWORD dwMyVar = 0;
}
```

Alarme dans la fenêtre de sortie du compilateur :

Compiling ...

0 Error(s), 0 Warning(s)

Voir aussi

Comportement des actions au Runtime (Page 890)

2.12.9 Pour renommer une fonction

Introduction

Il est bon de renommer une fonction qui a été recréée.

Vous devez pour cela modifier le nom de la fonction dans la fenêtre d'édition. Le code ayant été modifié, la fonction doit être recompilée. Après la compilation, l'ancien nom de la fonction affiché dans la fenêtre de navigation est précédé du signe *.

Pour terminer, enregistrez la fonction modifiée en changeant si vous le souhaitez le chemin et le nom de fichier. Il est recommandé de supprimer ensuite l'ancienne fonction pour éviter une accumulation de fonctions qui ne sont plus actuelles.

Remarque

Veillez à n'utiliser que des caractères autorisés pour les noms de fonction : uniquement des lettres, mais sans les caractères spéciaux nationaux, des chiffres et le trait de soulignement.

Marche à suivre

1. Changez le nom de la fonction dans la fenêtre d'édition
2. Activez le bouton de la barre d'outils "Edition". La fonction est compilée.
3. Cliquez sur le bouton de la barre d'outils "Standard" pour enregistrer la fonction.
4. Définissez le cas échéant un autre chemin et un autre nom de fichier.
5. Confirmez vos saisies avec le bouton "Enregistrer"

Autre manipulation possible

Il existe d'autres possibilités de lancer la procédure de compilation :

Ouvrez le menu "Éditions" et choisissez la rubrique "Compilation" ou bien choisissez la rubrique "Compilation" du menu contextuel de la fenêtre d'édition, ou encore, utilisez la combinaison de touches correspondante.

Il existe d'autres possibilités de lancer l'enregistrement :

Ouvrez le menu "Fichier" et choisissez la rubrique "Enregistrer" ou utilisez la combinaison de touches correspondante.

2.12.10 Pour utiliser des fonctions empruntées à d'autres sources

Introduction

Les fonctions de projet empruntées à d'autres projets WinCC et les fonctions standard d'autres systèmes WinCC peuvent aussi être utilisées dans le projet actuel. Vous devez pour cela les intégrer au projet actuel.

Mis à part le lieu de stockage des fonctions dans le système de fichiers, la procédure est la même pour les fonctions de projet et les fonctions standard.

Marche à suivre

1. Copiez les fonctions. Les fonctions de projet se copient, dans le chemin de projet, dans le répertoire "library". Les fonctions standard se copient, dans le chemin WinCC, dans le répertoire "aplib\...". Le contenu de la fenêtre de navigation est automatiquement rafraîchi.
2. Activez le bouton de la barre d'outils "Standard". De par la génération de l'en-tête, les fonctions copiées dans le projet sont connues du projet actuel, dans lequel vous pouvez alors les utiliser pour la configuration.

Autre manipulation possible

Vous pouvez aussi générer l'en-tête de la manière suivante:

Ouvrez le menu "Options" et sélectionnez la commande "Générer en-tête" ou utilisez la combinaison de touches correspondante.

Remarque

Lors d'une réinstallation ou d'une mise à jour de WinCC, les fonctions standard personnalisées ou modifiées sont supprimées / remplacées par les fonctions d'origine.

Si WinCC se trouve en mode Runtime, le système Runtime n'est pas affecté par la nouvelle génération de l'en-tête.

2.13 Création et édition d'actions

2.13.1 Création et édition d'actions

Introduction

On distingue les actions globales et les actions locales. Dans un projet Client-Serveur les actions globales sont exécutées sur tous les ordinateurs, les actions locales uniquement sur l'ordinateur désigné.

Une action globale peut, par ex., être utilisée pour effectuer un calcul sur tous les ordinateurs.

Une action locale peut, par ex., être utilisée pour imprimer un journal sur un serveur.

La création et l'édition des deux types d'actions sont absolument identiques.

Différences entre actions et fonctions

- A la différence des fonctions, les actions peuvent posséder un déclenchement. Cela signifie qu'une fonction ne peut jamais être exécutée seule au Runtime.
- Les actions peuvent être exportées et importées.
- Une autorisation peut être attribuée à une action. Les commandes disponibles dans la fenêtre de dialogue Global Script - Runtime permettent de délivrer une autorisation.
- Une action ne comporte aucun paramètre.

Recherche des actions

La fenêtre de navigation de Global Script donne accès aux actions existantes et à la création de nouvelles actions.

Les actions sont créées dans le système de fichiers de la manière suivante :

Édition et compilation d'actions

L'édition et la compilation d'une action se fait dans une fenêtre individuelle. Une fois la compilation exécutée, la fenêtre d'édition affiche les messages du compilateur. Il peut s'agir de messages d'avertissement ou d'erreur. Dans tous les cas, le nombre des avertissements et des erreurs est indiqué.

Représentation des actions

Si vous enregistrer une action de syntaxe incorrecte, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Si vous enregistrer une action de syntaxe correcte sans déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Si vous enregistrer une action de syntaxe correcte avec déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Pour la création des actions, observez les règles suivantes :

Le composant CrossReference de WinCC permet de créer des listes de références croisées. Pour que les variables et les images soient reconnues lors de la construction des listes pendant l'utilisation des appels de fonction à l'intérieur des actions, il faut incorporer les directives décrites ci-dessous.

Modification du nom des actions

Dans la fenêtre de navigation, les actions sont toujours désignées par le nom de leur fichier. Renommer une action revient à renommer le fichier qui contient le code exécutable de l'action.

Comportement du système lorsque des actions sont modifiées, effacées ou enregistrées en Runtime.

Si une action locale est enregistrée en Runtime, toutes les actions locales et globales de l'ordinateur sur lequel cette action s'exécute sont réinitialisées sur cet ordinateur.

Si une action globale est enregistrée en Runtime, toutes les actions locales et globales du projet et donc de tous les ordinateurs concernés sont réinitialisées.

La réinitialisation réinitialise également par ex. les variables et les horaires utilisés dans les actions comme déclenchement, et l'action se déclenche alors.

Les variables statistiques apparaissant dans les actions réinitialisées sont elles aussi réinitialisées.

Causes possibles de non-exécution d'une action en Runtime.

Une action peut ne pas s'exécuter en Runtime pour l'une des raisons suivantes :

- L'action n'a pas de déclenchement.
- L'action n'a pas été compilée.
- Le paramètre Global Script Runtime n'est pas activé dans la liste de démarrage du projet.

Remarque

Avant de créer une action, il faut vérifier si les fonctionnalités y afférentes peuvent être aussi réalisées en usine.

Voir aussi

Pour protéger une action contre les modifications et l'affichage (Page 876)

Pour utiliser des actions étrangères au projet (Page 888)

Pour renommer une action (Page 887)

Pour importer une action (Page 887)

Pour exporter une action (Page 886)

Pour attribuer une autorisation (Page 885)

Déclenchement (Page 879)

Pour compiler et enregistrer une action (Page 877)

Pour ajouter des informations supplémentaires à l'action (Page 876)

Pour éditer une action (Page 875)

Pour créer une nouvelle action (Page 874)

Règle d'écriture du code de WinCC (Page 873)

Pour ajouter Global Script Runtime à la liste de démarrage du projet (Page 837)

Caractéristiques d'actions globales (Page 836)

Caractéristiques d'actions locales (Page 835)

2.13.2 Règle d'écriture du code de WinCC

Règle d'écriture du code pour l'utilisation du composant CrossReference

Le composant CrossReference de WinCC permet de créer des listes de références croisées. Pour que, lors de la construction de la liste de références croisées, les variables et les vues soient reconnues en présence d'appels de fonctions dans les actions, il convient de respecter la règle décrite ci-dessous.

Le code de l'action commence par deux sections. La première sert à déclarer toutes les variables utilisées, la deuxièmes tous les noms de vue.

Vous ne devez inscrire aucune autre instruction dans les sections.

Les deux sections se présentent sous forme de commentaires lors de la création d'une nouvelle action:

```
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END
```

Les sections sont ensuite complétées, par ex. de la manière suivante:

```
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
#define ApcTagName1 "TagName1"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
#define ApcPicName1 "PicName1"
#define ApcPicName2 "PicName2"
```

```
#define ApcPicName3 "PicName3"
// WINCC:PICNAME_SECTION_END
```

Les fonctions doivent être appelées pour la lecture et l'écriture de variables et utilisées avec le nom défini:

```
GetTagDWord(ApcTagName1);
OpenPicture(ApcPicName1);
SetPictureName(ApcPicName2,"PictureWindow1",ApcPicName3);
```

2.13.3 Pour créer une nouvelle action

Introduction

Dans un projet Client-Serveur les actions globales sont exécutées sur tous les ordinateurs, les actions locales uniquement sur l'ordinateur désigné.

La procédure est la même pour les actions globales et les actions locales. En choisissant le lieu de stockage dans la fenêtre de navigation vous définissez le type de l'action (globale ou locale).

Global Script propose un nom de fichier par défaut pour la nouvelle action.

Une action nouvellement créée contient déjà l'instruction #include "apdefap.h". Toutes les fonctions sont ainsi connues de l'action. La troisième ligne contient le nom de l'action. Les trois première lignes ne peuvent être ni supprimées ni modifiées. Cela signifie que cela suffit pour que toutes les fonctions puissent être utilisées dans chaque action. En outre la valeur retournée par chaque action est toujours du type int et a déjà la valeur par défaut "0".

Vous pouvez modifier la valeur retournée d'une action et l'utiliser pour le diagnostic avec GSC-Runtime.

Le code de l'action commence par un canevas de code présenté comme un commentaire. Les variables et noms de vue sont reconnus par Cross Reference si ce canevas est rempli en respectant la règle d'écriture du code.

Marche à suivre

1. Dans la fenêtre de navigation, ouvrez le menu contextuel du type d'action souhaité
2. Sélectionnez "Nouveau"

Autre manipulation possible

Vous pouvez également créer une nouvelle action globale avec l'icône de la barre d'outils, le menu "Fichier" ou la combinaison de touches correspondante.

Voir aussi

GSC-Runtime (Page 891)

Règle d'écriture du code de WinCC (Page 873)

2.13.4 Pour éditer une action

Introduction

Une action s'édite dans la fenêtre d'édition tout comme une fonction. Seules les 3 premières lignes ne sont pas éditables.

L'action doit retourner une valeur. La valeur retournée est du type int et doit être "0" par défaut. Vous pouvez modifier la valeur retournée et l'utiliser pour le diagnostic avec GSC-Runtime. Le type de la valeur retournée n'est pas modifiable.

Pour que l'action soit exécutée au Runtime, un déclenchement doit y être lié.

Marche à suivre

1. Dans la fenêtre de navigation, double-cliquez sur l'action correspondante pour l'ouvrir pour l'édition.
2. Editez le code de l'action.

Autre manipulation possible

Les possibilités ci-dessous permettent également d'ouvrir une action:

Ouvrez dans la fenêtre de navigation le menu contextuel de l'action voulue et sélectionnez "Ouvrir" ou utilisez le menu "Fichier\Ouvrir..." Vous pouvez aussi cliquer sur le bouton dans la barre d'outils "Standard" ou utiliser la combinaison de touches correspondante.

Remarque

L'espace mémoire disponible pour les variables locales (variables définies entre les accolades de l'action) n'est que de 32 KO.

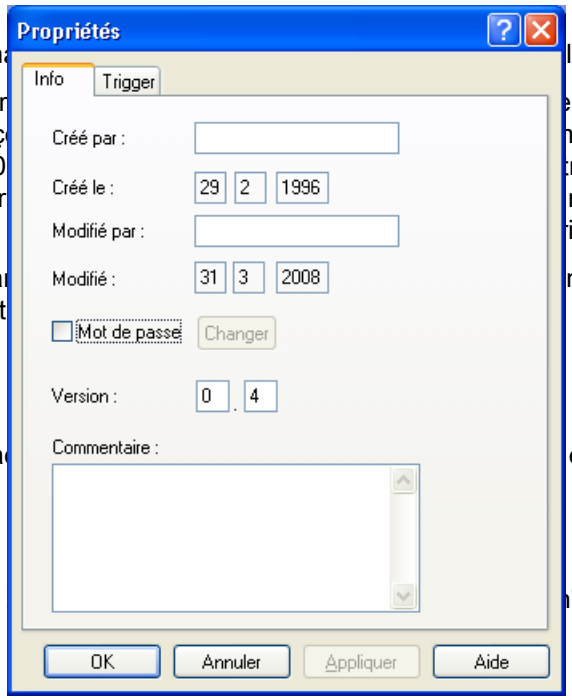
Voir aussi

GSC-Runtime (Page 891)

Pour écrire le code d'une fonction (Page 861)

2.13.5 Pour ajouter des informations supplémentaires à l'action

Introduction



Ch lémentaires.

Lor e création s'inscrit automatiquement et de faç nentaires de l'action. Le numéro de version 1.0 tribuer individuellement les numéros de ver modifiez et enregistrez une action, la date de ite de façon non modifiable.

Da n mot de passe contre toute modification et t

Condition

L'a être ouverte dans la fenêtre d'édition.

Marche à suivre

1. ". Le dialogue "Propriétés" s'ouvre.
2.
3. Validez votre saisie par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

Saisissez dans le menu "Edition" la commande "Info" ou utilisez la combinaison de touches correspondante.

Voir aussi

Pour protéger une action contre les modifications et l'affichage (Page 876)

Pour paramétrer plusieurs vues (Page 849)

Utiliser les barres d'outils (Page 847)

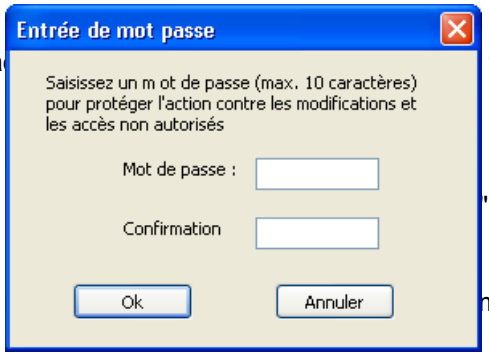
2.13.6 Pour protéger une action contre les modifications et l'affichage

Introduction

Vous pouvez protéger les actions par un mot de passe contre les modifications et l'affichage. Le mot de passe est un élément des informations supplémentaires de l'action.

Condition

L'a fenêtre d'édition.



Marche à suivre

1. 'Edition". Le dialogue "Propriétés" s'ouvre.
2.
3. maintenant activé
4. Saisissez le mot de passe dans la ligne "Mot de passe"
5. Saisissez le mot de passe une seconde fois dans la ligne "Confirmation"
6. Validez votre saisie par "OK"
7. Validez votre saisie par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

Saisissez dans le menu "Edition" la commande "Info" ou utilisez la combinaison de touches correspondante.

Remarque

Une action protégée par mot de passe ne peut être ouverte dans la fenêtre d'édition qu'en entrant le mot de passe.

Pour annuler la protection par mot de passe, désactivez la case "Mot de passe".

Voir aussi

Pour paramétrer plusieurs vues (Page 849)

Utiliser les barres d'outils (Page 847)

2.13.7 Pour compiler et enregistrer une action

Introduction

Pour être utilisée, une action doit d'abord être compilée. Seule l'action présente dans la fenêtre d'édition active est compilée.

Les erreurs signalées par le compilateur apparaissent au bas de la fenêtre d'édition. Chaque message est affiché sur forme d'une chaîne de caractères distincte. Cette chaîne contient le numéro de ligne du code source dans laquelle l'erreur apparaît, un code d'erreur et un message décrivant en clair l'erreur survenue.

Un double clic sur une ligne de ce type marque la ligne de code dans laquelle l'erreur s'est produite.

Il est très utile de vérifier le message d'erreur apparaissant le premier car les messages suivants peuvent être simplement la conséquence du premier. Dans ce cas, en éliminant la première erreur, les erreurs suivantes disparaissent également à la prochaine compilation.

Condition

L'action à compiler doit être ouverte dans la fenêtre d'édition.

Marche à suivre

1. Dans la barre d'outils, réglez la langue dans laquelle l'action en C doit être compilée.
2. Activez le bouton de la barre d'outils "Edition"
3. Contrôlez les messages du compilateur au bas de la fenêtre d'édition
4. Si le compilateur trouve des erreurs, il est nécessaire de corriger le code source de l'action. Il faut dans ce cas recommencer à l'étape 1 de ce tableau.
5. Si le compilateur envoie des avertissements, le code source de l'action peut éventuellement être corrigé. Dans ce cas, rendez vous à l'étape 1 de ce tableau, dans le cas contraire poursuivez à l'étape 6.
6. Activez le bouton de la barre d'outils "Standard"

Autre manipulation possible

Il existe d'autres possibilités de lancer la procédure de compilation :

Ouvrez le menu "Éditions" et choisissez la rubrique "Compilation" ou bien choisissez la rubrique "Compilation" du menu contextuel de la fenêtre d'édition, ou encore, utilisez la combinaison de touches correspondante.

Il existe d'autres possibilités de lancer l'enregistrement :

Ouvrez le menu "Fichier" et choisissez la rubrique "Enregistrer" ou utilisez la combinaison de touches correspondante.

Représentation des actions

Si vous enregistrer une action de syntaxe incorrecte, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Si vous enregistrer une action de syntaxe correcte sans déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Si vous enregistrer une action de syntaxe correcte avec déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Voir aussi

Comportement des actions au Runtime (Page 890)

2.13.8 Déclenchement

2.13.8.1 Déclenchement

Définition et utilisation des déclenchements

Des déclenchements sont utilisés pour exécuter les actions en Runtime. A cet effet, un déclenchement est mis en corrélation avec une action et constitue ainsi l'événement qui déclenchera l'appel de cette action. Les actions sans déclenchement ne sont pas exécutées.

Types de déclenchement

Les déclenchements suivants sont à disposition :

Déclenchements acycliques

Ils sont constitués d'une date et d'une heure. L'action liée à un déclenchement de ce type sera exécutée, une fois, à la date et à l'heure paramétrées.

Déclenchement cyclique

Ils sont constitués d'un intervalle de temps et de l'instant du début d'exécution. Les déclenchements cycliques suivants sont disponibles:

- Cycle standard. Le début du premier intervalle de temps coïncide avec le démarrage du Runtime. La longueur de l'intervalle de temps est déterminée par le cycle.
- Heure. Le début de l'intervalle de temps est indiqué en minutes et secondes. La longueur de l'intervalle est d'une heure.
- Jour. Le début de l'intervalle de temps est paramétré par indication de l'heure (heure, minute et seconde). La longueur de l'intervalle est d'un jour.
- Semaine. Le début de l'intervalle de temps est paramétré par indication du jour de la semaine (lundi, mardi...) et de l'heure. La longueur de l'intervalle est d'une semaine.
- Mois. Le début de l'intervalle de temps est indiqué par le jour et l'heure. La longueur de l'intervalle est d'un mois.
- An. Le début de l'intervalle de temps est indiqué par le jour, le mois et l'heure. La longueur de l'intervalle est d'un an.

Déclenchement par variable

Ils sont constitués d'une ou de plusieurs variables. L'action, liée à un tel déclenchement est exécutée à chaque fois qu'il se produit une modification de la valeur de l'une de ces variables. La manière dont les valeurs de variable sont scrutées est paramétrable individuellement. Vous pouvez choisir entre une scrutation selon un cycle standard paramétrable ou une réaction intervenant dès que le système constate une variation de valeur d'une variable.
Il peut arriver, en fonction du type de scrutation choisi, que la valeur de la variable varie sans que le système s'en aperçoive. Dans ce cas l'action n'est pas exécutée.

Effet des déclenchements sur une action

Si l'action n'est liée qu'à un seul déclenchement, elle est exécutée dès que l'événement se produit.

Une action peut cependant être liée à plusieurs déclenchements, comme par ex. à un déclenchement cyclique et un déclenchement par variable. L'action est alors exécutée dès que l'un des événements se produit. Si deux événements se produisent simultanément, l'action est exécutée deux fois de suite. Si deux variables sont modifiées simultanément, l'action n'est exécutée qu'une fois.

La routine d'interruption d'une action devrait être terminée avant d'être de nouveau lancée, dans le cas contraire, on pourrait aboutir à un dépassement de la file d'attente.

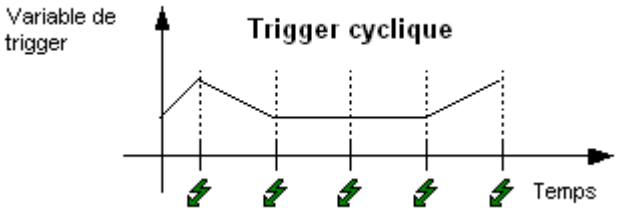
Astuce : Si l'action ne doit pas être exécutée à chaque événement, vous pouvez formuler dans l'action une condition commandant la suite de l'exécution de l'action en fonction du résultat du test. Si l'exécution de l'action ne doit pas être poursuivie, vous pouvez l'abandonner avec return <valeur>.

Règles à suivre pour la sélection du déclenchement

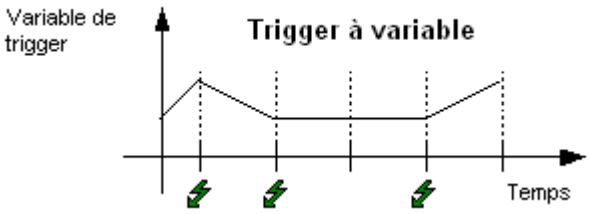
Pour des raisons internes au système, il n'est pas possible de garantir qu'une action à déclenchement cyclique soit exécutée exactement aux instants spécifiés. Si cette garantie est nécessaire, cette tâche (tests de vérifications, par exemple) doit être effectuée en usine.

Si les variables doivent être éditées, utilisez le déclenchement par variable de préférence au déclenchement cyclique.

- Si on utilise le déclenchement cyclique, l'action est toujours exécutée au moment exact de l'événement, ex. : toutes les 20 secondes. (= l'action sera exécutée)



- Le déclenchement de l'exécution d'une action par une variable ne se produit que si la variable de déclenchement a été modifiée. Cela permet de réduire la charge du système ((= l'action sera exécutée).



Pour les variables concernées par le déclenchement, la valeur est déjà connue au moment du démarrage de l'action. Un appel GetTag() transfère directement la valeur. La prise en compte est alors beaucoup plus rapide que lorsque la commande GetTag() est employée avec des variables qui ne sont pas des variables de déclenchement.

Représentation des actions

Si vous enregistrer une action de syntaxe incorrecte, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Si vous enregistrer une action de syntaxe correcte sans déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Si vous enregistrer une action de syntaxe correcte avec déclenchement, elle s'affiche dans la fenêtre de navigation de Global Script avec l'icône ci-contre.

Voir aussi

Pour supprimer un déclenchement (Page 884)

Pour modifier un déclenchement (Page 884)

Pour ajouter un nouveau déclenchement du type "Variable" (Page 883)

Pour ajouter un nouveau déclenchement du type "Temporisateur" (Page 882)

2.13.8.2 Pour ajouter un nouveau déclenchement du type "Temporisateur"

Introduction

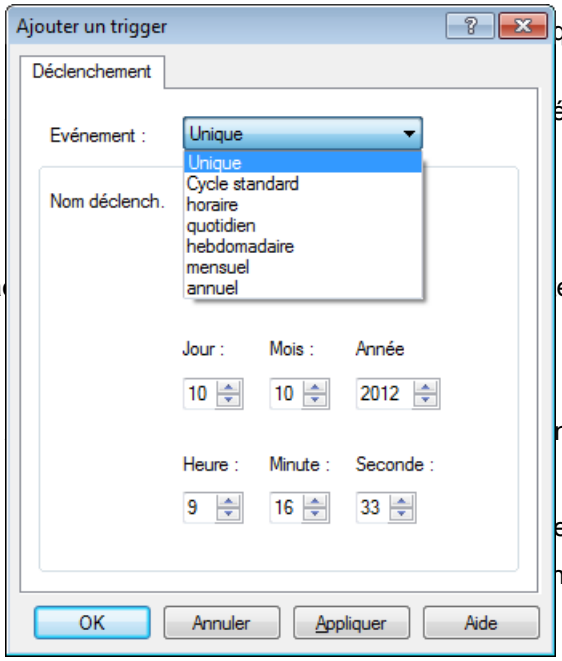
Les déclenchements (trigger) sont nécessaires pour exécuter des actions en Runtime. On lie un déclenchement à une action pour créer l'événement déclencheur pour l'appel de l'action. Les actions sans déclenchement ne sont pas exécutées.

Les déclenchements du type "Temporisateurs" sont acycliques ou cycliques.

Les déclenchements acycliques sont constitués d'une date et d'une heure. L'action liée à un déclenchement de ce type sera exécutée, une fois, à la date et à l'heure paramétrées.

Les déclenchements cycliques sont constitués d'un intervalle de temps et de l'instant du début d'exécution. Les déclenchements cycliques suivants sont disponibles:

- Cycle standard. Le début du premier intervalle de temps coïncide avec le lancement du système Runtime. La longueur de l'intervalle de temps est déterminée par le cycle.
- Heure. Le début de l'intervalle de temps est indiqué en minutes et secondes. La longueur de l'intervalle est d'une heure.
- Jour. Le début de l'intervalle de temps est paramétré par indication de l'heure (heure, minute et seconde). La longueur de l'intervalle est d'un jour.
- Semaine. Le début de l'intervalle de temps est paramétré par indication du jour de la semaine (lundi, mardi...) et de l'heure. La longueur de l'intervalle est d'une semaine.
- qué par le jour et l'heure. La longueur de
- é par le jour, le mois et l'heure. La longueur



Condition

L'a e dans la fenêtre d'édition.

Marche à suivre

1. n". Le dialogue "Propriétés" s'ouvre.
2.
3. ent et cliquez sur le bouton "Ajouter"
4. nement acyclique ou choisissez un cycle
5. Complétez le paramétrage du dialogue
6. Validez les paramétrages par "OK"
7. Fermez le dialogue "Propriétés" par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

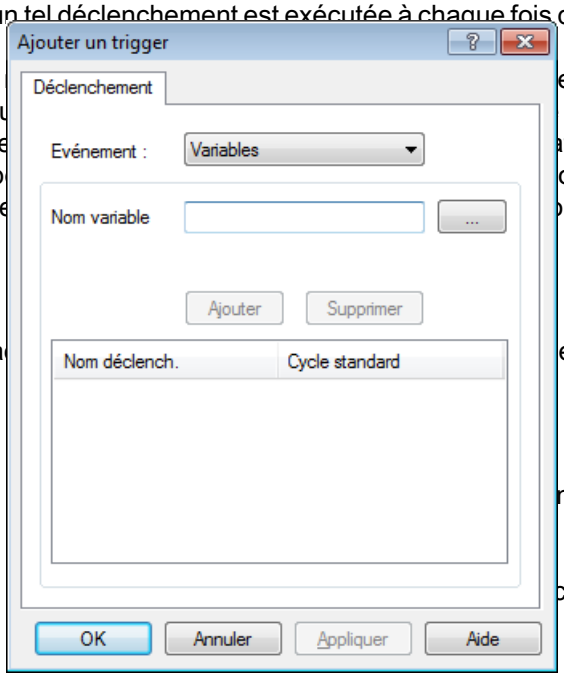
Sélectionnez dans le menu "Edition" la commande "Info" ou sélectionnez la commande "Info/ Trigger" dans le menu contextuel de la fenêtre d'édition ou utilisez la combinaison de touches correspondante.

2.13.8.3 Pour ajouter un nouveau déclenchement du type "Variable"

Introduction

Les déclenchements (trigger) sont nécessaires pour exécuter des actions en Runtime. On lie un déclenchement à une action pour créer l'événement déclencheur pour l'appel de l'action. Les actions sans déclenchement ne sont pas exécutées.

Le déclenchement par variable(s) est constitué d'une ou de plusieurs variables. L'action liée à un tel déclenchement est exécutée à chaque fois qu'il se produit une modification de la valeur de



La es est paramétrable individuellement. Vous pou standard paramétrable ou une réaction inte tion de valeur d'une variable.
Il p oisi, que la valeur de la variable varie sans que n n'est pas exécutée.

Condition

L'a e dans la fenêtre d'édition.

Marche à suivre

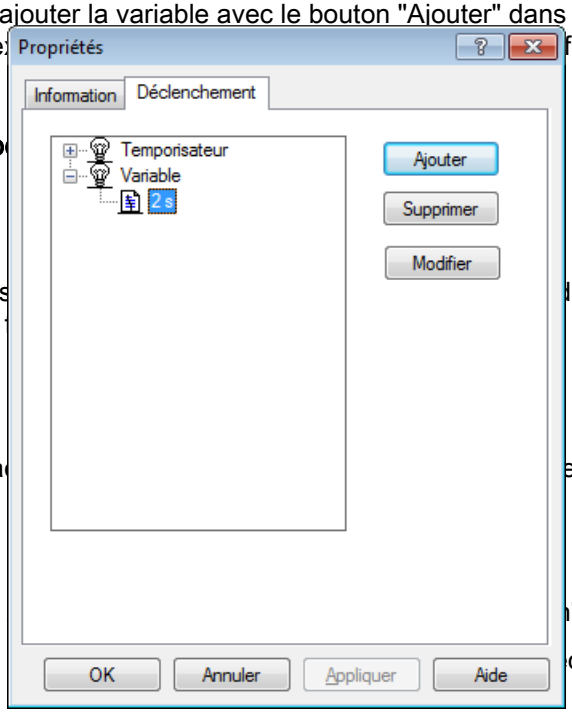
1. n". Le dialogue "Propriétés" s'ouvre.
2.
3. cliquez sur le bouton "Ajouter". Le dialogue
4. Avec le bouton ..., ouvrez le dialogue de sélection de variable, sélectionnez une variable et validez la sélection par "OK"
5. Dans le dialogue "Ajouter un trigger", ouvrez le menu contextuel dans "Cycle standard" et sélectionnez le cycle souhaité. "Sur événement" signifie que la surveillance est permanente.
6. Répétez les étapes 4 et 5 pour ajouter une autre variable
7. Validez les paramétrages par "OK"
8. Fermez le dialogue "Propriétés" par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

Sélectionnez dans le menu "Edition" la commande "Info" ou sélectionnez la commande "Info/ Trigger" dans le menu contextuel de la fenêtre d'édition ou utilisez la combinaison de touches correspondante.

Dans le dialogue "Ajouter un trigger" vous pouvez aussi saisir directement un nom de variable et ajouter la variable avec le bouton "Ajouter" dans la colonne "Nom de déclenchement". L'e fiée.



2.13.8.4 Pour mo

Introduction

Les ifiés à tout moment. Ceci peut également se

Condition

L'a e d'édition.

Marche à suivre

1. ". Le dialogue "Propriétés" s'ouvre.
2. clenchement à modifier
3. Cliquez sur le bouton "Modifier" pour ouvrir le dialogue "Modifier trigger
4. Effectuez les changements
5. Validez par "OK"
6. Fermez le dialogue "Propriétés" par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

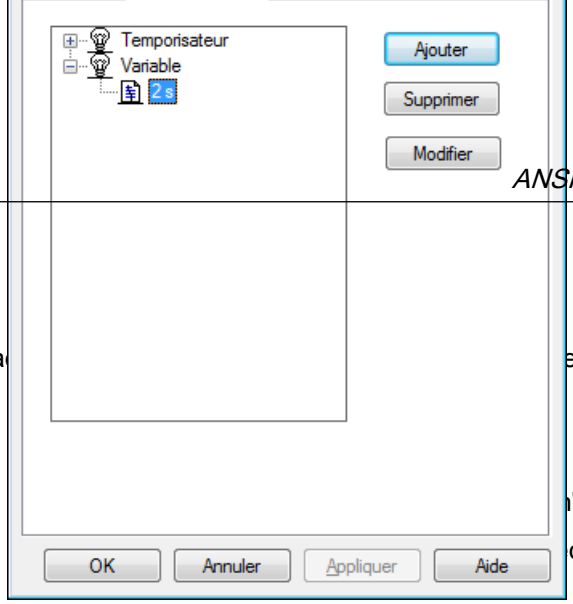
Sélectionnez dans le menu "Edition" la commande "Info" ou sélectionnez la commande "Info/ Trigger" dans le menu contextuel de la fenêtre d'édition ou utilisez la combinaison de touches correspondante.

2.13.8.5 Pour supprimer un déclenchement

Introduction

Les déclenchements paramétrés peuvent être supprimés à tout moment. Ceci peut également se faire pendant le Runtime.

Lorsqu'un déclenchement est supprimé au Runtime, la suppression n'est effective qu'après l'enregistrement de l'action.



Condition

L'a e d'édition.

Marche à suivre

1. ". Le dialogue "Propriétés" s'ouvre.
2. clenchement à supprimer
3. Supprimez le déclenchement marqué avec le bouton "Supprimer"
4. Fermez le dialogue "Propriétés" par "OK"

Autre manipulation possible

Vous pouvez aussi ouvrir le dialogue "Propriétés" de la manière suivante:

Sélectionnez dans le menu "Edition" la commande "Info" ou sélectionnez la commande "Info/ Trigger" dans le menu contextuel de la fenêtre d'édition ou utilisez la combinaison de touches correspondante.

2.13.9 Pour attribuer une autorisation

Introduction

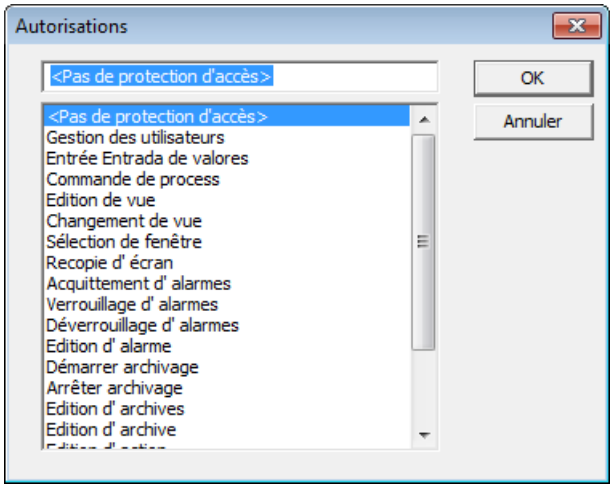
L'outil de diagnostic Global Script - Runtime permet de piloter l'exécution d'actions au Runtime. Vous pouvez attribuer une autorisation à une action. Cette autorisation n'a des effets que sur la fenêtre Global Script - Runtime.

Condition

L'action concernée doit être ouverte dans la fenêtre d'édition.

Marche à suivre

1. Choisissez dans le menu "Editer" la commande "Autorisation de conduite".
   Le dialogue "Autorisations" s'ouvre.
2. Sélectionnez une autorisation.



3. Confirmez votre sélection par "OK".

Voir aussi

GSC-Runtime (Page 891)

2.13.10 Pour ex

Introduction

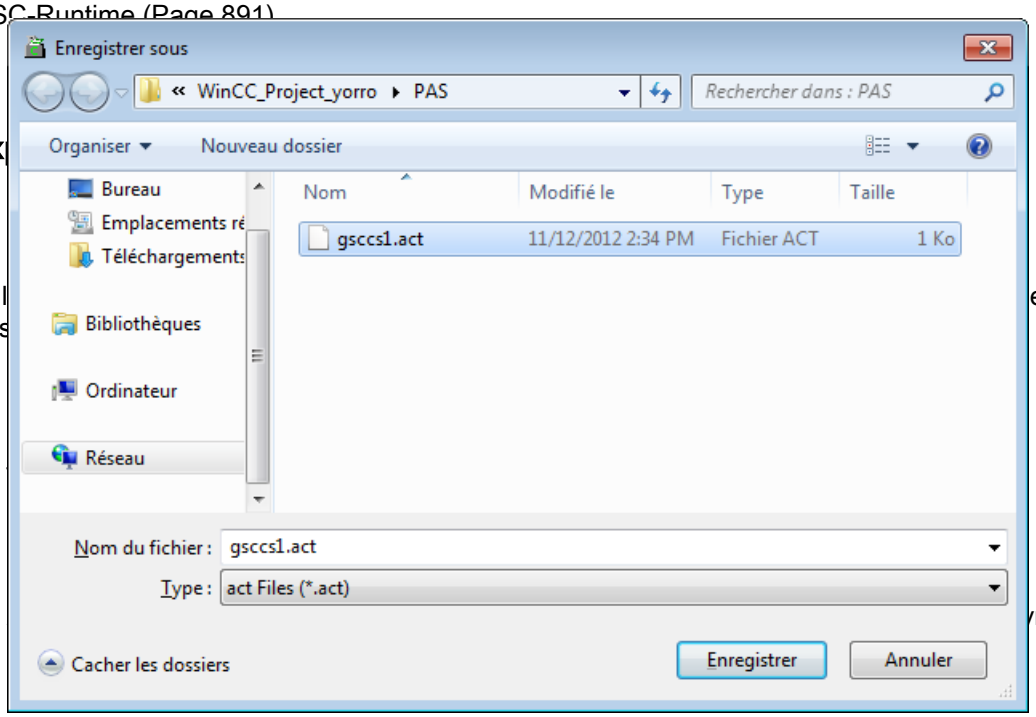
Util et.
Les

Condition

La

Marche à suivre

1. re.
2.

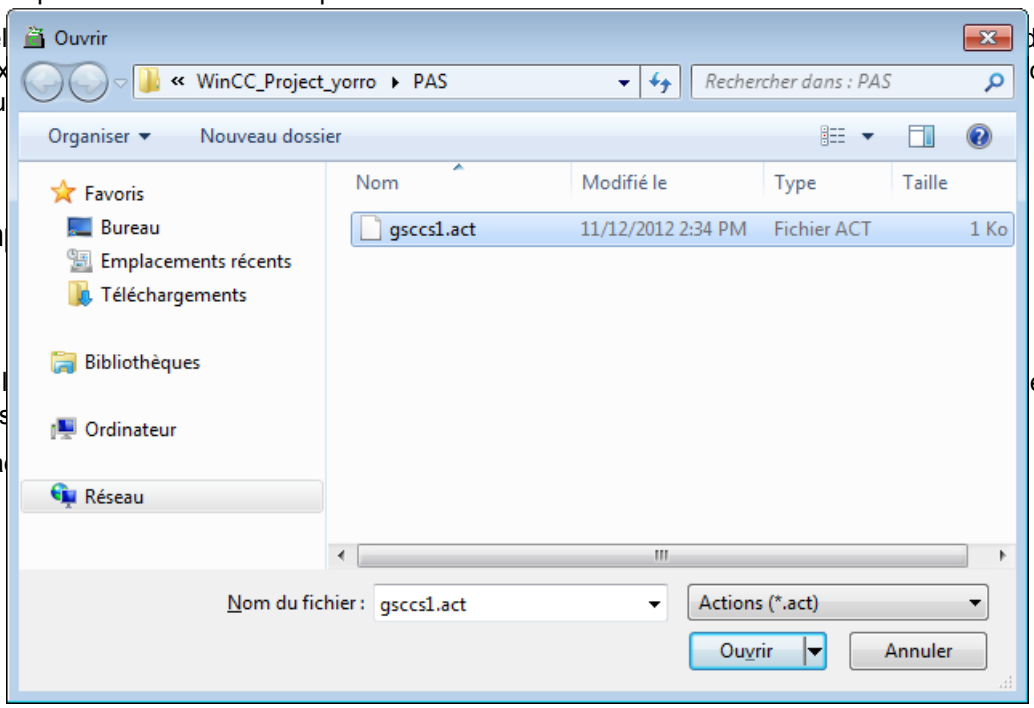


3. Exportez l'action avec le bouton "Enregistrer"

Autre manipulation possible

Vous pouvez aussi lancer l'exportation de la manière suivante:

Sél de
"Ex de
tou



2.13.11 Pour im

Introduction

Util et.
Les
L'a

Marche à suivre

1.
2.
3. Importez l'action avec le bouton "Ouvrir"

Autre manipulation possible

Vous pouvez aussi lancer l'importation de la manière suivante:

Sélectionnez dans le menu "Edition" la commande "Importer" ou sélectionnez la commande "Importer action" dans le menu contextuel de la fenêtre d'édition ou utilisez la combinaison de touches correspondante.

2.13.12 Pour renommer une action

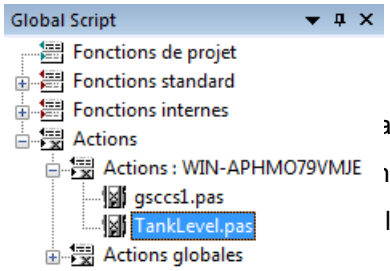
Introduction

Vous pouvez renommer une action. L'action reçoit ainsi un autre nom de fichier.

Condition

L'action que vous voulez renommer ne doit pas être ouverte dans la fenêtre d'édition.

Marche à suivre



1. action que vous voulez renommer
2. nommer"
3. l'extension .pas
4. Validez le nouveau nom avec la touche <Return>

Autre manipulation possible

Vous pouvez renommer une action par deux clics simples sur le nom de l'action.

Remarque

Le nom de l'action reste inchangé si vous ne saisissez pas l'extension ".pas".

2.13.13 Pour utiliser des actions étrangères au projet

Introduction

Vous avez deux possibilités pour utiliser dans le projet actuel des actions empruntées à d'autres sources:

- Importation d'actions exportées
- Copie du fichier contenant l'action souhaitée dans le chemin de votre projet. Le chemin des actions locales est dans le chemin du projet, dans le répertoire <Nom d'ordinateur>\Pas. Le chemin des actions globales est dans le chemin du projet, dans le répertoire Pas.

L'affichage doit être rafraîchi pour afficher dans la fenêtre de navigation les actions copiées. Ceci peut se faire en arrêtant et en redémarrant Global Script.

Lorsque le mode Runtime est actif, les actions ainsi reprises ne sont exécutées qu'après avoir été ouvertes dans l'éditeur Global Script et avoir été enregistrées.

Remarque

Les actions peuvent contenir des appels de fonctions de projet et de fonctions standard. Celles-ci peuvent à leur tour contenir des appels de fonctions de projet et de fonctions standard et ainsi de suite. Il faut donc s'assurer en reprenant des actions étrangères au projet que toutes les fonctions nécessaires sont bien disponibles dans le projet actuel.

Une attention toute particulière s'impose lorsqu'il s'agit d'actions venant d'un autre ordinateur. Les fonctions standard pouvant être modifiées par l'utilisateur en fonction de ses besoins, il peut arriver que les fonctions standard appelées dans l'action possèdent sur l'ordinateur source une autre fonctionnalité que les fonctions standard de même nom sur l'ordinateur cible.

Voir aussi

Pour importer une action (Page 887)
Pour exporter une action (Page 886)

2.14 Comportement des actions au Runtime

2.14.1 Comportement des actions au Runtime

Analyse du comportement au Runtime

WinCC propose trois outils pour l'analyse du comportement des actions au Runtime. Ce sont: la fenêtre d'application GSC-Runtime, Diagnostic GSC et l'application apdiag.exe.

Pour utiliser la fenêtre d'application GSC-Runtime et Diagnostic GSC, vous devez les placer dans une vue de process. Celle-ci peut être une vue de process configurée uniquement pour le diagnostic. Elle est appelée au Runtime.

Ces fenêtres d'application permettent les stratégies suivantes:

- GSC-Runtime renseigne sur le comportement dynamique de toutes les actions (Global Script), permet de lancer individuellement et de déclarer le début et la fin de chaque action; il permet aussi de basculer dans l'éditeur Global Script le runtime étant actif
- La fonction Diagnostic GSC affiche, dans l'ordre chronologique de leur appel, les instructions printf contenues dans les actions. Il affiche également les instructions printf codées dans les fonctions appelées dans des actions. L'utilisation ciblée d'instructions printf, par ex. pour afficher des valeurs de variable, permet de suivre l'exécution d'actions et des fonctions appelées par celles-ci. Les situations génératrices d'erreurs qui entraînent l'appel de la fonction OnErrorExecute sont affichées dans la fenêtre de diagnostic.

Remarque

Lors des dynamisations de script C avec accès aux objets de la vue, prenez en considération le fait qu'on ne quitte pas automatiquement un script figurant dans un traitement en désélectionnant la vue.

Cela peut d'ailleurs avoir pour conséquence l'échec de l'accès aux objets adressés dans le script. Par exemple, si les propriétés du type "Texte" sont lues et que les valeurs fournies sont modifiées ou traitées dans les opérations de chaîne suivantes.

Voir aussi

Diagnostic GSC (Page 896)

GSC-Runtime (Page 891)

2.14.2 GSC-Runtime

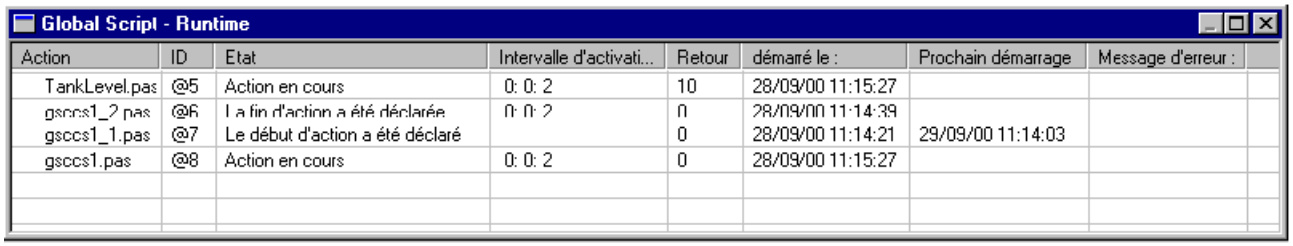
2.14.2.1 GSC-Runtime

Fenêtre GSC-Runtime

GSC-Runtime est une fenêtre qui affiche le comportement dynamique de toutes les actions (Global Script) au runtime. En outre GSC-Runtime permet de piloter l'exécution de chaque action et de basculer dans l'éditeur Global Script au runtime.

Les informations suivantes sont affichées:

- Nom de l'action: Le nom de l'action
- ID : Les ID d'action. Ces identificateurs sont utilisés en interne par le système et affichés avec la description de l'erreur par la fonction OnErrorExecute, par ex. lorsqu'une erreur apparaît dans une action . GSC-Runtime fournit le nom de l'action correspondant à l'ID de l'action. L'association de l'ID au nom de l'action n'est valide que jusqu'à l'arrêt du runtime ou, lorsque le runtime est actif, jusqu'à l'enregistrement d'une action.
- Etat: Renseigne sur l'état courant de l'action. Les états possibles sont indiqués dans le tableau ci-dessous
- Intervalle d'activation: C'est le temps au format Heure:Minute:Seconde qui s'est écoulé entre deux appels de l'action
- Retour: la valeur retournée par l'action
- Démarré le: Date et heure du déclenchement de l'action
- Prochain dém: Date et heure du prochain déclenchement de l'action
- Message d'erreur: contient le texte d'erreur en cas d'erreur



Global Script - Runtime

Action	ID	Etat	Intervalle d'activati...	Retour	démarré le :	Prochain démarrage	Message d'erreur :
TankLevel.pas	@5	Action en cours	0: 0: 2	10	28/09/00 11:15:27		
gsccs1_2.pas	@6	La fin d'action a été déclarée	0: 0: 2	0	28/09/00 11:14:39		
gsccs1_1.pas	@7	Le début d'action a été déclaré		0	28/09/00 11:14:21	29/09/00 11:14:03	
gsccs1.pas	@8	Action en cours	0: 0: 2	0	28/09/00 11:15:27		

Etats des actions

Les différents états possibles d'une action:

- Le début d'action a été déclaré
- La fin d'action a été déclarée
- L'action a été arrêtée
- Action en cours

- Erreur lors de la déclaration de début d'action !
- Erreur lors de l'exécution de l'action !

Messages d'erreur

Les messages d'erreur suivants sont possibles:

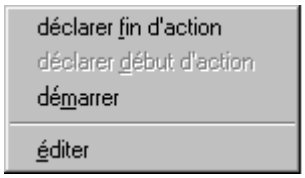
- Aucune erreur n'est survenue
- L'application est déjà reliée à la commande d'action. L'établissement d'une nouvelle liaison n'est pas possible.
- Pas de liaison à la commande d'action. La liaison n'a éventuellement pas été établie.
- Une erreur est survenue lors de la communication interprocess. La cause de l'erreur est inconnue.
- Erreur non définie.
- Erreur de paramétrage. Il se peut que certains paramètres n'aient pas été définis.
- La commande d'action n'a pas pu être lancée. Vérifiez si WinCC a été démarré.
- Time out. Contrôlez la liaison ou prolongez le temps de surveillance.
- Commande d'action terminée.
- La voie de maintenance n'a pas pu être installée.
- Le numéro affecté au contrat EndAct est inconnu.
- Une erreur s'est produite durant l'exécution de l'action. Les informations en retour ne sont pas valables.
- Une erreur s'est produite dans l'application de serveur.
- Le nombre maximal de liaisons pour la commande d'action est atteint.
- Transaction inconnue. Tentative de clôture d'une transaction qui n'a pas été préalablement déclarée.
- Il n'est pas possible de générer un fichier d'en-tête précompilé à partir d'un fichier d'en-tête précompilé.
- Cette action n'est pas accessible. Le module est en cours d'utilisation.
- Programme non valide.
- Action non valide.
- Le fichier n'a pas pu être créé par la commande d'action.
- La mémoire dont dispose l'interpréteur d'action ne suffit plus.
- Format de fichier non valide pour la commande d'action.
- La commande d'action n'a pas pu ouvrir le fichier.
- Le programme est momentanément bloqué par la commande d'action. Accès impossible.
- L'action a déjà été transférée à la commande d'action pour traitement.
- Conflit avec une autre action au sein de l'action.
- L'action n'a pas été trouvée par la commande d'action.

- La fonction n'a pas été trouvée par la commande d'action.
- Information de ligne spécifiée pas valide.
- Symbole spécifié hors du domaine de validité.
- La mémoire mise à disposition est insuffisante pour l'interpréteur d'action.
- L'interpréteur d'action ne connaît pas le type spécifié.
- Symbole spécifié introuvable.
- Chargement des fonctions de projet.
- Un débordement de pile s'est produit durant l'exécution dans l'interpréteur d'action. L'exécution de l'action a été abandonnée.
- Une division par 0 est survenue durant l'exécution de l'action. L'action a été abandonnée.
- Durant l'exécution, il a été fait référence au sein de l'action à un symbole introuvable.
- Accès à une mémoire non définie au sein de l'action durant son exécution.
- L'interpréteur d'action a rencontré un point de rupture.
- L'opération suivante de l'interpréteur d'action a été exécutée au sein du débogueur.
- L'action ne contient pas de code interpréteur.
- Le format de données de l'action est erroné.
- La valeur en retour de l'action n'est pas représentée comme variant.
- L'application permettant d'exécuter cette opération n'est plus disponible en mémoire.
- Une erreur s'est produite dans l'application. Vous trouverez une description détaillée de l'erreur dans les AP_ACT_KEY.
- Une erreur s'est produite dans l'exécution de l'action. Vous trouverez une description détaillée dans les AP_ACT_KEY.
- Une erreur s'est produite dans l'exécution de l'action. Vous trouverez une description détaillée dans les AP_ACT_KEY.
- Une mise à jour du présent format de données n'est pas possible. L'action ne peut être lue.

Menu contextuel correspondant aux actions

Le menu contextuel de chaque action contient les fonctions suivantes:

- Déclarer fin d'action: L'action concernée ne sera plus exécutée à l'issue de l'exécution en cours
- Déclarer début d'action: L'action concernée sera à nouveau exécutée au prochain déclenchement
- Démarrer: L'action concernée est exécutée une fois
- Editer: L'action concernée est ouverte dans l'éditeur Global Script pour édition. Le runtime reste actif. Lorsque l'action exécutée est compilée (si nécessaire) et enregistrée, les modifications sont prises en compte immédiatement par le système runtime



Vous pouvez, pour chaque action, gérer la possibilité d'ouvrir le menu contextuel en attribuant une autorisation.

Pour utiliser GSC-Runtime vous devez insérer une fenêtre d'application du type GSC-Runtime dans une vue de process. Les attributs de GSC-Runtime permettent de configurer l'apparence de la fenêtre de GSC-Runtime.

Remarque

Le rafraîchissement de la fenêtre GSC-Runtime augmente la charge de calcul du système. La charge du système dépend du nombre d'actions affichées dans la fenêtre. Vous pouvez réduire la charge du système en diminuant la hauteur de la fenêtre pour qu'elle affiche moins de lignes.

Voir aussi

Pour éditer une action (Page 895)

Les attributs de GSC-Runtime (Page 895)

Pour placer GSC-Runtime dans une vue de process (Page 894)

Pour attribuer une autorisation (Page 885)

2.14.2.2 Pour placer GSC-Runtime dans une vue de process

Introduction

Pour utiliser GSC-Runtime vous devez insérer GSC-Runtime dans une vue de process. Cette vue de process peut être une vue existante ou une vue servant uniquement au diagnostic. Il n'est pas possible d'insérer GSC-Runtime directement dans la vue de process; il doit être placé comme application dans une fenêtre d'application. La fenêtre d'application elle-même est alors un élément de la vue de process. Les actions décrites doivent être exécutées dans Graphics Designer.

Condition

Graphics Designer est lancé et la vue de process est ouverte.

Marche à suivre

1. Sélectionnez dans la palette d'objets "Objets complexes\Fenêtre d'application"
2. Agrandissez la fenêtre d'application en tirant dessus dans la surface de dessin
3. Sélectionnez dans le dialogue "Contenu de la fenêtre" la commande "Global Script"
4. Validez par "OK"
5. Sélectionnez dans le dialogue "Modèle" la commande "GSC-Runtime"
6. Validez par "OK"

Voir aussi

Les attributs de GSC-Runtime (Page 895)

2.14.2.3 Les attributs de GSC-Runtime

Apparence de la fenêtre de GSC-Runtime

GSC-Runtime possède des attributs qui permettent de configurer l'apparence de la fenêtre GSC-Runtime au Runtime. Ce sont les attributs de géométrie et en particulier les suivants:

- Affichage: Cet attribut sert à paramétrer si la fenêtre est affichée ou masquée. Cet attribut est dynamisable par le nom "Visible"
- Taille modifiable: indique si la fenêtre peut être redimensionnée au Runtime
- Déplaçable: Indique si la fenêtre doit pouvoir être déplacée au Runtime
- Bordure: Indique si la fenêtre doit être entourée d'une bordure. Si la fenêtre a une bordure, sa hauteur et sa largeur sont redimensionnables au Runtime
- Titre: Indique si la fenêtre doit posséder un titre
- Maximisable: Indique si la barre de titre contient le bouton Maximiser de la fenêtre
- Peut être fermé: indique si la barre de titre contient le bouton servant à fermer la fenêtre
- Premier plan: indique si la fenêtre doit toujours se trouver au premier plan

Les attributs sont affichés et se paramètrent dans Graphics Designer.

2.14.2.4 Pour éditer une action

Introduction

Toutes les actions contenues dans votre projet sont affichées, chacune sur une ligne, dans la fenêtre GSC-Runtime. Vous pouvez ouvrir une action dans la fenêtre GSC-Runtime et l'éditer dans l'éditeur Global Script. Après l'enregistrement l'action éditée est utilisée au Runtime.

Marche à suivre

1. Ouvrez le menu contextuel de l'action concernée
2. Sélectionnez la commande "Edition"

Voir aussi

Création et édition d'actions (Page 870)

2.14.3 Diagnostic GSC

2.14.3.1 Diagnostic GSC

Description de la fonction

Diagnostic GSC affiche dans la fenêtre de diagnostic, dans l'ordre chronologique de leur appel, les instructions printf contenues dans les actions. Il affiche également les instructions printf codées dans les fonctions appelées dans des actions. L'utilisation ciblée d'instructions printf, par ex. pour afficher des valeurs de variable, permet de suivre l'exécution d'actions et des fonctions appelées par celles-ci. Les situations génératrices d'erreurs qui entraînent l'appel de la fonction OnErrorExecute sont affichées dans la fenêtre de diagnostic.

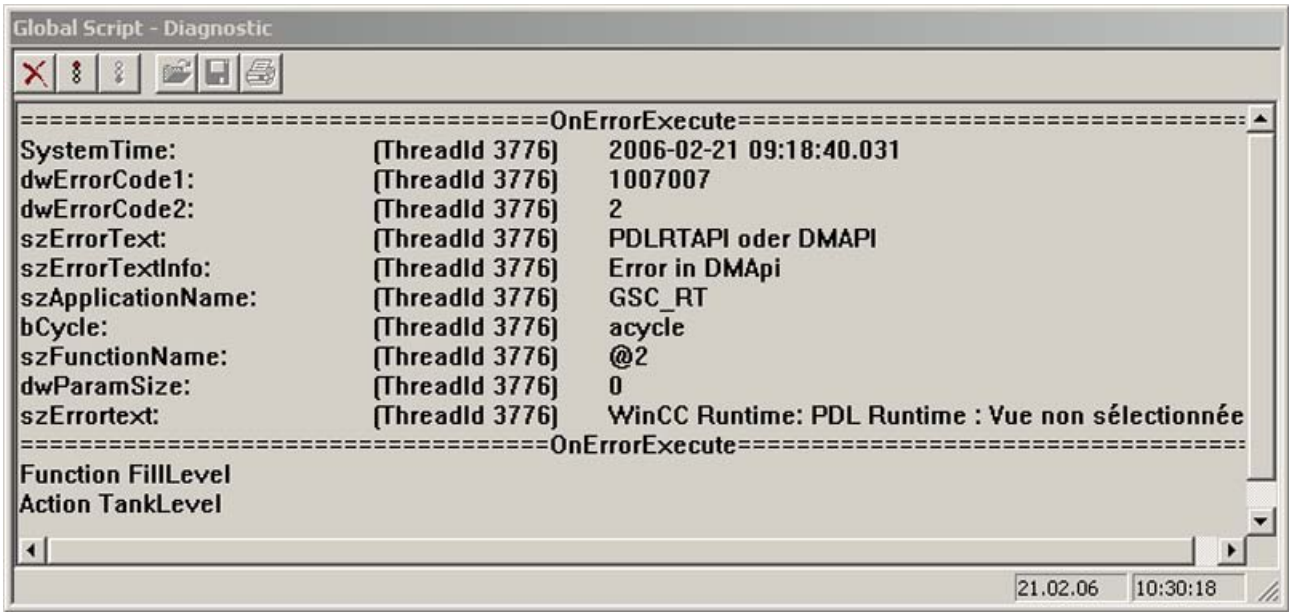


Figure 2-1 La fenêtre de diagnostic de Diagnostic GSC

Pour utiliser Diagnostic GSC vous devez placer une fenêtre d'application du type Diagnostic GSC dans une vue de process. Les attributs de Diagnostic GSC permettent de paramétrer l'apparence de la fenêtre de Diagnostic GSC.

La fenêtre de Diagnostic GSC est supprimée aux changements de vues.

Remarque

Un printf() peut comprendre au maximum 360 caractères.

Voir aussi

La barre d'outils du diagnostic GSC (Page 898)

Les attributs de GSC-Runtime (Page 895)

Pour placer Diagnostic GSC dans une vue de process (Page 897)

2.14.3.2 Pour placer Diagnostic GSC dans une vue de process

Introduction

Pour utiliser Diagnostic GSC, vous devez placer Diagnostic GSC dans une vue de process. Cette vue de process peut être une vue existante ou une vue servant uniquement au diagnostic. Diagnostic GSC ne peut pas être inséré directement comme application dans la vue de process, mais doit être placé comme application dans une fenêtre d'application. La fenêtre d'application elle-même est alors un élément de la vue de process. Les actions décrites doivent être exécutées dans Graphics Designer.

Condition

Graphics Designer est lancé et la vue de process est ouverte.

Marche à suivre

1. Sélectionnez dans la palette d'objets "Objets complexes\Fenêtre d'application"
2. Agrandissez la fenêtre d'application en tirant dessus dans la surface de dessin
3. Sélectionnez dans le dialogue "Contenu de la fenêtre" la commande "Global Script"
4. Validez par "OK"
5. Sélectionnez dans le dialogue "Modèle" la commande "Diagnostic GSC"
6. Validez par "OK"

Voir aussi

Les attributs de GSC-Runtime (Page 895)

La barre d'outils du diagnostic GSC (Page 898)

2.14.3.3 Les attributs de Diagnostic GSC

Les attributs de Diagnostic GSC

Diagnostic GSC possède des attributs qui permettent de configurer l'apparence de la fenêtre de diagnostic GSC au Runtime. Ce sont les attributs de géométrie et en particulier les suivants:

- Affichage: Cet attribut sert à paramétrer si la fenêtre est affichée ou masquée. Cet attribut est dynamisable par le nom "Visible"
- Taille modifiable: indique si la fenêtre peut être redimensionnée au Runtime
- Déplaçable: Indique si la fenêtre doit pouvoir être déplacée au Runtime
- Bordure: Indique si la fenêtre doit être entourée d'une bordure. Si la fenêtre a une bordure, sa hauteur et sa largeur sont redimensionnables au Runtime
- Titre: Indique si la fenêtre doit posséder un titre
- Maximisable: Indique si la barre de titre contient le bouton Maximiser de la fenêtre
- Peut être fermé: indique si la barre de titre contient le bouton servant à fermer la fenêtre
- Premier plan: indique si la fenêtre doit toujours se trouver au premier plan

Les attributs sont affichés et se paramètrent dans Graphics Designer.

2.14.3.4 La barre d'outils du diagnostic GSC

Fonctions de la barre d'outils

La barre d'outils de Diagnostic GSC permet de commander l'affichage dans la fenêtre de diagnostic ainsi que l'enregistrement, l'impression et l'ouverture du contenu de la fenêtre.

La barre d'outils contient des boutons pour les fonctions suivantes:

Bouton	Fonction
	supprime le contenu de la fenêtre
	stoppe le rafraîchissement de la fenêtre
	reprend le rafraîchissement de la fenêtre
	ouvre un fichier texte dans la fenêtre
	enregistre le contenu de la fenêtre dans un fichier texte
	imprime le contenu de la fenêtre

2.15 ANSI-C Descriptions de fonctions

2.15.1 lpszPictureName

Vue d'ensemble

"lpszPictureName" est le nom de la vue.

Si vous configurez dans WinCC une action sur une propriété ou sur l'événement "Clic de souris", le nom de la vue vous est fourni dans l'action en tant que "lpszPictureName". Le nom de la vue a alors la structure suivante :

<NOM DE VUE DE BASE>.<NOM DE FENÊTRE DE VUE>:<NOM DE VUE>. ... .<Nom de fenêtre de vue>:<Nom de vue>.

Le "NOM DE VUE DE BASE" et le "NOM DE VUE" sont fournis sans l'extension de fichier ".PDL".

Vous pouvez ainsi constater quel est le chemin de vue de l'objet. Vous pouvez également démarrer de manière ciblée des fenêtres de vue données, p. ex. lorsqu'une vue de process est ouverte plus d'une fois.

Remarque

Vous n'avez pas le droit de modifier le texte dans "lpszPictureName" ni avec la fonction "strcat".

2.15.2 Fonctions standard

2.15.2.1 Fonctions standard - Description sommaire

Le système met à disposition des fonctions standard. Vous pouvez modifier ces fonctions et les adapter ainsi à vos besoins. Vous pouvez en outre créer vos propres fonctions standard.

Le système de base met à votre disposition des fonctions standard. Elles sont divisées en groupes de fonction comme suit :

- Alarm
- Graphics
- Rapport
- TagLog
- WinCC
- Windows

Le répertoire "Obsolete functions" contient des fonctions standard utilisées avant WinCC V7 pour la commande des contrôles.

Si les options correspondantes sont installées, les groupes de fonctions supplémentaires suivantes sont disponibles :

- Options
- Split Screen Manager
- userarc (Archives utilisateur)

2.15.2.2 Alarm

AcknowledgeMessage

Fonction

Acquitte, dans le système d'alarme, l'alarme avec le numéro qui est transmis comme paramètre.

Syntaxe

void AcknowledgeMessage (DWORD MsgNr)

Paramètres

MsgNr

Alarme à acquitter

Remarque

Assurez-vous qu'il existe une alarme configurée correspondant au numéro d'alarme transmis. Si vous voulez utiliser les fonctions sur un Client avec un propre projet, un serveur standard doit y être configuré pour alarmes.

Voir aussi

Exemple AcknowledgeMessage

AXC_SetFilter

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction place un filtre pour le WinCC Alarm Control pour afficher une partie des alarmes existantes conformément au critère de filtrage.

Syntaxe

BOOL AXC_SetFilter(char* lpszPictureName, char* lpszObjectName, LPMSG_FILTER_STRUCT lpMsgFilter, LPCMN_ERROR, lpError)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom WinCC Alarm Control

lpMsgFilter

Pointeur sur la structure avec le critère de filtrage.

lpError

Pointeur sur la structure de la description d'erreurs.

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Définition de structure CMN_ERROR

Exemple ResetFilter

Exemple AXC_SetFilter

Définition de structure MSG_FILTER_STRUCT

GCreateMyOperationMsg

Fonction

La fonction standard "GCreateMyOperationMsg" vous permet de déclencher une autorisation de conduite propre dans le système d'alarmes. Vous avez sûrement déjà configuré l'alarme portant le numéro "dwMsgNum" comme message de conduite.

Syntaxe

int GCreateMyOperationMsg(DWORD dwFlags, DWORD dwMsgNum, char* lpszPictureName, char* lpszObjectName, DWORD dwMyTextID, double doValueOld, double doValueNew, char* pszComment)

Paramètres

dwFlags

Vous pouvez sélectionner la catégorie de l'alarme avec le paramètre "dwFlags".

Nom	Valeur	Signification
FLAG_COMMENT_PARAMETER	0x00000001	Le texte s'inscrit directement dans l'alarme en Runtime en tant que commentaire, sans dialogue Commentaire propre. Le pointeur sur commentaire doit être inégal à "ZERO".
FLAG_COMMENT_DIALOG	0x00000003	Un dialogue Commentaire s'affiche. Le commentaire y étant spécifié est repris dans l'alarme.
FLAG_TEXTID_PARAMETER	0x00000100	L'identificateur de texte d'un texte provenant de la TextLibrary est ajouté en tant que valeur associée de process de l'alarme.

dwMsgNum

Numéro d'alarme WinCC d'un message de conduite créé.

lpszPictureName

Pointeur sur le nom de la vue depuis laquelle la fonction est appelée.

lpszObjectName

Pointeur sur le nom de variable WinCC auquel appartiennent les valeurs anciennes et les nouvelles valeurs.

Le nom est transmis comme nom d'instance du message de conduite et inscrit dans la valeur associée de processus "1".

dwMyTextID

Identificateur de texte d'un texte provenant de la TextLibrary.

Si ""FLAG_TEXTID_PARAMETER" est activé, l'identificateur de texte est ajouté à l'alarme comme valeur associée de process numérique "8" et indiqué en tant que chiffre dans le bloc

de valeurs de process 8. Pour que le texte dépendant de la langue, provenant de la TextLibrary, apparaisse dans l'alarme, vous devez taper l'instruction de format "@8%s@" dans le bloc de texte d'alarme.

doValueOld

Ancienne valeur numérique des variables WinCC avec le nom spécifié dans "lpszObjectName". "doValueOld" est inscrit dans la valeur associée de process "2" de l'alarme. La fonction n'a pas elle-même la possibilité d'extraire une valeur de valable avant l'action. Utilisez pour cela les fonctions "GetTag..." disponibles.

doValueNew

Nouvelle valeur numérique des variables WinCC avec le nom spécifié dans "lpszObjectName". "doValueNew" est inscrit dans la valeur associée de process "3" de l'alarme.
La fonction n'a pas elle-même la possibilité d'extraire une valeur de valable après l'action. Utilisez pour cela les fonctions "GetTag..." disponibles.

pszComment

Texte de commentaire ou string vide.

Si "FLAG_COMMENT_PARAMETER" est activé, le texte s'inscrit directement comme commentaire dans l'alarme en Runtime. Pour cela, l'alarme n'a pas besoin de dialogue Commentaire propre.

Valeur en retour

Valeur	Signification
0	La fonction a été exécutée sans erreur.
-101	L'édition d'alarme n'a pas pu être démarrée.
-201	Une erreur s'est produite lors de l'appel de la fonction "MSRTGetComment()".
-301	Une erreur s'est produite lors de l'appel de la fonction "MSRTCreateMsgInstanceWithComment()".

Remarque

Assurez-vous que seules des alarmes de conduite sont utilisées pour la fonction "GCreateMyOperationMsg". L'utilisation d'alarmes provenant d'autres classes d'alarmes n'est pas autorisée.

Tenez compte du rôle du serveur standard au niveau du client lors de l'utilisation de la fonction. Pour plus d'informations, consultez le chapitre "Configuration du client".

GMsgFunction

Fonction

Cette fonction fournit les données d'alarme.

Elle représente une fonction globale pour alarmes individuelles. Elle est appelée lors de chaque alarme pour laquelle le paramètre "Déclenche une action" a été activé.

Une évaluation des données d'alarme est exécutée de préférence dans une fonction de projet qui est ensuite appelée à partir de GMsgFunction.

Syntaxe

BOOL GMsgFunction(char* pszMsgData)

Paramètres

pszMsgData

Pointeur sur une chaîne de caractères dont les données sont reproduites avec scanf sur la structure MSG_RTDATA_STRUCT.

Le string "MSG_RTDATA_STRUCT" contient les donnée suivantes, séparées entre elles par "#" :

1. Heure télégramme
2. Valeurs de process
3. Instance
4. Utilisateur
5. Ordinateur
6. Heure arrivée au format "yyyy.mm.dd, hh:mm:ss.mmm"

Remarque

La valeur "Instance" du string "MSG_RTDATA_STRUCT" est seulement affectée quand une alarme d'instance a été déclenchée.

Les valeurs "Utilisateur" et "Ordinateur" du string "MSG_RTDATA_STRUCT" sont seulement affectées quand un commentaire a été ajouté lors de la création de l'alarme avec le même appel.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Veuillez noter que des fonctions standard modifiées sont écrasées par une installation WinCC et que les modifications sont ainsi perdues.

Voir aussi

Définition de structure MSG_RTDATA_STRUCT

2.15.2.3 Graphics

Graphics - brève description

Le groupe Graphics contient des fonctions pour la programmation du système Graphic.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetLinkedVariable

Fonction

Fournit le nom de la variable qui est reliée à une propriété d'objet précise.

Syntaxe

char* GetLinkedVariable(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName);

Paramètres

lpszPictureName

Pointeur sur la vue

lpszObjectName

Pointeur sur l'objet

lpszPropertyName

Pointeur sur la propriété de l'objet

Valeur de retour

Pointeur sur le nom de la variable qui est reliée à une propriété d'objet précise

Voir aussi

Exemple GetLinkedVariable

GetLocalPicture

Fonction

Fournit un pointeur sur le nom de la vue. Le nom de vue y est le nom de fichier sans l'extension ".PDL".

Syntaxe

char* GetLocalPicture(char *lpszPictureName);

Paramètres

lpszPictureName

Pointeur sur la vue

Valeur de retour

Pointeur sur le nom de la vue.

Remarque

Le paramètre d'appel transmis lpszPictureName doit avoir la structure que crée le système graphique pour les chemin de vue :

<Nom de la vue de base>.<Nom de la fenêtre de vue>:<Nom de la vue>. ... .<Nom de la fenêtre de vue>:<Nom de la vue>,

<Nom de la vue de base> et <Nom de la vue> y sont intégrés sans l'extension de fichier ".PDL".

Exemple:

Dans une vue de base "AAA" se trouve une fenêtre de vue "bbb" dans laquelle est appelée une vue "CCC" qui, à son tour, contient une fenêtre de vue "ddd" dans laquelle est appelée une vue "EEE".

Ensuite, l'appel de fonction fournit

```
GetLocalPicture(lpszPictureName)
```

le pointeur sur le nom de la vue :

"EEE" renvoyé lorsque la fonction est appelée dans la vue "EEE";

"CCC" renvoyé lorsque la fonction est appelée dans la vue "CCC";

"AAA" renvoyé lorsque la fonction est appelée dans la vue "AAA".

Voir aussi

Exemple GetLocalPicture

GetParentPicture

Fonction

Fournit un pointeur sur le nom de la vue. Le nom de vue y est le nom de fichier sans l'extension ".PDL".

Syntaxe

char* GetParentPicture(char *lpszPictureName);

Paramètres

lpszPictureName

Pointeur sur la vue

Valeur de retour

Nom de la vue actuelle lorsque la fonction est appelée dans la vue de base

Chemin de nom de la vue supérieure lorsque la fonction est appelée dans une fenêtre de vue

---

Remarque

Le paramètre d'appel transmis lpszPictureName doit avoir la structure que crée le système graphique pour les chemin de vue :

<Nom de la vue de base>.<Nom de la fenêtre de vue>:<Nom de la vue>. ... .<Nom de la fenêtre de vue>:<Nom de la vue>,

<Nom de la vue de base> et <Nom de la vue> y sont intégrés sans l'extension de fichier ".PDL".

Voir aussi

Exemple GetParentPicture

GetParentPictureWindow

Fonction

Fournit un pointeur sur le nom de la fenêtre de vue.

Syntaxe

char *GetParentPictureWindow(char *lpszPictureName);

Paramètres

lpszPictureName

Pointeur sur la vue

Valeur de retour

Pointeur sur le nom de la fenêtre de vue lorsque la fonction est appelée dans une vue qui est affichée dans une fenêtre de vue d'une vue supérieure.

Paramètre d'appel lpszPictureName non changé lorsque la fonction est appelée dans la vue de base

---

Remarque

Le paramètre d'appel transmis lpszPictureName doit avoir la structure que crée le système graphique pour les chemin de vue :

<Nom de la vue de base>.<Nom de la fenêtre de vue>:<Nom de la vue>. ... .<Nom de la fenêtre de vue>:<Nom de la vue>,

<Nom de la vue de base> et <Nom de la vue> y sont intégrés sans l'extension de fichier ".PDL".

Exemple:

Dans une vue de base "Bild_1" se trouve une fenêtre de vue "Bildfenster_1" dans laquelle est appelée une vue "Bild_2".

Dans la vue "Bild_2" se trouve une fenêtre de vue "Bildfenster_2" dans laquelle est appelée une vue "Bild_3".

Ensuite, l'appel de fonction fournit

```
GetParentPictureWindow(lpszPictureName)
```

le pointeur sur la fenêtre de vue :

"Bild_2" renvoyé lorsque la fonction est appelée dans la vue "Bild_3";

"Bildfenster_1" renvoyé lorsque la fonction est appelée dans la vue "Bild_2";

"Bild_1" renvoyé lorsque la fonction est appelée dans la vue "Bild_1";

OpenPicture

Fonction

Réalise un changement de la vue de base indiquée. Au client et dans le cas d'un nom de vue avec préfixe de serveur, un changement de vue est réalisé dans la fenêtre de vue.

Si la fenêtre de vue se trouve par ex. dans une autre fenêtre de vue avec préfixe de serveur, le changement de vue n'est pas exécuté dans la fenêtre dans laquelle la fonction a été appelée.

Syntaxe

void OpenPicture(Picture Nom de vue)

Paramètres

Nom de vue

Nom de la vue

Registry2

Fonction

Cette fonction gère une liste de paires de chaînes de caractères (String0, String1).

Elle connaît les types d'appel suivants qui sont commandés via le paramètre mode :

- Registry2("set", "String0", "String1");

Enregistre la paire de chaînes de caractères transmise dans la liste.

- Registry2("get", "String0", NULL);

Renvoie le premier partenaire de la paire de chaînes de caractère, String1, qui appartient à la chaîne de caractère transmise String0 et supprime ensuite la paire de chaînes de caractères de la liste.

- Registry2("reset", NULL, NULL);

Supprime toutes les paires de chaînes de caractères de la liste.

- Registry2("display", NULL, NULL);

Affiche, dans une fenêtre de diagnostic Global Script, les paires de chaînes de caractères enregistrées actuellement dans la liste.

Syntaxe

char* Registry2(char* mode, char* String0, char* String1);

Paramètres

mode

Détermine le fonctionnement de la fonction.

set	Enregistrement de la paire de chaînes de caractères dans la liste.
get	Détermination du premier partenaire de la paire de chaînes de caractères appartenant à String0 et suppression de la paire de la liste.
reset	Suppression de toutes les paires de chaînes de caractères
display	Affichage des paires de chaînes de caractères dans une fenêtre de diagnostic Global Script.

String0

L'affectation du paramètre dépend du fonctionnement de la fonction.

String1

L'affectation du paramètre dépend du fonctionnement de la fonction.

Valeur de retour

Dans le mode mode=get, un pointeur est renvoyé sur le premier partenaire de la paire de chaînes de caractères.

Remarque

Cette fonction est utilisée en rapport avec la technique à vues modulables.

Dans Dynamic Wizard, quand vous travaillez sous l'onglet "Blocs d'affichage" avec les assistants "Créer bloc d'affichage comme type" et "Créer instance(s) dans la vue de l'installation", vous ne devez pas utiliser la fonction "Registry2" !

2.15.2.4 Obsolete functions

Alarm

AXC_OnBtnAlarmHidingList

Fonction

La fonction affiche la liste des alarmes masquées dans une fenêtre d'alarmes.

Syntaxe

BOOL AXC_OnBtnAlarmHidingList(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

AXC_OnBtnArcLong

Fonction

La fonction affiche les alarmes enregistrées dans une liste d'archives longues dans une fenêtre d'alarmes.

Syntaxe

BOOL AXC_OnBtnArcLong (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnArcShort

Fonction

La fonction affiche les alarmes enregistrées dans une liste d'archives courtes dans une fenêtre d'alarmes.

Syntaxe

BOOL AXC_OnBtnArcShort (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnComment

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction montre le commentaire de l'alarme sélectionnée auparavant.

Syntaxe

BOOL AXC_OnBtnComment (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnEmergAckn

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction ouvre le dialogue d'acquittement (Acquittement d'urgence/Réinitialisation).

Syntaxe

BOOL AXC_OnBtnEmergAckn (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnHideDlg

Fonction

Cette fonction ouvre le dialogue d'options d'affichage dans lequel sont définies les alarmes qui s'afficheront dans la fenêtre d'alarmes. Les options sont "Toutes les alarmes", "Alarmes affichées" ou "alarmes masquées".

Syntaxe

BOOL AXC_OnBtnHideDlg(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

AXC_OnBtnHideUnhideMsg

Fonction

La fonction masque l'alarme sélectionnée ou affiche de nouveau l'alarme masquée.

Syntaxe

BOOL AXC_OnBtnHideUnhideMsg (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

AXC_OnBtnHit

Fonction

La fonction affiche les alarmes enregistrées dans la liste d'occurences dans une fenêtre d'alarmes.

Syntaxe

BOOL AXC_OnBtnHit (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnHornAckn

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction acquitte le signal de l'avertisseur sonore.

Syntaxe

BOOL AXC_OnBtnHornAckn (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnInfo

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction affiche le texte d'info.

Syntaxe

BOOL AXC_OnBtnInfo (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnLock

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction ouvre le dialogue "Paramétrer la liste de verrouillages".

Syntaxe

BOOL AXC_OnBtnLock (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnLockUnlock

Fonction

La fonction verrouille l'alarme sélectionnée dans la fenêtre d'alarmes. Cette alarme n'est plus archivée après.

La fonction déverrouille l'alarme sélectionnée dans la liste de verrouillages.

Syntaxe

BOOL AXC_OnBtnLockUnlock (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnLockWin

Fonction

Commande externe des fenêtres d'alarmes.

Cette fonction appelle la liste de verrouillages.

Syntaxe

BOOL AXC_OnBtnLockWin (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnLoop

Fonction

Commande externe des fenêtres d'alarmes

Avec cette fonction, la fonction "LoopInAlarm" de l'alarme sélectionnée est déclenchée.

Syntaxe

BOOL AXC_OnBtnLoop (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnMsgFirst

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction passe au début de la liste d'alarmes.

Syntaxe

BOOL AXC_OnBtnMsgFirst (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnMsgLast

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction passe à la fin de la liste d'alarmes.

Syntaxe

BOOL AXC_OnBtnMsgLast (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgLast

AXC_OnBtnMsgNext

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction passe à l'alarme suivante dans la liste d'alarmes.

Syntaxe

BOOL AXC_OnBtnMsgNext (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnMsgPrev

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction passe à l'alarme précédente dans la liste d'alarmes.

Syntaxe

BOOL AXC_OnBtnMsgPrev (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnMsgWin

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction appelle la liste d'alarmes.

Remarque

La liste d'alarmes contient les alarmes actuellement présentes et non acquittées.

Syntaxe

BOOL AXC_OnBtnMsgWin (char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnPrint

Fonction

Commande externe des fenêtres d'alarmes

Toutes les alarmes conformes au critère de sélection paramétré dans Alarm Control sont imprimées.

Syntaxe

BOOL AXC_OnBtnPrint(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnProtocol

Fonction

Commande externe des fenêtres d'alarmes

L'impression de la vue actuelle d'Alarm Controls démarre. Toutes les alarmes conformes au critère de sélection paramétré dans Alarm Control sont imprimées.

Syntaxe

BOOL AXC_OnBtnProtocol(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

AXC_OnBtnScroll

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction active ou désactive la fonction de défilement horizontal et vertical.

Syntaxe

BOOL AXC_OnBtnScroll(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnScroll

AXC_OnBtnSelect

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction ouvre le dialogue "Définir sélection" pour la liste affichée.

Syntaxe

BOOL AXC_OnBtnSelect(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

AXC_OnBtnSinglAckn

Fonction

Commande externe des fenêtres d'alarmes

Cette fonction acquitte l'alarme sélectionnée actuellement.

Syntaxe

BOOL AXC_OnBtnSinglAckn(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnSinglAckn

AXC_OnBtnSortDlg

Fonction

Commande externe de la fenêtre d'alarmes

Cette fonction ouvre le dialogue permettant de paramétrer un tri personnalisé des alarmes affichées pour la liste affichée.

Syntaxe

BOOL AXC_OnBtnSortDlg(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

AXC_OnBtnTimeBase

Fonction

Commande externe de la fenêtre d'alarmes

Cette fonction ouvre le dialogue servant au paramétrage de la base de temps pour les données horaires affichées dans les alarmes.

Syntaxe

BOOL AXC_OnBtnTimeBase(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

AXC_OnBtnVisibleAckn

Fonction

Commande externe des fenêtres d'alarmes

Toutes les alarmes visibles dans la fenêtre d'alarmes sont acquittées (acquittement groupé).

Syntaxe

BOOL AXC_OnBtnVisibleAckn(char* lpszPictureName, char* lpszObjectName)

Paramètres

lpszPictureName

Pointeur sur le nom de la vue dans laquelle se trouve WinCC Alarm Control

lpszObjectName

Pointeur sur le nom d'objet du WinCC Alarm Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC AlarmControl à partir de WinCC V7.0.

Voir aussi

Exemple AXC_OnBtnMsgFirst

Report

ReportJob

Fonction

Selon la valeur du paramètre lpMethodName, un travail d'impression ou l'aperçu avant impression d'un travail d'impression est démarré.

Syntaxe

void ReportJob(LPSTR lpJobName, LPSTR lpMethodName)

Paramètres

lpJobName

Pointeur sur le nom du travail d'impression

lpMethodName

PRINTJOB Travail d'impression est démarré
PREVIEW Aperçu avant impression d'un travail d'impression est démarré

Remarque

Cette fonction est remplacée par les fonctions RPTJobPreview et RPTJobPrint et ne devrait plus être utilisée.

TagLog

TOOLBAR_BUTTONS

TlgTableWindowPressEditRecordButton

Fonction

L'édition d'une fenêtre de tableau est bloquée ou autorisée (fonction Toggle).

Si l'édition est autorisée, l'actualisation de la fenêtre de tableau est arrêtée en même temps.

L'actualisation de la fenêtre de tableau reste arrêtée après, même si l'édition est bloquée par un autre appel de la fonction.

Syntaxe

BOOL TlgTableWindowPressEditRecordButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

TlgTableWindowPressFirstButton

Fonction

Affiche les premiers enregistrements de la zone de représentation dans la fenêtre de tableau.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTableWindowPressFirstButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressHelpButton

Fonction

Affiche l'aide en ligne pour la fenêtre de tableau.

Syntaxe

BOOL TlgTableWindowPressHelpButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressInsertRecordButton

Syntaxe

BOOL TlgTableWindowPressInsertRecordButton(char* lpszWindowName)

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

TlgTableWindowPressLastButton

Fonction

Affiche les derniers enregistrements de la zone de représentation dans la fenêtre de tableau.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTableWindowPressLastButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressNextButton

Fonction

Les enregistrements de la zone de représentation suivant la zone de représentation actuelle sont affichés dans la fenêtre de tableau.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTableWindowPressNextButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressNextItemButton

Fonction

Les colonnes de la fenêtre de tableau sont déplacées d'une colonne à gauche, la colonne gauche étant placée à la position de la colonne droite.

Syntaxe

BOOL TlgTableWindowPressNextItemButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressOpenArchiveVariableSelectionDlgButton

Fonction

Ouvre le dialogue pour la liaison de colonnes de tableau avec des archives et des variables.

Syntaxe

BOOL TlgTableWindowPressOpenArchiveVariableSelectionDlgButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressOpenDlgButton

Fonction

Ouvre le dialogue pour la configuration en ligne de la fenêtre de tableau.

Syntaxe

BOOL TlgTableWindowPressOpenDlgButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressOpenDlgButton

TlgTableWindowPressOpenItemSelectDlgButton

Fonction

Ouvre le dialogue pour la sélection des colonnes visibles et de la première colonne de la fenêtre de tableau.

Syntaxe

BOOL TlgTableWindowPressOpenItemSelectDlgButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressOpenTimeSelectDlgButton

Fonction

Ouvre le dialogue pour définir la plage de temps à représenter dans les colonnes de tableau.

Syntaxe

BOOL TlgTableWindowPressOpenTimeSelectDlgButton(char* lpszWindowNumber)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressPrevButton

Fonction

Les enregistrements de la zone de représentation devant la zone de représentation actuelle sont affichés dans la fenêtre de tableau.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTableWindowPressPrevButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressPrevItemButton

Fonction

Les colonnes de la fenêtre de tableau sont déplacées d'une colonne à droite, la colonne droite étant placée à la position de la colonne gauche.

Syntaxe

BOOL TlgTableWindowPressPrevItemButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTableWindowPressRemoveRecordButton

Syntaxe

BOOL TlgTableWindowPressRemoveRecordButton(char* lpszWindowName)

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

TlgTableWindowPressStartStopButton

Fonction

L'actualisation de la fenêtre de tableau est activée ou désactivée (fonction Toggle).

Syntaxe

BOOL TlgTableWindowPressStartStopButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressFirstButton

Fonction

Affiche les premiers enregistrements de la zone de représentation dans la fenêtre de courbe.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTrendWindowPressFirstButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressHelpButton

Fonction

Affiche l'aide en ligne pour la fenêtre de courbe.

Syntaxe

BOOL TlgTrendWindowPressHelpButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressLastButton

Fonction

Affiche les derniers enregistrements de la zone de représentation dans la fenêtre de courbe.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTrendWindowPressLastButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressLinealButton

Fonction

La règle de la fenêtre de courbe est affichée ou masquée (fonction Toggle).

La règle peut être déplacée via les touches "Curseur gauche" et "Curseur droite".

Syntaxe

BOOL TlgTrendWindowPressLinealButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressNextButton

Fonction

Les enregistrements de la zone de représentation suivant la zone de représentation actuelle sont affichés dans la fenêtre de courbe.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTrendWindowPressNextButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressNextItemButton

Fonction

Déplace toutes les courbes dans la fenêtre de courbe d'un niveau devant.

La courbe affichée au premier plan y est déplacée à l'arrière-plan.

Syntaxe

BOOL TlgTrendWindowPressNextItemButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressOneToOneButton

Fonction

Rétablit la taille standard (1/1) dans la fenêtre de courbe.

Syntaxe

BOOL TlgTrendWindowPressOneToOneButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressOpenArchiveVariableSelectionDlgButton

Fonction

Ouvre le dialogue pour la liaison de courbes avec des archives et des variables.

Syntaxe

BOOL TlgTrendWindowPressOpenArchiveVariableSelectionDlgButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressOpenDlgButton

Fonction

Ouvre le dialogue pour la configuration en ligne de la fenêtre de courbe.

Syntaxe

BOOL TlgTrendWindowPressOpenDlgButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressOpenItemSelectDlgButton

Fonction

Ouvre le dialogue pour la sélection des courbes visibles et de la courbe qui doit être au premier plan.

Syntaxe

BOOL TlgTrendWindowPressOpenItemSelectDlgButton(char* lpszWindowNumber)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressOpenTimeSelectDlgButton

Fonction

Ouvre le dialogue pour déterminer la plage de temps à représenter.

Syntaxe

BOOL TlgTrendWindowPressOpenTimeSelectDlgButton(char* lpszWindowNumber)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressPrevButton

Fonction

Les enregistrements de la zone de représentation devant la zone de représentation actuelle sont affichés dans la fenêtre de courbe.

Le nombre des enregistrements affichés dépend de la plage de temps configurée.

Syntaxe

BOOL TlgTrendWindowPressPrevButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressPrevItemButton

Fonction

Déplace toutes les courbes dans la fenêtre de courbe d'un niveau vers l'arrière.

La courbe affichée en arrière-plan y est déplacée au premier plan.

Syntaxe

BOOL TlgTrendWindowPressPrevItemButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressPrintButton

Fonction

La vue actuelle des courbes s'affiche selon la représentation configurée du WinCC Trend Control.

Syntaxe

BOOL TlgTrendWindowPressPrintButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

TlgTrendWindowPressReportSaveButton

Fonction

Les données affichées de la fenêtre de courbe sont enregistrées dans un fichier texte.

Syntaxe

BOOL TlgTrendWindowPressReportSaveButton (char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control.

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

TlgTrendWindowPressStartStopButton

Fonction

L'actualisation de la fenêtre de courbe est activée ou désactivée (fonction Toggle).

Syntaxe

BOOL TlgTrendWindowPressStartStopButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressStatsResultButton

Fonction

Démarre l'évaluation des données dans la plage de temps sélectionnée.
Les valeurs statistiques Minimum, Maximum, Moyenne et Différence standard sont calculées.

Syntaxe

BOOL TlgTrendWindowPressStatsResultButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressStatsSelectRangeButton

Fonction

Pour la sélection de la plage de temps pour la fonction statistique, les règles pour le point de départ et le point final sont affichées.

Syntaxe

BOOL TlgTrendWindowPressStatsSelectRangeButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressStartStopButton

TlgTrendWindowPressZoomInButton

Fonction

La loupe dans la fenêtre de courbe est activée. La sélection de la zone de zoom n'est possible qu'à l'aide de la souris.

Syntaxe

BOOL TlgTrendWindowPressZoomInButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressZoomInButton

TlgTrendWindowPressZoomOutButton

Fonction

La fenêtre de courbe est ramenée à l'état qu'elle avait avant l'activation de la loupe. La loupe est désactivée.

La sélection de la zone de zoom n'est possible qu'à l'aide de la souris (voir aussi TlgTrendWindowPressZoomInButton).

Syntaxe

BOOL TlgTrendWindowPressZoomOutButton(char* lpszWindowName)

Paramètres

lpszWindowName

Pointeur sur le titre de fenêtre du WinCC Online Trend Control

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgTrendWindowPressZoomOutButton

Modèle

TlgGetNumberOfColumns

Fonction

Fournit le nombre des colonnes dans la fenêtre de tableau.

Le titre de fenêtre du WinCC Online Table Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

int TlgGetNumberOfColumns(char* lpszTemplate)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

Nombre de colonnes dans une fenêtre de tableau

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

TlgGetNumberOfRows

Fonction

Fournit le nombre des lignes dans la fenêtre de tableau.

Le titre de fenêtre du WinCC Online Table Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

int TlgGetNumberOfRows(char* lpszTemplate)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

Nombre des lignes dans la fenêtre de tableau

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgGetNumberOfRows

TlgGetNumberOfTrends

Fonction

Fournit le nombre des courbes dans la fenêtre de courbe.

Le titre de fenêtre du WinCC Online Trend Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

int TlgGetNumberOfTrends(char* lpszTemplate)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Trend Control.

Valeur de retour

Nombre des courbes dans la fenêtre de courbe

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

TlgGetRowPosition

Fonction

Fournit la position actuelle du pointeur de ligne dans la fenêtre de tableau.

Le titre de fenêtre du WinCC Online Table Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

int TlgGetRowPosition(char* lpszTemplate)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

Position actuelle du pointeur de ligne dans la fenêtre de tableau.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

TlgGetRulerArchivNameTrend

Fonction

Fournit le nom d'archive de la courbe avec le numéro nTrend dans la fenêtre de courbe à la position de la règle.

Le titre de fenêtre du WinCC Online Trend Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

char* TlgGetRulerArchivNameTrend(char* lpszTemplate, int nTrend)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Trend Control.

nTrend

Numéro de la courbe

(0 <= nTrend <= Nombre des courbes visibles - 1)

Valeur de retour

Nom d'archive de la courbe avec le numéro nTrend dans la fenêtre de courbe à la position de la règle

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgGetRulerVariableNameTrend

TlgGetRulerTimeTrend

Fonction

Fournit le temps de la courbe avec le numéro nTrend dans la fenêtre de courbe à la position de la règle.

Le titre de fenêtre du WinCC Online Trend Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

SYSTEMTIME TlgGetRulerTimeTrend(char* lpszTemplate, int nTrend)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Trend Control.

nTrend

Numéro de la courbe

(0 <= nTrend <= Nombre des courbes visibles - 1)

Valeur de retour

Temps de la courbe avec le numéro nTrend dans la fenêtre de courbe à la position de la règle

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgGetRulerTimeTrend (Page 1577)

TlgGetRulerValueTrend

Fonction

Fournit la valeur de la courbe avec le numéro nTrend dans la fenêtre de courbe à la position de la règle.

Le titre de fenêtre du WinCC Online Trend Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

double TlgGetRulerValueTrend(char* lpszTemplate, int nTrend)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Trend Control.

nTrend

Numéro de la courbe

(0 <= nTrend <= Nombre des courbes visibles - 1)

Valeur de retour

Valeur de la courbe avec le numéro nTrend dans la fenêtre de courbe à la position de la règle

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

TlgGetRulerVariableNameTrend

Fonction

Fournit le nom de variable de la courbe avec le numéro nTrend dans la fenêtre de courbe.

Le titre de fenêtre du WinCC Online Trend Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

char* TlgGetRulerVariableNameTrend(char* lpszTemplate, int nTrend)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Trend Control.

nTrend

Numéro de la courbe

(0 <= nTrend <= Nombre des courbes visibles - 1)

Valeur de retour

Nom de variable de la courbe avec le numéro nTrend dans la fenêtre de courbe.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgGetRulerVariableNameTrend

TlgGetTextAtPos

Fonction

Fournit le contenu d'une cellule de la fenêtre de tableau sous forme de texte pour des archives de valeurs processus et des archives utilisateur.

La cellule est spécifiée par nColumn et nLine.

Le titre de fenêtre du WinCC Online Table Control correspondant est transmis avec le paramètre lpszTemplate.

Syntaxe

char* TlgGetTextAtPos(char* lpszTemplate, int nColumn, int nLine)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

nColumn

Numéro de la colonne

nLine

Numéro de la ligne

Valeur de retour

Contenu d'une cellule de la fenêtre de tableau sous forme de texte

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgGetRulerVariableNameTrend

TlgGetColumnPosition

Fonction

Fournit la position actuelle du pointeur de colonne dans la fenêtre de tableau sous forme d'un indice de colonne.

Syntaxe

int TlgGetColumnPosition(char* lpszTemplate)

Paramètres

lpszTemplate

Pointeur sur le titre de fenêtre du WinCC Online Table Control.

Valeur de retour

Position actuelle du pointeur de colonne dans une fenêtre de tableau

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTableControl à partir de WinCC V7.0.

Voir aussi

Exemple TlgGetNumberOfColumns

TlgTrendWindowActivateCurve

Fonction

Active une certaine courbe dans le WinCC Online Trend Control via le nom configuré de la courbe. L'exécution de cette fonction ne dépend pas du fait si la courbe est visible ou se trouve au premier plan.

Syntaxe

BOOL TlgTrendWindowActivateCurve(char* lpszPictureName, char* lpszObjectName, char* szValue)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom du Trend Control

szValue

Nom de la courbe

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction standard n'est plus prise en charge pour le nouveau WinCC OnlineTrendControl à partir de WinCC V7.0.

2.15.2.5 Rapport

Journal - brève description

Le groupe Journal contient des fonctions avec lesquelles vous pouvez démarrer l'aperçu avant impression d'un travail d'impression ou le travail d'impression proprement dit.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

RPTJobPreview

Fonction

L'aperçu avant impression d'un travail d'impression est démarré.

Syntaxe

BOOL RPTJobPreview(LPSTR lpJobName)

Paramètres

lpJobName

Pointeur sur le nom du travail d'impression

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple RPTJobPreview

RPTJobPrint

Fonction

Un travail d'impression est démarré.

Syntaxe

BOOL RPTJobPrint(LPSTR lpJobName)

Paramètres

lpJobName

Pointeur sur le nom du travail d'impression

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple RPTJobPrint

RptShowError

Fonction

La fonction fournit une description d'erreurs pour un travail d'impression échoué.

L'appel de la fonction fait déjà partie des fonctions standard RptJobPrint et RptJobPreview, et ne demande donc pas un appel exprès.

La description d'erreurs est affichée dans une fenêtre de diagnostic Global Script.

Remarque

Comme RptShowError est une fonction standard, la façon et la forme de la sortie peuvent être modifiées si besoin est.

Veuillez noter que des fonctions standard modifiées sont écrasées par une installation WinCC et que les modifications sont ainsi perdues.

Syntaxe

void RptShowError(LPCSTR pszFailedFunction, CMN_ERRORA* pCmnErrorA)

Paramètres

pszFailedFunction

Pointeur sur le nom de la fonction échouée.

Si ce pointeur est NULL, le nom de la fonction n'est pas sorti.

pCmnErrorA

Pointeur sur la structure d'erreurs de la fonction échouée.

Si ce pointeur est NULL, la structure d'erreurs n'est pas sortie.

STRUKTUREN_TABELLEN_FEHLERSTRUKTUR

2.15.2.6 WinCC

WinCC - brève description

Le groupe WinCC contient des fonctions qui concernent tout le système WinCC.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetHWDiag

Fonction

Cette fonction vous permet de réaliser un lancement direct du diagnostic pendant l'exécution, le déclenchement provenant d'un événement à configurer pour un objet.

A l'occurrence de l'événement, la fonction "Diagnostic du matériel" de STEP7 est lancée pour la commande correspondante.

Les conditions préalables suivantes doivent être satisfaites pour que vous puissiez utiliser la fonction :

- Le projet WinCC avec la vue depuis laquelle doit se faire l'entrée et le projet STEP7 doivent se trouver sur le même ordinateur.
- Le projet WinCC doit être créé en tant que sous-répertoire du projet STEP7 (Projet STEP7\wincproj\Projet WinCC).
- Les variables S7 sont représentées sur WinCC.

Syntaxe

BOOL GetHWDiag(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpProperties)

Paramètres

lpszPictureName

Nom de la vue (fichier PDL) contenant les variables utilisées pour le passage au diagnostic du matériel.

Etant donné que la désignation "lpszPictureName" est utilisée pour la vue actuelle, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un objet se trouvant dans une autre vue.

lpszObjectName

Nom de l'objet dans la vue lié aux variables utilisées pour l'entrée dans le diagnostic du matériel.

Etant donné que la dénomination "lpszObjectName" est utilisée pour l'objet actuel, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un autre objet.

lpProperties

Nom de l'attribut lié à la variable utilisée pour l'entrée dans le diagnostic du matériel.

Si vous indiquez plusieurs attributs, vous devez les séparer par des points-virgules (";").

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

GetHWDiagLevel

Fonction

Effectue un contrôle d'autorisation pour l'utilisateur connecté à l'aide du numéro de fonction User Administrator indiqué sous dwLevel.

Le diagnostic est ensuite lancé directement pendant l'exécution - déclenché par un événement à configurer pour un objet.

A l'occurrence de l'événement, la fonction "Diagnostic du matériel" de STEP7 est lancée pour la commande correspondante.

Les conditions préalables suivantes doivent être satisfaites pour que vous puissiez utiliser la fonction :

- Le projet WinCC avec la vue depuis laquelle doit se faire l'entrée et le projet STEP7 doivent se trouver sur le même ordinateur.
- Le projet WinCC doit être créé en tant que sous-répertoire du projet STEP7 (Projet STEP7\wincproj\Projet WinCC).
- Les variables S7 sont représentées sur WinCC.
- Afin que l'utilisateur connecté dans WinCC puisse éditer le dialogue servant au diagnostic du matériel, il doit disposer de l'autorisation utilisateur WinCC avec le numéro transmis dans le paramètre "dwLevel" lors de l'appel de la fonction.

Syntaxe

BOOL GetHWDiagLevel(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpProperties, DWORD dwLevel)

Paramètres

lpszPictureName

Nom de la vue (fichier PDL) contenant les variables utilisées pour le passage au diagnostic du matériel.

Etant donné que la désignation "lpszPictureName" est utilisée pour la vue actuelle, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un objet se trouvant dans une autre vue.

lpszObjectName

Nom de l'objet dans la vue lié aux variables utilisées pour l'entrée dans le diagnostic du matériel.

Etant donné que la dénomination "lpszObjectName" est utilisée pour l'objet actuel, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un autre objet.

lpProperties

Nom de l'attribut lié à la variable utilisée pour l'entrée dans le diagnostic du matériel.

Si vous indiquez plusieurs attributs, vous devez les séparer par des points-virgules (";").

dwLevel

Numéro de niveau pour l'autorisation d'écriture STEP7

Ce numéro peut être déterminé dans User Administrator.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

GetKopFupAwl

Fonction

Cette fonction effectue l'entrée dans le réseau de WinCC dans l'éditeur STEP7 "KFA".

Lors de l'exécution de la fonction, deux tâches sont remplies :

- Détermination des données nécessaires à l'entrée dans le réseau au départ de WinCC.
- Transmission des données à Step7 et localisation du point d'emploi de l'opérande dans un programme STEP7 à l'aide de l'AUTAPI.

Syntaxe

BOOL GetKopFupAwl(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpProperties)

Paramètres

lpszPictureName

Nom de la vue (fichier PDL) contenant les variables utilisées pour l'entrée dans le réseau.

Etant donné que la désignation "lpszPictureName" est utilisée pour la vue actuelle, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un objet se trouvant dans une autre vue.

lpszObjectName

Nom de l'objet dans la vue lié aux variables utilisées pour l'entrée dans le réseau.

Etant donné que la dénomination "lpszObjectName" est utilisée pour l'objet actuel, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un autre objet.

lpProperties

Nom de l'attribut lié à la variable utilisée pour l'entrée dans le réseau.

Si vous indiquez plusieurs attributs, vous devez les séparer par des points-virgules (";").

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

GetKopFupAwlLevel

Fonction

Effectue un contrôle d'autorisation pour l'utilisateur actif à l'aide du numéro de fonction User Administrator indiqué sous dwLevel et effectue l'entrée dans l'éditeur STEP7 "KFA".

Lors de l'exécution de la fonction, trois tâches sont remplies :

- Détermination des données nécessaires à l'entrée dans le réseau au départ de WinCC.
- Réalisation du contrôle d'autorisation pour l'utilisateur actif dans WinCC.
- Transmission des données à STEP7 et localisation du point d'emploi de l'opérande dans un programme STEP7 à l'aide de l'AUTAPI.

Remarque

Selon le résultat du contrôle d'autorisation dans WinCC, l'utilisateur possède, dans STEP7, soit uniquement le droit de lecture soit l'autorisation à la modification des données S7.

Syntaxe

BOOL GetKopFupAwlLevel(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,LPCTSTR lpProperties, DWORD dwLevel)

Paramètres

lpszPictureName

Nom de la vue (fichier PDL) contenant les variables utilisées pour l'entrée dans le réseau.

Etant donné que la désignation "lpszPictureName" est utilisée pour la vue actuelle, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un objet se trouvant dans une autre vue.

lpszObjectName

Nom de l'objet dans la vue lié aux variables utilisées pour l'entrée dans le réseau.

Etant donné que la dénomination "lpszObjectName" est utilisée pour l'objet actuel, une entrée n'est ici nécessaire que dans les cas où l'on désire passer à la variable d'un autre objet.

lpProperties

Nom de l'attribut lié à la variable utilisée pour l'entrée dans le réseau.

Si vous indiquez plusieurs attributs, vous devez les séparer par des points-virgules (";").

dwLevel

Numéro de niveau pour l'autorisation d'écriture STEP7.

Ce numéro peut être déterminé dans User Administrator.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

OnDeactivateExecute

Fonction

Cette fonction est appelée lors de l'arrêt de WinCC Runtime.

Comme il s'agit d'une fonction standard, vous pouvez insérer des instructions qui seront exécutées.

Remarque

En donnant des instructions, il faut tenir compte du fait que Runtime est sur le point de se fermer et donc ne pas toutes les fonctionnalités sont disponibles.

Veuillez noter que des fonctions standard modifiées sont écrasées par une installation WinCC et que les modifications sont ainsi perdues

Syntaxe

void OnDeactivateExecute()

OnErrorExecute

Fonction

OnErrorExecute est appelé par le système si une erreur est survenue lors de l'exécution d'une action ou d'une fonction.

Vous disposez ainsi de la possibilité de déterminer la cause exacte de l'erreur.

La fonction est appelée par le système et ne demande donc pas un appel exprès.

Comme cette fonction est une fonction standard, la façon et la forme de la sortie peuvent être influencées si besoin est.

Remarque

Veuillez noter que des fonctions standard modifiées sont écrasées lors d'une réinstallation et que les modifications sont ainsi perdues.

Syntaxe

void OnErrorExecute(CCAPErrorExecute ErrorExecute)

Paramètres

ErrorExecute

Structure qui donne des informations sur l'erreur survenue

Informations de diagnostic

Ces informations sont affichées dans une fenêtre de diagnostic Global Script.

SystemTime	Le moment (UTC) où l'erreur est survenue
dwErrorCode1	Vous trouverez les codes d'erreur et leurs signification dans la définition de structure
dwErrorCode2	Vous trouverez les codes d'erreur et leurs signification dans la définition de structure
szErrorText	Description en texte de la cause d'erreur
bCycle	Type de cycle
szApplicationName	Application qui déclenche l'erreur
szFunctionName	ID de fonction
szTagName	Nom de variable
dwCycle	Type de cycle
szErrorTextTagName	Description en texte de l'état de variable
status	Etat de la variable
lpszPictureName	La vue dans laquelle l'erreur est survenue
lpszObjectName	L'objet dans lequel l'erreur est survenue
lpszPropertyName	La propriété d'objet dans laquelle l'erreur est survenue
dwParamSize	n'est utilisé que de manière interne
szErrortext	Description en texte de la cause d'erreur, fournie par la structure d'erreur "pError".

Voir aussi

Définition de structure CCAPErrorExecute

OnTime

Fonction

OnTime est uniquement appelé par le système. La fonction sort le temps d'exécution de toutes les actions ou détecte les actions qui sont exécutées pendant une période plus longue que la durée prédéfinie. La mesure du temps peut être activée et désactivée via l'APDIAG.

Comme cette fonction est une fonction standard, la façon de la sortie peut être influencée par une modification du code de fonction.

Remarque

Veuillez noter que des fonctions standard modifiées sont écrasées par une installation WinCC et que les modifications sont ainsi perdues

Syntaxe

void OnTime(CCAPTime time)

Paramètres

time

Structure de résultat

STRUKTUREN_TABELLEN_CCAPTIME

2.15.2.7 Windows

Windows - brève description

Le groupe Windows contient la fonction ProgramExecute.

Cette fonction vous permet d'exécuter n'importe quel programme.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

ProgramExecute

Fonction

Démarre le programme avec le nom indiqué.

Syntaxe

unsigned int ProgramExecute(char* Program_Name)

Paramètres

Program_Name

Pointeur sur le nom de programme

Valeur de retour

Si la valeur en retour est supérieure à 31, la fonction a été exécutée sans erreur.

Dans le cas d'une erreur, la valeur en retour contient un des codes d'erreur suivants :

0	out of memory
2	Le fichier indiqué n'a pas pu être trouvé.
3	Le chemin indiqué n'a pas pu être trouvé.
11	Le programme n'a pas pu être démarré.

Voir aussi

ProgramExecute

2.15.3 Fonctions internes

2.15.3.1 Fonctions internes - Brève description

Vous utiliser des fonctions internes à dynamiser des objets graphiques et des archives ainsi que dans des fonctions de projet, des fonctions standard et des actions Global Script.

Les fonctions internes sont connues par tous les projets.

Vous ne pouvez ni créer de nouvelles fonctions internes, ni modifier des fonctions internes existantes.

Les fonctions internes sont divisées dans les groupes suivants :

allocate

Fonctions pour la réservation et le partage de la mémoire de travail

c_bib

Fonctions de la bibliothèque standard C

graphics

Fonctions pour lire et régler les propriétés d'objets graphiques

tag

Fonctions pour écrire et lire des variables

wincc

Fonctions pour changer la langue, désactiver Runtime et quitter WinCC

2.15.3.2 allocate

SysFree

Fonction

Débloque la zone de mémoire réservée avec la fonction SysMalloc.

Syntaxe

void SysFree(void* lpFree);

Paramètres

lpFree

Pointeur sur la zone de mémoire réservée avec la fonction SysMalloc

SysMalloc

Fonction

Réserve de la mémoire pour une action. La zone de mémoire est affectée à l'action. Quand l'action est exécutée et le résultat envoyé, la mémoire est débloquée par le système.

La fonction SysFree permet de débloquer la mémoire à l'avance.

Syntaxe

void* SysMalloc(unsigned long int size);

Paramètres

size

Taille de la zone de mémoire en octets.

2.15.3.3 c_bib

c_bib - brève description

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ctype

isalnum

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio

- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

isalpha

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

isdigit

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

isgraph

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

islower

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

isprint

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ispunct

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

isspace

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

isupper

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

isxdigit

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

tolower

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

toupper

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

math

acos

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib

- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

asin

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

atan

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

atan2

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ceil

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

cos

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

cosh

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

exp

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fabs

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

floor

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fmod

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

frexp

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ldexp

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

log

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

log10

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

modf

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

pow

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

sin

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

sinh

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

sqrt

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

tan

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

tanh

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

memory

memchr

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

memcmp

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

memcpy

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

memmove

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

memset

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

stdio

char_io

fgetc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fgets

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio

- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fputc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fputs

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

getc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

putc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ungetc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

Directio

fread

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fwrite

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

Error

clearerr

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib

- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

feof

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ferror

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

File

fclose

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fflush

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib

- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fopen

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

freopen

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

remove

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

rename

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

setbuf

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

setvbuf

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

tmpfile

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

tmpnam

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

File_pos

fgetpos

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fseek

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

fsetpos

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ftell

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

rewind

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

Output

vfprintf

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

vsprintf

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

stdlib

abs

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib

- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

atof

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

atoi

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

atol

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

bsearch

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

calloc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

div

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

free

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

getenv

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

labs

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ldiv

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

malloc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

qsort

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

rand

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

realloc

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

srand

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strtod

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strtol

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strtoul

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

string (chaîne)

strcat

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

---

strchr

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib

- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strcmp

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strcpy

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strcspn

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strerror

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strlen

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strncat

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strncmp

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strncpy

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strpbrk

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strrchr

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strspn

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strstr

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strtok

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

time

asctime

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

clock

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

ctime

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

difftime

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

gmtime

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

localtime

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

mktime

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

strftime

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

time

Le groupe de fonction c_bib contient des fonction C de la bibliothèque C et est divisé en :

- ctype
- math
- memory
- stdio
- stdlib
- string (chaîne)
- time

stdio est encore une fois divisé en :

- char_io
- directio
- error
- file
- file_pos
- output

Pour la description de ces fonctions, veuillez vous référer aux ouvrages spécialisés.

Remarque

La fonction localtime présente le comportement suivant lors de la sortie de la date :

Le comptage des mois commence par 0.

Les ans sont comptés à partir de l'an 1900 et en commençant par 0.

Les fonctions de la bibliothèque C printf(), sprintf(), fprintf() ne peuvent traiter que 360 caractères dans WinCC.

2.15.3.4 graphics

Graphics - brève description

Les fonctions du groupe Graphics vous permettent de modifier ou d'interroger des propriétés graphiques d'objets WinCC.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Remarque

Si l'appel de la fonction se réfère à l'objet d'image, le paramètre lpszObjectName = ZÉRO doit être défini.

get

axes

GetAlignment

Fonction

Indique, pour les objets de barre, si le texte se trouve à droite ou à gauche de la barre.

Syntaxe

BOOL GetAlignment(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Texte à droite de la barre

FALSE

Texte à gauche de la barre

Voir aussi

Exemple GetScaling

Exemple GetScaling (Page 1533)

GetAxisSection

Fonction

Fournit, pour les objets de barre, la différence de valeur entre deux textes des axes voisins.

Syntaxe

double GetAxisSection(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Différence de valeur entre deux textes des axes voisins

GetExponent

Fonction

Indique, pour les objets de barre, si les textes des axes correspondent à la notation exponentielle ou décimale.

Syntaxe

BOOL GetExponent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Textes des axes en notation exponentielle

FALSE

Textes des axes en notation décimale

Voir aussi

Exemple GetScaling

Exemple GetScaling (Page 1533)

GetLeftComma

Fonction

Indique, pour les objets de barre, le nombre des chiffres avant la virgule dans les textes des axes.

Syntaxe

long int GetLeftComma(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Nombre des chiffres avant la virgule dans les textes des axes

GetLongStrokesBold

Fonction

Indique, pour les objets de barre, si les grandes divisions sur l'échelle sont représentées en gras ou normalement.

Syntaxe

BOOL GetLongStrokesBold(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Grandes divisions sur l'échelle de la barre en gras

FALSE

Grandes divisions sur l'échelle de la barre normales

Voir aussi

Exemple GetScaling

Exemple GetScaling (Page 1533)

GetLongStrokesOnly

Fonction

Indique, pour les objets de barre, si également des subdivisions sont utilisées sur l'échelle.

Syntaxe

BOOL GetLongStrokesOnly(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Sur l'échelle de la barre ne sont utilisées que des grandes divisions.

FALSE

Sur l'échelle de la barre sont utilisées des grandes divisions ainsi que des subdivisions.

Voir aussi

Exemple GetScaling

Exemple GetScaling (Page 1533)

GetLongStrokesSize

Fonction

Indique, pour les objets de barre, la longueur des grandes divisions.

Syntaxe

```
long int GetLongStrokesSize(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Longueur des grandes divisions sous forme de valeur en chiffre.

GetLongStrokesTextEach

Fonction

Indique, pour les objets de barre, la grande division de quelle position comporte un texte.

Syntaxe

long int GetLongStrokesTextEach(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Textes des grandes divisions sous forme de valeur en chiffre.

Exemple:

Valeur en retour = 1 -> Chaque grande division comporte un texte.

Valeur en retour = 2 -> Une grande division sur deux comporte un texte.

etc.

GetRightComma

Fonction

Indique, pour les objets de barre, le nombre des chiffres après la virgule dans les textes des axes.

Syntaxe

long int GetRightComma(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Nombre des chiffres après la virgule dans les textes des axes

GetScaleTicks

Fonction

Fournit, pour les objets de barre, la division d'échelle sous forme de nombre de sections d'échelle. La section d'échelle étant la zone délimitée par deux grandes divisions.

Syntaxe

long int GetScaleTicks(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Division d'échelle sous forme de nombre des sections d'échelle.

Remarque

Le nombre des sections d'échelle est indiqué par 0 quand l'objet de barre calcule lui-même une division d'échelle appropriée.

GetScaling

Fonction

Renvoie, pour les objets de barre, si l'échelle est activée ou désactivée.

Syntaxe

BOOL GetScaling(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Représentation avec échelle

FALSE

Représentation sans échelle

Voir aussi

Exemple GetScaling

Exemple GetScaling (Page 1533)

GetScalingType

Fonction

Indique, pour les objets de barre, le type de la mise à l'échelle de la barre.

Syntaxe

long int GetScalingType(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Type de la mise à l'échelle sous forme de valeur en chiffre.

Voir aussi

Mise à l'échelle du bargraphe

Mise à l'échelle du bargraphe (Page 1583)

color

Color- brève description

Les fonctions du groupe Color vous permettent de modifier ou d'interroger de différentes propriétés de couleur sur des objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetBackColor

Fonction

Indique la couleur d'arrière-plan de l'objet sous forme de valeur en chiffre.

Syntaxe

long int GetBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur d'arrière-plan de l'objet sous forme de valeur en chiffre

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBackColor2

Fonction

Indique, pour les objets de barre, la couleur de la barre sous forme de valeur en chiffre.

Syntaxe

long int GetBackColor2(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de la barre sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBackColor3

Fonction

Indique, pour les objets de barre, la couleur d'arrière-plan de la barre sous forme de valeur en chiffre.

Syntaxe

long int GetBackColor3(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur d'arrière-plan de la barre sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBackColorBottom

Fonction

Indique, pour les objets slider, la couleur d'arrière-plan en bas à droite.

Syntaxe

long int GetBackColorBottom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les objets slider, valeur en chiffre de la couleur d'arrière-plan en bas à droite.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBackColorTop

Fonction

Indique, pour les objets slider, la couleur d'arrière-plan en haut à gauche.

Syntaxe

long int GetBackColorTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les objets slider, valeur en chiffre de la couleur d'arrière-plan en haut à gauche.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBorderBackColor

Fonction

Indique la couleur d'arrière-plan des lignes ou de la bordure.

Syntaxe

long int GetBorderBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur d'arrière-plan des lignes ou de la bordure sous forme de valeur en chiffre.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBorderColor

Fonction

Indique la couleur des lignes ou de la bordure sous forme de valeur en chiffre.

Syntaxe

long int GetBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur des lignes ou de la bordure sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBorderColorBottom

Fonction

Indique la couleur de bordure 3D en bas.

Syntaxe

long int GetBorderColorBottom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de bordure 3D sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetBorderColorTop

Fonction

Indique la couleur de bordure 3D en haut.

Syntaxe

long int GetBorderColorTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de bordure 3D sous forme de valeur en chiffre en haut

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetButtonColor

Fonction

Indique, pour les objets slider, la couleur de bouton.

Syntaxe

long int GetButtonColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les objets slider, couleur de bouton sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetColorBottom

Fonction

Indique, pour les objets slider, la couleur de la limite inférieure.

Syntaxe

long int GetColorBottom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les objets slider, couleur de la limite inférieure sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetColorTop

Fonction

Indique, pour les objets slider, la couleur de la limite supérieure.

Syntaxe

long int GetColorTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les objets slider, couleur de la limite supérieure sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetFillColor

Fonction

Indique la couleur du motif de remplissage.

Syntaxe

long int GetFillColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur du motif de rempl. sous forme de valeur en chiffre

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetForeColor

Fonction

Indique la couleur de police.

Syntaxe

long int GetForeColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de police sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetGridColor

Fonction

Indique la couleur de quadrillage de Graphics Designer.

Syntaxe

long int GetGridColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de quadrillage de Graphics Designer sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetItemBorderBackColor

Fonction

Indique, pour l'objet "liste de textes", la couleur d'arrière-plan de la ligne de séparation.

Syntaxe

long int GetItemBorderBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour l'objet "liste de textes", couleur d'arrière-plan de la ligne de séparation sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetItemBorderColor

Fonction

Indique, pour l'objet "liste de textes", la couleur de la ligne de sép.

Syntaxe

long int GetItemBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour l'objet "liste de textes", couleur de la ligne de sép. sous forme de valeur en chiffre

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetScaleColor

Fonction

Indique, pour les objets de barre, la couleur d'échelle.

Syntaxe

long int GetScaleColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les obj. de barre, couleur d'échelle sous forme de valeur en chiffre.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetSelBGColor

Fonction

Indique, pour l'objet "liste de textes", la couleur d'arrière-plan pour l'entrée sélectionnée.

Syntaxe

long int GetSelBGColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Valeur en chiffre de la couleur d'arrière-plan pour l'entrée sélectionnée.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetSelTextColor

Fonction

Indique, pour l'objet "liste de textes", la couleur de police pour l'entrée sélectionnée.

Syntaxe

long int GetSelTextColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Valeur en chiffre de la couleur de police pour l'entrée sélectionnée.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetTrendColor

Fonction

Indique, pour les objets de barre, la couleur de tendance.

Syntaxe

long int GetTrendColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les objets de barre, valeur en chiffre de la couleur de tendance

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

GetUnselBGColor

Fonction

Indique, pour l'objet "liste de textes", la couleur d'arrière-plan pour les entrées non sélectionnées.

Syntaxe

long int GetUnselBGColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Valeur en chiffre de la couleur d'arrière-plan pour les entrées non sélectionnées.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Tableau de couleurs (Page 1585)

Exemple GetBackColor (Page 1516)

GetUnselTextColor

Fonction

Indique, pour l'objet "liste de textes", la couleur de police pour les entrées non sélectionnées.

Syntaxe

long int GetUnselTextColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Valeur en chiffre de la couleur de police pour les entrées non sélectionnées.

Voir aussi

Exemple GetBackColor

Tableau de couleurs

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

fill

Fill - brève description

Les fonctions du groupe Fill pilotent le remplissage dynamique d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetFilling

Fonction

Indique si le remplissage dynamique avec la couleur d'arrière-plan est activé.

Syntaxe

BOOL GetFilling(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Le remplissage dynamique avec la couleur d'arrière-plan est activé.

FALSE

Le remplissage dynamique avec la couleur d'arrière-plan n'est pas activé.

Voir aussi

Exemple GetFilling

Exemple GetFilling (Page 1517)

GetFillingIndex

Fonction

Indique le taux de remplissage actuel.

Syntaxe

long int GetFillingIndex(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Taux de remplissage actuel sous forme de valeur en chiffre (0 – 100)

Voir aussi

Exemple GetFillingIndex

Exemple GetFillingIndex (Page 1518)

flash

Flash - brève description

Les fonctions du groupe Flash permettent de modifier ou d'interroger de différentes propriétés concernant le clignotement.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetBackFlashColorOff

Fonction

Indique la couleur clignotante d'arrière-plan pour l'état désactivé.

Syntaxe

long int GetBackFlashColorOff(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur clignotante d'arrière-plan pour l'état désactivé sous forme de valeur en chiffre

Voir aussi

Tableau de couleurs (Page 1585)

Exemple GetFlashBackColorOn

Exemple GetFlashBackColorOn (Page 1519)

Tableau de couleurs

GetBackFlashColorOn

Fonction

Indique la couleur clignotante d'arrière-plan pour l'état activé.

Syntaxe

long int GetBackFlashColorOn(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur clignotante d'arrière-plan pour l'état activé sous forme de valeur en chiffre

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)

Tableau de couleurs (Page 1585)

Tableau de couleurs

Exemple GetFlashBackColorOn

GetBorderFlashColorOff

Fonction

Indique la couleur clignotante de la bordure et de ligne pour l'état désactivé.

Syntaxe

long int GetBorderFlashColorOff(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de bordure ou de ligne clignotante pour l'état désactivé sous forme de valeur en chiffre

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)

Tableau de couleurs (Page 1585)

Tableau de couleurs

Exemple GetFlashBackColorOn

GetBorderFlashColorOn

Fonction

Indique la couleur clignotante de la bordure et de la ligne pour l'état activé.

Syntaxe

long int GetBorderFlashColorOn(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de bordure ou de ligne clignotante pour l'état activé sous forme de valeur en chiffre

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)
Tableau de couleurs (Page 1585)
Tableau de couleurs
Exemple GetFlashBackColorOn

GetFlashBackColor

Fonction

Indique si le clignotement dans l'arrière-plan est activé.

Syntaxe

BOOL GetFlashBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Le clignotement de l'arrière-plan est activé.

FALSE

Le clignotement de l'arrière-plan n'est pas activé.

Voir aussi

Exemple GetFlashBackColor (Page 1519)

Exemple GetFlashBackColor

GetFlashBorderColor

Fonction

Indique si le clignotement de la bordure ou de ligne dans l'arrière-plan est activé.

Syntaxe

BOOL GetFlashBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Le clignotement de la bordure ou du trait est activé.

FALSE

Le clignotement de la bordure ou de la ligne n'est pas activé.

Voir aussi

Exemple GetFlashBackColor (Page 1519)

Exemple GetFlashBackColor

GetFlashForeColor

Fonction

Indique si le clignotement de la police est activé.

Syntaxe

BOOL GetFlashForeColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Le clignotement de la police est activé.

FALSE

Le clignotement de la police n'est pas activé.

Voir aussi

Exemple GetFlashBackColor (Page 1519)

Exemple GetFlashBackColor

GetFlashRateBackColor

Fonction

Indique la fréquence du clignotement de l'arrière-plan.

Syntaxe

long int GetFlashRateBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Fréquence de clignotement de l'arrière-plan

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)

Fréquences de clignotement (Page 1583)

Exemple GetFlashBackColorOn

Fréquences de clignotement

GetFlashRateBorderColor

Fonction

Indique la fréquence du clignotement de la ligne ou de la bordure.

Syntaxe

long int GetFlashRateBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Fréquence du clignotement de la ligne ou de la bordure.

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)

Fréquences de clignotement (Page 1583)

Exemple GetFlashBackColorOn

Fréquences de clignotement

GetFlashRateForeColor

Fonction

Indique la fréquence du clignotement de la police.

Syntaxe

long int GetBorderFlashColorOff(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Fréquence de clignotement de la police

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)

Fréquences de clignotement (Page 1583)

Exemple GetFlashBackColorOn

Fréquences de clignotement

GetForeFlashColorOff

Fonction

Indique la couleur clignotante de la police pour l'état désactivé.

Syntaxe

long int GetForeFlashColorOff(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur clignotante de la police pour l'état désactivé sous forme de valeur en chiffre

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)
Tableau de couleurs (Page 1585)
Exemple GetFlashBackColorOn
Fréquences de clignotement

GetForeFlashColorOn

Fonction

Indique la couleur clignotante de la police pour l'état activé.

Syntaxe

long int GetForeFlashColorOn(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur clignotante de la police pour l'état activé sous forme de valeur en chiffre

Voir aussi

Exemple GetFlashBackColorOn (Page 1519)

Tableau de couleurs (Page 1585)

Exemple GetFlashBackColorOn

Tableau de couleurs

focus

Focus - abrégé descriptif

Les fonctions du groupe Focus permettent de placer le focus sur un objet ou bien interroger quel objet possède le focus.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Get_Focus

Fonction

Indique le nom de 'objet qui possède focus ou l'a possédé avant.

Syntaxe

char *Get_Focus();

Valeur en retour

Nom de l'objet qui possède le focus ou l'a possédé avant.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Exemple GetFocus (Page 1520)

Exemple GetFocus

font

Font - Brève description

Les fonctions du groupe Font permettent de modifier ou d'interroger les propriétés concernant le texte.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetAlignmentLeft

Fonction

Indique le sens du texte horizontal (à gauche, centré, à droite).

Syntaxe

long int GetAlignmentLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Sens du texte horizontal sous forme de valeur en chiffre

Voir aussi

Sens du texte (Page 1591)

Exemple GetFontSize (Page 1522)

Exemple GetFontSize

Alignement du texte

GetAlignmentTop

Fonction

Indique le sens du texte vertical (en haut, centré, en bas).

Syntaxe

long int GetAlignmentTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Sens du texte vertical sous forme de valeur en chiffre

Voir aussi

Exemple GetFontSize (Page 1522)
Sens du texte (Page 1591)
Exemple GetFontSize
Alignement du texte

GetFontBold

Fonction

Indique si le style de police est représenté "en gras".

Syntaxe

BOOL GetFontBold(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Style de police "en gras activé.

FALSE

Style de police "en gras" désactivé.

Voir aussi

Exemple GetFontBold (Page 1521)

Exemple GetFontBold

GetFontItalic

Fonction

Indique si le style de police est "en italique".

Syntaxe

BOOL GetFontItalic(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Style de police "en italique" activé.

FALSE

Style de police "en italique" désactivé

Voir aussi

Exemple GetFontBold (Page 1521)

Exemple GetFontBold

GetFontName

Fonction

Indique la couleur de police activé.

Syntaxe

char* GetFontName(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pointeur sur le nom de la police actuellement réglée.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Exemple GetText (Page 1550)

Exemple GetText

GetFontSize

Fonction

Indique la taille de police.

Syntaxe

long int GetFontSize(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Taille de police actuelle

Voir aussi

Exemple GetFontSize (Page 1522)

Exemple GetFontSize

GetFontUnderline

Fonction

Indique si le style de police est "souligné".

Syntaxe

BOOL GetFontUnderline(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Style de police "souligné" activé

FALSE

Style de police "souligné" désactivé

Voir aussi

Exemple GetFontBold (Page 1521)

Exemple GetFontBold

GetOrientation

Fonction

Indique le sens du texte (vertical/horizontal).

Syntaxe

BOOL GetOrientation(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Sens du texte "vertical"

FALSE

Sens du texte "horizontal"

Voir aussi

Exemple GetFontBold (Page 1521)

Exemple GetFontBold

GetText

Fonction

Indique la valeur de la propriété "Texte" pour des objets comme texte statique, case à cocher et case d'option.

Syntaxe

char* GetText(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pointeur sur un texte.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Remarque

Pour les cases à cocher et les cases d'option, il faut déterminer l'élément à définir via la fonction "SetIndex" avant l'appel de cette fonction.

Voir aussi

Exemple GetText (Page 1550)

general

GetLayer

Fonction

Indique le niveau de vue dans lequel se trouve l'objet.

Syntaxe

long int GetLayer(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Niveau de vue dans lequel se trouve l'objet

geometry

Geometry - brève description

Les fonctions du groupe Geometry vous permettent de modifier ou d'interroger la taille, la position et d'autres propriétés géométriques d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetActualPointLeft

Fonction

Indique la valeur X du point actuel pour des polygones ou des traits polygonaux.

Syntaxe

long int GetActualPointLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur X du point actuel d'un polygone ou d'un trait polygonal

Remarque

La fonction SetIndex permet de régler le point actuel d'un polygone.

Voir aussi

Exemple GetLeft (Page 1524)

Exemple GetLeft

GetActualPointTop

Fonction

Indique la valeur Y du point actuel pour des polygones ou des traits polygonaux.

Syntaxe

long int GetActualPointTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur Y du point actuel d'un polygone ou d'un trait polygonal

Remarque

La fonction SetIndex permet de régler le point actuel d'un polygone.

Voir aussi

Exemple GetTop (Page 1551)

Exemple GetTop

GetBoxCount

Fonction

Indique le nombre de champs pour des cases à cocher et des cases d'option.

Syntaxe

long int GetBoxCount(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Nombre de champs dans une case à cocher ou une case d'option

GetDirection

Fonction

Indique, pour les objets de barre, l'orientation du bargraphe.

Syntaxe

long int GetDirection(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pour les objets de barre, orientation du bargraphe sous forme de valeur en chiffre

Voir aussi

Orientation du bargraphe (Page 1582)

Orientation du bargraphe

GetEndAngle

Fonction

Indique l'angle final pour des segments de cercle et d'ellipse ainsi que pour des arcs de cercle et d'ellipse.

Syntaxe

long int GetEndAngle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Angle final pour des segments de cercle et d'ellipse ainsi que pour des arcs de cercle et d'ellipse.

GetGrid

Fonction

Indique si le quadrillage dans la surface de dessin du Graphics Desginer est activé.

Syntaxe

BOOL GetGrid(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Le quadrillage dans Graphics Designer est activé.

FALSE

Le quadrillage dans Graphics Designer est désactivé.

GetGridHeight

Fonction

Indique la hauteur du quadrillage dans la surface de dessin du Graphics Desginer.

Syntaxe

long int GetGridHeight(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Hauteur du quadrillage dans Graphics Designer

GetGridWidth

Fonction

Indique la largeur du quadrillage dans la surface de dessin du Graphics Desginer.

Syntaxe

long int GetGridWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Largeur du quadrillage dans Graphics Designer

GetHeight

Fonction

Indique la hauteur du rectangle circonscrit d'un objet.

Syntaxe

long int GetHeight(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Hauteur du rectangle circonscrit d'un objet

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple GetHeight (Page 1522)

GetLeft

Fonction

Indique la position X de l'angle gauche supérieur du rectangle circonscrit d'un objet.

Syntaxe

long int GetLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur X actuelle de l'angle gauche supérieur du rectangle circonscrit d'un objet

Voir aussi

Exemple GetLeft (Page 1524)

Exemple GetLeft

GetPointCount

Fonction

Indique le nombre de points d'angle d'un polygone ou d'un trait polygonal.

Syntaxe

long int GetPointCount(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Nombre de points d'angle d'un polygone ou d'un trait polygonal

GetRadius

Fonction

Indique le rayon d'un cercle, d'un segment de cercle ou d'un arc de cercle.

Syntaxe

long int GetRadius(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Rayon d'un cercle, d'un segment de cercle ou d'un arc de cercle

Voir aussi

Exemple GetHeight (Page 1522)

Exemple GetHeight

GetRadiusHeight

Fonction

Indique le rayon d'une ellipse, d'un segment d'ellipse ou d'un arc d'ellipse dans le sens vertical.

Syntaxe

long int GetRadiusHeight(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Rayon d'une ellipse, d'un segment d'ellipse ou d'un arc d'ellipse dans le sens vertical

Voir aussi

Exemple GetHeight (Page 1522)

Exemple GetHeight

GetRadiusWidth

Fonction

Indique le rayon d'une ellipse, d'un segment d'ellipse ou d'un arc d'ellipse dans le sens horizontal.

Syntaxe

long int GetRadiusWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Rayon d'une ellipse, d'un segment d'ellipse ou d'un arc d'ellipse dans le sens horizontal

Voir aussi

Exemple GetHeight (Page 1522)

Exemple GetHeight

GetReferenceRotationLeft

Fonction

Indique la valeur X de la référence de rotation pour une ligne, un polygone et un trait polygonal (Point de rotation autour duquel l'objet peut être tourné).

Syntaxe

long int GetReferenceRotationLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur X de la référence de rotation pour une ligne, un polygone et un trait polygonal

Voir aussi

Exemple GetLeft (Page 1524)

Exemple GetLeft

GetReferenceRotationTop

Fonction

Indique la valeur Y de la référence de rotation pour une ligne, un polygone et un trait polygonal (Point de rotation autour duquel l'objet peut être tourné).

Syntaxe

long int GetReferenceRotationTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur Y de la référence de rotation pour une ligne, un polygone et un trait polygonal

Voir aussi

Exemple GetTop (Page 1551)

Exemple GetTop

GetRotationAngle

Fonction

Indique l'angle de rotation autour du point de rotation pour une ligne, un polygone ou un trait polygonal.

Syntaxe

long int GetRotationAngle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Angle de rotation autour du point de rotation

Voir aussi

Exemple GetHeight (Page 1522)

Exemple GetHeight

GetRoundCornerHeight

Fonction

Indique le rayon vertical de l'angle d'un rectangle arrondi.

Syntaxe

long int GetRoundCornerHeight(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Rayon vertical de l'angle d'un rectangle arrondi

Voir aussi

Exemple GetHeight (Page 1522)

Exemple GetHeight

GetRoundCornerWidth

Fonction

Indique le rayon horizontal de l'angle d'un rectangle arrondi.

Syntaxe

long int GetRoundCornerWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Rayon horizontal de l'angle d'un rectangle arrondi

Voir aussi

Exemple GetWidth (Page 1552)

Exemple GetWidth

GetStartAngle

Fonction

Indique l'angle de référence pour des segments de cercle et d'ellipse ainsi que pour des arcs de cercle et d'ellipse.

Syntaxe

long int GetStartAngle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Angle de référence pour des segments de cercle et d'ellipse ainsi que pour des arcs de cercle et d'ellipse

Voir aussi

Exemple GetHeight (Page 1522)

Exemple GetHeight

GetTop

Fonction

Indique la position Y de l'angle gauche supérieur du rectangle circonscrit d'un objet.

Syntaxe

long int GetTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur Y actuelle de l'angle gauche supérieur du rectangle circonscrit d'un objet

Voir aussi

Exemple GetTop (Page 1551)

Exemple GetTop

GetWidth

Fonction

Indique la largeur du rectangle circonscrit d'un objet.

Syntaxe

long int GetWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Largeur du rectangle circonscrit d'un objet

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple GetWidth (Page 1552)

Exemple GetWidth

GetZeroPoint

Fonction

Indique l'origine pour les objets de barre.

Syntaxe

long int GetZeroPoint(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Origine pour objets de barre

Voir aussi

Exemple GetHeight (Page 1522)

Exemple GetHeight

i_o

i_o - brève description

Les fonctions du groupe i_o permettent de modifier ou d'interroger de différentes propriétés concernant les valeurs de sortie et d'entrée.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetAssignments

Fonction

Pour des listes, affectation de texte à la plage de valeurs.

Syntaxe

char* GetAssignments(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

L'affectation de texte à la plage de valeurs dépend du type de liste.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
```

```
}
```

Voir aussi

Types de liste (Page 1589)

Types de liste

GetAssumeOnExit

Fonction

Indique, pour les champs E/S, si les valeurs entrées seront validées au moment de quitter le champ.

Syntaxe

BOOL GetAssumeOnExit(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Valeurs sont validées au moment de quitter le champ.

FALSE

Valeurs ne sont pas validées au moment de quitter le champ.

GetAssumeOnFull

Fonction

Indique, pour les champs E/S, si les valeurs entrées seront validées lors de l'entrée complète.

Syntaxe

BOOL GetAssumeOnFull(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Valeurs sont validées lors de l'entrée complète.

FALSE

Valeurs ne sont pas validées lors de l'entrée complète.

GetBitNumber

Fonction

Indique, dans le cas du type de liste "bit", le bit pertinent dans la valeur de sortie.

Syntaxe

long int GetBitNumber(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Indication, dans le cas du type de liste "bit", du bit pertinent dans la valeur de sortie

Voir aussi

Exemple GetHiddenInput (Page 1523)

Types de liste (Page 1589)

Types de liste

Exemple GetHiddenInput

GetClearOnError

Fonction

Indique, pour les champs E/S, si la suppression du contenu en cas d'entrée erronée est activée.

Syntaxe

BOOL GetClearOnError(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Suppression du contenu en cas d'entrée erronée est activée

FALSE

Suppression du contenu en cas d'entrée erronée n'est pas activée

GetClearOnNew

Fonction

Indique, pour les champs E/S, si la suppression du contenu en cas de nouvelle entrée est activée.

Syntaxe

BOOL GetClearOnNew(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Suppression du contenu en cas de nouvelle entrée est activée.

FALSE

Suppression du contenu en cas de nouvelle entrée n'est pas activée.

GetDataFormat

Fonction

Indique, pour les champs E/S, le type de données du contenu du champ.

Syntaxe

long int GetDataFormat(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Type de données du contenu du champ sous forme de valeur en chiffre

Voir aussi

Exemple GetHiddenInput (Page 1523)
Champ E/S, Type de données du contenu du champ (Page 1585)
Exemple GetHiddenInput
Champ d'E/S, type de données du contenu du champ

GetHiddenInput

Fonction

Indique, pour les champs E/S, si l'entrée cachée est activée.

Syntaxe

BOOL GetHiddenInput(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Entrée cachée est activée

FALSE

Entrée cachée n'est pas activée

Voir aussi

Exemple GetHiddenInput (Page 1523)

Exemple GetHiddenInput

GetInputValueChar

Fonction

Indique, pour les champs E/S, la valeur d'entrée dans le type de données "char".

Syntaxe

char* GetInputValueChar(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pointeur sur la valeur d'entrée dans le type de données "char".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

GetInputValueDouble

Fonction

Indique, pour les champs E/S, la valeur d'entrée dans le type de données "double".

Syntaxe

double GetInputValueDouble(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur d'entrée dans le type de données "double".

GetListType

Fonction

Indique, pour l'objet "liste de textes", le type de liste.

Syntaxe

long int GetListType(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Type de liste pour l'objet "liste de textes"

Voir aussi

Exemple GetHiddenInput (Page 1523)

Types de liste (Page 1589)

Exemple GetHiddenInput

Types de liste

GetNumberLines

Fonction

Indique, pour l'objet "liste de textes", le nombre des lignes visibles.

Syntaxe

long int GetNumberLines(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Nombre des lignes visibles pour l'objet "liste de textes"

Remarque

Si la quantité des textes configurées est supérieure au nombre des lignes visibles, l'objet "liste de textes" est complété par une barre de défilement verticale.

Voir aussi

Exemple GetHiddenInput (Page 1523)

Exemple GetHiddenInput

GetOutputFormat

Fonction

Indique le format de sortie pour les champs E/S.

Syntaxe

char* GetOutputFormat(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pointeur sur le format de sortie.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Champ E/S, Type de données du contenu du champ (Page 1585)

Champ d'E/S, Format de sortie (Page 1583)

Champ d'E/S, type de données du contenu du champ

Champ d'E/S, Format de sortie

GetOutputValueChar

Fonction

Détermine, pour les champs E/S, la valeur de sortie dans le type de données "char". La fonction ne devrait être utilisée que si le contenu du champ E/S est du type de données "string".

Syntaxe

char* GetOutputValueChar(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Pointeur sur la valeur de sortie dans le type de données "char".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

GetOutputValueDouble

Fonction

Détermine, pour les champs E/S, la valeur de sortie dans le type de données "double". La fonction ne devrait être utilisée que si le contenu du champ E/S n'est pas du type de données "string".

Syntaxe

double GetOutputValueDouble(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur de sortie dans le type de données "double".

Voir aussi

Exemple GetOutputValueDouble (Page 1527)

Exemple GetOutputValueDouble

Limites

Limites - brève description

Les fonctions du groupe Limites permettent de modifier ou d'interroger de différentes propriétés concernant les valeurs limites.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetAlarmHigh

Fonction

Indique, pour les objets de barre, le seuil d'alarme supérieur.

Syntaxe

double GetAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil d'alarme supérieur pour objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetAlarmLow

Fonction

Indique, pour les objets de barre, le seuil d'alarme inférieur.

Syntaxe

double GetAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil d'alarme inférieur pour objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetCheckAlarmHigh

Fonction

Indique, pour les objets de barre, si le seuil d'alarme supérieur est surveillé.

Syntaxe

BOOL GetCheckAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'alarme supérieur est surveillé.

FALSE

Pour les objets de barre, le seuil d'alarme supérieur n'est pas surveillé.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckAlarmLow

Fonction

Indique, pour les objets de barre, si le seuil d'alarme inférieur est surveillé.

Syntaxe

BOOL GetCheckAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'alarme inférieur est surveillé.

FALSE

Pour les objets de barre, le seuil d'alarme inférieur n'est pas surveillé.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckLimitHigh4

Fonction

Indique, pour les objets de barre, si la valeur limite supérieure Réserve 4 est surveillée.

Syntaxe

BOOL GetCheckLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre la valeur limite supérieure Réserve 4 est surveillée.

FALSE

Pour les objets de barre, la valeur limite supérieure Réserve 4 n'est pas surveillée.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckLimitHigh5

Fonction

Indique, pour les objets de barre, si la valeur limite supérieure Réserve 5 est surveillée.

Syntaxe

BOOL GetCheckLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre la valeur limite supérieure Réserve 5 est surveillée.

FALSE

Pour les objets de barre, la valeur limite supérieure Réserve 5 n'est pas surveillée.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckLimitLow4

Fonction

Indique, pour les objets de barre, si la valeur limite inférieure Réserve 4 est surveillée.

Syntaxe

BOOL GetCheckLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre la valeur limite inférieure Réserve 4 est surveillée.

FALSE

Pour les objets de barre la valeur limite inférieure Réserve 4 n'est pas surveillée.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckLimitLow5

Fonction

Indique, pour les objets de barre, si la valeur limite inférieure Réserve 5 est surveillée.

Syntaxe

BOOL GetCheckLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre la valeur limite inférieure Réserve 5 est surveillée.

FALSE

Pour les objets de barre la valeur limite inférieure Réserve 5 n'est pas surveillée.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckToleranceHigh

Fonction

Indique, pour les objets de barre, si le seuil de tolérance supérieur est surveillé.

Syntaxe

```
BOOL GetCheckToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil de tolérance supérieur est surveillé.

FALSE

Pour les objets de barre, le seuil de tolérance supérieur n'est pas surveillé.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckToleranceLow

Fonction

Indique, pour les objets de barre, si le seuil de tolérance inférieur est surveillé.

Syntaxe

BOOL GetCheckToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil de tolérance inférieur est surveillé.

FALSE

Pour les objets de barre, le seuil de tolérance inférieur n'est pas surveillé.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckWarningHigh

Fonction

Indique, pour les objets de barre, si le seuil d'avertissement supérieur est surveillé.

Syntaxe

BOOL GetCheckWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'avertissement supérieur est surveillé.

FALSE

Pour les objets de barre, le seuil d'avertissement supérieur n'est pas surveillé.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetCheckWarningLow

Fonction

Indique, pour les objets de barre, si le seuil d'avertissement inférieur est surveillé.

Syntaxe

BOOL GetCheckWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'avertissement inférieur est surveillé.

FALSE

Pour les objets de barre, le seuil d'avertissement inférieur n'est pas surveillé.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetColorAlarmHigh

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil d'alarme supérieur est atteint.

Syntaxe

long int GetColorAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil d'alarme supérieur est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorAlarmLow

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil d'alarme inférieur est atteint.

Syntaxe

long int GetColorAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil d'alarme inférieur est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorLimitHigh4

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil supérieur Réserve 4 est atteint.

Syntaxe

long int GetColorLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil supérieur Réserve 4 est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorLimitHigh5

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil supérieur Réserve 5 est atteint.

Syntaxe

long int GetColorLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil supérieur Réserve 5 est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorLimitLow4

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil inférieur Réserve 4 est atteint.

Syntaxe

long int GetColorLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil inférieur Réserve 4 est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorLimitLow5

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil inférieur Réserve 5 est atteint.

Syntaxe

long int GetColorLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil inférieur Réserve 5 est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorToleranceHigh

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil de tolérance supérieur est atteint.

Syntaxe

long int GetColorToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil de tolérance supérieur est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorToleranceLow

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil de tolérance inférieur est atteint.

Syntaxe

long int GetColorToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil de tolérance inférieur est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorWarningHigh

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil d'avertissement supérieur est atteint.

Syntaxe

long int GetColorWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil d'avertissement supérieur est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetColorWarningLow

Fonction

Indique, pour les objets de barre, la couleur de barre lorsque le seuil d'avertissement inférieur est atteint.

Syntaxe

long int GetColorWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Couleur de barre sous forme de valeur en chiffre lorsque le seuil d'avertissement inférieur est atteint.

Voir aussi

Exemple GetBackColor (Page 1516)

Tableau de couleurs (Page 1585)

Exemple GetBackColor

Tableau de couleurs

GetLimitHigh4

Fonction

Indique, pour les objets de barre, la seuil supérieur pour Réserve 4.

Syntaxe

double GetLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil supérieur pour Réserve 4 pour les objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetLimitHigh5

Fonction

Indique, pour les objets de barre, la seuil supérieur pour Réserve 5.

Syntaxe

double GetLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil supérieur pour Réserve 5 pour les objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetLimitLow4

Fonction

Indique, pour les objets de barre, la seuil inférieur pour Réserve 4.

Syntaxe

double GetLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil inférieur pour Réserve 4 pour les objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetLimitLow5

Fonction

Indique, pour les objets de barre, la seuil inférieur pour Réserve 5.

Syntaxe

double GetLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil inférieur pour Réserve 5 pour les objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetLimitMax

Fonction

Indique la seuil supérieur pour les champs E/S.

Syntaxe

double GetLimitMax(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil supérieur pour les champs E/S

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetLimitMin

Fonction

Indique la seuil inférieur pour les champs E/S.

Syntaxe

double GetLimitMin(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil inférieur pour les champs E/S

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetMarker

Fonction

Indique, pour les objets de barre, si le repère de seuil est affiché.

Syntaxe

BOOL GetMarker(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Le repère de seuil est affichée pour les objets de barre.

FALSE

Le repère de seuil n'est pas affichée pour les objets de barre.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetToleranceHigh

Fonction

Indique, pour les objets de barre, le seuil de tolérance supérieur.

Syntaxe

double GetToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil de tolérance supérieur pour objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetToleranceLow

Fonction

Indique, pour les objets de barre, le seuil de tolérance inférieur.

Syntaxe

double GetToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil de tolérance inférieur pour objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetTypeAlarmHigh

Fonction

Indique, pour les objets de barre, si le seuil d'alarme supérieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'alarme supérieur est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil d'alarme supérieur est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeAlarmLow

Fonction

Indique, pour les objets de barre, si le seuil d'alarme inférieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'alarme inférieur est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil d'alarme inférieur est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeLimitHigh4

Fonction

Indique, pour les objets de barre, si le seuil supérieur Réserve 4 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil supérieur Réserve 4 est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil supérieur Réserve 4 est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeLimitHigh5

Fonction

Indique, pour les objets de barre, si le seuil supérieur Réserve 5 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil supérieur Réserve 5 est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil supérieur Réserve 5 est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeLimitLow4

Fonction

Indique, pour les objets de barre, si le seuil inférieur Réserve 4 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil inférieur Réserve 4 est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil inférieur Réserve 4 est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeLimitLow5

Fonction

Indique, pour les objets de barre, si le seuil inférieur Réserve 5 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil inférieur Réserve 5 est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil inférieur Réserve 5 est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeToleranceHigh

Fonction

Indique, pour les objets de barre, si le seuil de tolérance supérieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil de tolérance supérieur est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil de tolérance supérieur est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeToleranceLow

Fonction

Indique, pour les objets de barre, si le seuil de tolérance inférieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil de tolérance inférieur est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil de tolérance inférieur est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeWarningHigh

Fonction

Indique, pour les objets de barre, si le seuil d'avertissement supérieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

```
BOOL GetTypeWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'avertissement supérieur est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil d'avertissement supérieur est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetTypeWarningLow

Fonction

Indique, pour les objets de barre, si le seuil d'avertissement inférieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL GetTypeWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Pour les objets de barre, le seuil d'avertissement inférieur est indiqué en pour cent.

FALSE

Pour les objets de barre, le seuil d'avertissement inférieur est indiqué en valeur absolue.

Voir aussi

Exemple GetMarker (Page 1526)

Exemple GetMarker

GetWarningHigh

Fonction

Indique, pour les objets de barre, le seuil d'avertissement supérieur.

Syntaxe

double GetWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil d'avertissement supérieur pour objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

GetWarningLow

Fonction

Indique, pour les objets de barre, le seuil d'avertissement inférieur.

Syntaxe

double GetWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Seuil d'avertissement inférieur pour objets de barre

Voir aussi

Exemple GetAlarmHigh (Page 1516)

Exemple GetAlarmHigh

lien

Lien - brève description

Les fonctions du groupe Lien permettent de créer ou d'interroger la liaison de variable d'une propriété.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetLink

Fonction

Indique la liaison de variable actuelle pour des propriétés d'objet.

Syntaxe

BOOL GetLink(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName, LPLINKINFO *pLink);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Propriété de l'objet

pLink

Pointeur sur une structure du type : LINKINFO

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Définition de structure LINKINFO (Page 1598)

Exemple GetLink (Page 1524)

Exemple GetLink

Définition de structure LINKINFO

miscs

Miscs - brève description

Les fonctions du groupe Miscs vous permettent de modifier ou d'interroger de différentes propriétés d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetAdaptBorder

Fonction

Indique si la bordure du champ est adapté dynamiquement à la taille de texte pour des textes statiques, des champs E/S, des cases à cocher et des cases d'option.

Syntaxe

BOOL GetAdaptBorder(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Bordure est adaptée

FALSE

Bordure n'est pas adaptée

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetAdaptPicture

Fonction

Indique, pour les fenêtres de vue, si la vue est adaptée à la taille de la fenêtre.

Syntaxe

BOOL GetAdaptPicture(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Vue est adaptée

FALSE

Vue n'est pas adaptée

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetAdaptSize

Fonction

Indique, pour les fenêtres de vue, si la fenêtre doit être adaptée.

Syntaxe

BOOL GetAdaptSize(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Fenêtre est adaptée

FALSE

Fenêtre n'est pas adaptée

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetAverage

Fonction

Indique, pour les objets de barre, si le calcul de la moyenne est activé.

Syntaxe

BOOL GetAverage(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Calcul de la moyenne est activé pour les objets de barre.

FALSE

Calcul de la moyenne n'est pas activé pour les objets de barre.

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetBoxType

Fonction

Indique, pour les champs E/S, le type de champ (champ d'entrée, champ de sortie, champ d'entrée/sortie).

Syntaxe

long int GetBoxType(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Type de champ d'un champ E/S

Voir aussi

Champ E/S, Type de champ (Page 1585)

Champ d'E/S, Type de champ

GetCaption

Fonction

Indique, pour une fenêtre de vue et d'application si celle-ci possède un titre.

Syntaxe

BOOL GetCaption(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Fenêtre de vue/application possède un titre

FALSE

Fenêtre de vue/application ne possède pas de titre

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetCloseButton

Fonction

Indique, pour une fenêtre de vue, si la fenêtre de vue peut être fermée.

Syntaxe

BOOL GetCloseButton(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Fenêtre de vue peut être fermée

FALSE

Fenêtre de vue ne peut pas être fermée

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetColorChangeType

Fonction

Indique, pour des objets de barre, si le changement de couleur au cas où une valeur limite serait atteinte ne s'effectue que dans un segment de barre ou dans toute la barre.

Syntaxe

BOOL GetColorChangeType(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Changement de couleur s'effectue dans segment de barre

FALSE

Changement de couleur s'effectue dans toute la barre

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetCursorControl

Fonction

Indique, pour les champs E/S, si la commande du curseur est activée.

Syntaxe

BOOL GetCursorControl(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Commande du curseur est activée pour les champs E/S

FALSE

Commande du curseur est désactivée pour les champs E/S

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetCursorMode

Fonction

Indique si le mode curseur est curseur alpha ou pointeur pour la vue.

Syntaxe

BOOL GetCursorMode(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

TRUE

Mode curseur pour la vue est "Curseur alpha"

FALSE

Mode curseur pour la vue est "Curseur alpha"

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetEditAtOnce

Fonction

Indique, pour les champs E/S, si la propriété "Entrée immédiate" est activée.

Syntaxe

BOOL GetEditAtOnce(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Propriété "Entrée immédiate" est activée

FALSE

Propriété "Entrée immédiate" est désactivée

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetExtendedOperation

Fonction

Indique, pour les objets Slider, si la propriété "Commande étendue" est activée.

Syntaxe

```
BOOL GetExtendedOperation(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Propriété "Commande étendue" est activée

FALSE

Propriété "Commande étendue" est désactivée

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetHotkey

Fonction

Indique la combinaison de touches pour les cases à cocher.

Syntaxe

long int GetHotkey(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Code de touches pour les combinaisons de touches dans le cas de cases à cocher.

GetHysteresis

Fonction

Indique, pour les objets de barre, si l'affichage s'effectue avec ou sans hystérésis.

Syntaxe

BOOL GetHysteresis(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Affichage avec hystérésis dans le cas d'objets de barre

FALSE

Affichage sans hystérésis dans le cas d'objets de barre

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetHysteresisRange

Fonction

Indique, pour objets de barre, la valeur de l'hystérésis à l'affichage.

Syntaxe

double GetHysteresisRange(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Hystérésis à l'affichage pour des objets de barre

GetLanguageSwitch

Fonction

Indique pour l'objet "Liste de textes" si les textes à affecter doivent être enregistrés dans la bibliothèque de textes ou dans l'objet proprement dit.

Syntaxe

BOOL GetLanguageSwitch(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Les textes à affecter sont enregistrés dans la bibliothèque de textes

FALSE

Les textes à affecter sont enregistrés dans l'objet de liste de textes

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetLastChange

Fonction

Indique la date de la dernière modification de la vue.

Syntaxe

char* GetLastChange(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Date de la dernière modification de la vue.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Exemple GetPictureName (Page 1529)

Exemple GetPictureName

GetMax

Fonction

Indique la valeur maximale pour les objets de barre et les objets Slider.

Syntaxe

double GetMax(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur maximale pour les objets de barre et les objets Slider

GetMaximizeButton

Fonction

Indique, pour une fenêtre de vue ou d'application, si la fenêtre est maximisable.

Syntaxe

BOOL GetMaximizeButton(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Fenêtre de vue ou d'application est maximisable.

FALSE

Fenêtre de vue ou d'application n'est pas maximisable.

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetMin

Fonction

Indique la valeur minimale pour les objets de barre et les objets Slider.

Syntaxe

double GetMin(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur minimale pour les objets de barre et les objets Slider

GetMoveable

Fonction

Indique, pour une fenêtre de vue ou d'application, si la fenêtre est déplaçable.

Syntaxe

BOOL GetMoveable(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Fenêtre de vue ou d'application est déplaçable.

FALSE

Fenêtre de vue ou d'application n'est pas déplaçable.

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetOffsetLeft

Fonction

Indique,pour des fenêtres de vue, la distance horizontale de la vue du bord de fenêtre gauche.

Syntaxe

long int GetOffsetLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Distance horizontale de la vue du bord de fenêtre gauche dans le cas de fenêtres de vue

GetOffsetTop

Fonction

Indique,pour des fenêtres de vue, la distance verticale de la vue du bord de fenêtre supérieur.

Syntaxe

long int GetOffsetTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Distance verticale de la vue du bord de fenêtre supérieur dans le cas des fenêtres de vue.

GetOnTop

Fonction

Indique, pour une fenêtre de vue ou d'application, si la fenêtre se trouve toujours au premier plan.

Syntaxe

BOOL GetOnTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Fenêtre de vue ou d'application n'est pas toujours au premier plan

FALSE

Fenêtre de vue ou d'application peut être recouverte par d'autres fenêtres

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetOperation

Fonction

Indique si l'objet est manipulable.

Syntaxe

BOOL GetOperation(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Objet est manipulable

FALSE

Objet n'est pas manipulable

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetOperationMessage

Fonction

Indique, pour des champs E/S, des cases à cocher, des cases d'option ou des Slider, si une alarme est sortie lors de la commande.

Syntaxe

BOOL GetOperationMessage(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Une alarme est sortie lors de la commande

FALSE

Une alarme n'est pas sortie lors de la commande

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetOperationReport

Fonction

Indique pour tous les objets, sauf fenêtres d'application et de vue ainsi que OLE-Control, si la raison pour la commande est journalisée.

Syntaxe

BOOL GetOperationReport(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La raison pour la commande est journalisée

FALSE

La raison pour la commande n'est pas journalisée

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetPasswordLevel

Fonction

Indique pour tous les objets, sauf fenêtres d'application et de vue ainsi que OLE-Control, le niveau d'autorisation pour la commande de l'objet.

Syntaxe

long int GetPasswordLevel(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Niveau d'autorisation pour la commande de l'objet

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

GetPictureName

Fonction

Indique le nom de la vue affichée actuellement quand il s'agit de fenêtres de vue.

Syntaxe

char* GetPictureName(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de la fenêtre de vue

Valeur de retour

Pointeur sur le nom de la vue actuellement affichée

Remarque

Si les deux paramètres sont NULL, le pointeur montre le nom de la vue de base.

Voir aussi

Exemple GetPictureName (Page 1529)

Exemple GetPictureName

GetProcess

Fonction

Indique, pour les objets de barre et les objets Slider, la valeur du paramétrage par défaut pour la valeur de process à afficher.

Indique les champs sélectionnés pour des cases à cocher et des cases d'option.

Syntaxe

double GetProcess(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

- Pour les objets de barre et les objets Slider : Valeur du paramétrage par défaut pour la valeur de process à afficher
- Pour cases à cocher et cases d'option : Chaque champ est représenté par un bit dans un mot de 32 bits (Champ1 correspond à la valeur indicative Bit 0). Les champs sélectionnés sont marqués par un bit positionné. Des champs non existants sont assignés par 0.

GetScrollBars

Fonction

Indique, pour les fenêtres de vue, si la fenêtre de vue possède une barre de défilement.

Syntaxe

BOOL GetScrollBars(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Fenêtre de vue possède une barre de défilement

FALSE

Fenêtre de vue ne possède pas de barre de défilement

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetServerName

Fonction

Indique, pour OLE-Control et objet OLE, le paramétrage par défaut pour la valeur de process à afficher.

Syntaxe

char* GetServerName(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Nom de l'objet (OLE-Control et objet OLE) sous lequel il est enregistré dans WINDOWS.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Exemple GetPictureName (Page 1529)

Exemple GetPictureName

GetSizeable

Fonction

Indique, pour une fenêtre de vue ou d'application, si la taille de la fenêtre peut être modifiée.

Syntaxe

BOOL GetSizeable(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La taille de la fenêtre d'application ou de vue peut être modifiée

FALSE

La taille de la fenêtre d'application ou de vue ne peut pas être modifiée

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetSmallChange

Fonction

Indique, dans le cas d'un objet Slider, le nombre de pas qu'effectue le curseur lors d'un clic de souris.

Syntaxe

long int GetSmallChange(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Nombre de pas qu'effectue le curseur lors d'un clic de souris

GetTagPrefix

Fonction

Renvoie, pour des fenêtres de vue, le préfixe de variable d'une fenêtre de vue.

Syntaxe

char* GetTagPrefix(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Préfixe de variable de la fenêtre de vue.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Exemple GetTagPrefix (Page 1544)

Exemple GetTagPrefix

GetTrend

Fonction

Indique, pour les objets de barre, si l'affichage de la tendance est activée.

Syntaxe

BOOL GetTrend(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Affichage de la tendance est activée pour un objet de barre

FALSE

Affichage de la tendance n'est pas activée pour un objet de barre

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetUpdateCycle

Fonction

Indique le cycle de rafraîchissement pour la vue entière.

Syntaxe

long int GetUpdateCycle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Cycle de rafraîchissement sous forme de valeur en chiffre

Voir aussi

Définition de structure LINKINFO (Page 1598)

Définition de structure LINKINFO

GetVisible

Fonction

Indique si l'objet est affiché.

Syntaxe

BOOL GetVisible(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Objet est affiché

FALSE

Objet n'est pas affiché

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetWindowBorder

Fonction

Indique, pour une fenêtre de vue ou d'application, si l'objet est représenté avec bordure.

Syntaxe

BOOL GetWindowBorder(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La fenêtre de vue ou d'application est représentée avec bordure

FALSE

La fenêtre de vue ou d'application est représentée sans bordure

Voir aussi

Exemple GetVisible (Page 1551)

Exemple GetVisible

GetZeroPointValue

Fonction

Indique, pour objets de barre, la valeur absolue de l'origine.

Syntaxe

double GetZeroPointValue(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur absolue de l'origine à l'affichage de barre

GetZoom

Fonction

Indique, pour la fenêtre de vue, le facteur de mise à l'échelle.

Syntaxe

long int GetZoom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Facteur de mise à l'échelle d'une fenêtre de vue.

ole_control

OLE_control - brève description

Les fonctions du groupe ole_Control sont seulement applicables aux objets OCX-Slider.

A l'aide de ces fonctions vous pouvez modifier ou interroger les différentes propriétés et paramétrages d'un objet OCX-Slider.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetPosition

Fonction

Indique la position de la coulisse du curseur pour les objets OCX-Slider.

Syntaxe

long int GetPosition(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Position de la coulisse du curseur de l'objet OCX Slider sous forme de valeur en chiffre

Voir aussi

Exemple GetPosition (Page 1530)

Exemple GetPosition

GetRangeMax

Fonction

Indique la plage de réglage "Max" pour les objets OCX Slider.

Syntaxe

long int GetRangeMax(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Plage de réglage "Max" pour les objets OCX Slider sous forme de valeur en chiffre

Voir aussi

Exemple GetRangeMax (Page 1532)

Exemple GetRangeMax

GetRangeMin

Fonction

Indique la plage de réglage "Min" pour les objets OCX Slider.

Syntaxe

long int GetRangeMin(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Plage de réglage "Min" pour les objets OCX Slider sous forme de valeur en chiffre

Voir aussi

Exemple GetRangeMin (Page 1533)

Exemple GetRangeMin

pictures

Pictures - brève description

Les fonctions du groupe Pictures vous permettent de modifier ou d'interroger de différentes propriétés des vues d'objets graphiques ou de boutons ronds.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetPicDeactReferenced

Fonction

Indique pour les boutons ronds si la vue est référenciée pour l'état " désactivé".

Syntaxe

BOOL GetPicDeactReferenced(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La vue assignée à l'état "désactivé" n'a pas été mémorisée dans l'objet.

FALSE

La vue assignée à l'état "désactivé" n'a pas été mémorisée dans l'objet.

GetPicDeactTransparent

Fonction

Indique pour les boutons ronds la couleur transparente pour l'état "désactivé".

Syntaxe

long int GetPicDeactTransparent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur transparente pour l'état "désactivé" sous forme de valeur en chiffre

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)
Exemple GetBackColor (Page 1516)
Exemple GetBackColor
Tableau de couleurs

GetPicDeactUseTransColor

Fonction

Indique pour les boutons ronds si la couleur transparente est utilisée pour l'état "désactivé".

Syntaxe

BOOL GetPicDeactUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Couleur transparente pour l'état "désactivé" est utilisée.

FALSE

Couleur transparente pour l'état "désactivé" n'est pas utilisée.

GetPicDownReferenced

Fonction

Indique pour les boutons ronds si la vue est référenciée pour l'état "activé/enfoncé".

Syntaxe

BOOL GetPicDownReferenced(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La vue assignée à l'état "activé/enfoncé" n'a pas été enregistrée dans l'objet.

FALSE

La vue assignée à l'état "activé/enfoncé" a été enregistrée dans l'objet.

GetPicDownTransparent

Fonction

Indique pour les boutons ronds la couleur transparente pour l'état "activé/enfoncé".

Syntaxe

long int GetPicDownTransparent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur transparente pour l'état "activé/enfoncé" sous forme de valeur en chiffre

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple GetPictureDown (Page 1528)

Tableau de couleurs

Exemple GetPictureDown

GetPicDownUseTransColor

Fonction

Indique pour les boutons ronds si la couleur transparente est utilisée pour l'état "activé/enfoncé".

Syntaxe

BOOL GetPicDownUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Couleur transparente pour l'état "activé/enfoncé" est utilisée

FALSE

Couleur transparente pour l'état "activé/enfoncé" n'est pas utilisée

GetPicReferenced

Fonction

Indique, pour des objets graphiques, si la vue est référencée.

Syntaxe

BOOL GetPicReferenced(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La vue assignée n'a pas été enregistrée dans l'objet.

FALSE

La vue assignée a été enregistrée dans l'objet.

GetPicTransColor

Fonction

Indique, pour des objets graphiques, la couleur transparente pour une vue d'arrière-plan.

Syntaxe

long int GetPicTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur transparente pour une vue d'arrière-plan d'un objet graphique sous forme de valeur en chiffre

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple GetBackColor (Page 1516)

Exemple GetBackColor

Tableau de couleurs

GetPictureDeactivated

Fonction

Indique pour les boutons ronds le nom de vue pour l'état "désactivé".

Syntaxe

char* GetPictureDeactivated(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Nom de vue pour l'état "désactivé".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Remarque

Des fichiers Bitmap (*.bmp, *.dib) ainsi que des métafichiers (*.emf, *.wmf) peuvent être intégrés.

GetPictureDown

Fonction

Indique pour les boutons ronds le nom de vue pour l'état "activé/enfoncé".

Syntaxe

char* GetPictureDown(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Nom de vue pour l'état "activé/enfoncé".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Remarque

Des fichiers Bitmap (*.bmp, *.dib) ainsi que des métafichiers (*.emf, *.wmf) peuvent être intégrés.

Voir aussi

Exemple GetPictureDown (Page 1528)

Exemple GetPictureDown

GetPictureUp

Fonction

Indique pour les boutons ronds le nom de vue pour l'état "désactivé/non enfoncé".

Syntaxe

char* GetPictureUp(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Nom de vue pour l'état "désactivé/non enfoncé".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Remarque

Des fichiers Bitmap (*.bmp, *.dib) ainsi que des métafichiers (*.emf, *.wmf) peuvent être intégrés.

Voir aussi

Exemple GetPictureUp (Page 1529)

Exemple GetPictureUp

GetPicUpReferenced

Fonction

Indique pour les boutons ronds si la vue est référencée pour l'état "désactivé/non enfoncé".

Syntaxe

BOOL GetPicUpReferenced(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La vue assignée à l'état "désactivé/non enfoncé" n'a pas été enregistrée dans l'objet.

FALSE

La vue assignée à l'état "désactivé/non enfoncé" a été enregistrée dans l'objet.

GetPicUpTransparent

Fonction

Indique pour les boutons ronds la couleur transparente pour l'état "désactivé/non enfoncé".

Syntaxe

long int GetPicUpTransparent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur transparente pour l'état "désactivé/non enfoncé" sous forme de valeur en chiffre

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple GetBackColor (Page 1516)

Exemple GetBackColor

Tableau de couleurs

GetPicUpUseTransColor

Fonction

Indique pour les boutons ronds si la couleur transparente est utilisée pour l'état " Mise hors circuit/ Non enfoncé".

Syntaxe

BOOL GetPicUpUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Couleur transparente est utilisée pour l'état " Mise hors circuit/ Non enfoncé".

FALSE

Couleur transparente n'est pas utilisée pour l'état " Mise hors circuit/ Non enfoncé".

GetPicUseTransColor

Fonction

Indique pour les objets graphiques si la couleur transparente est utilisée pour une vue d'arrière-plan.

Syntaxe

BOOL GetPicUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Couleur transparente est utilisée pour une vue d'arrière-plan.

FALSE

Couleur transparente n'est pas utilisée pour une vue d'arrière-plan.

property

Property - brève description

Les fonctions du groupe Property vous permettent de modifier ou d'interroger des propriétes d'objets qui ne possèdent pas de propre fonction.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetPropBOOL

Fonction

Indique l'état actuel d'une propriété du type de données "BOOL".

Syntaxe

BOOL GetPropBOOL(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

Valeur de retour

Valeur de l'attribut dans le type de données "BOOL"

Voir aussi

Exemple GetPropBOOL (Page 1531)

Exemple GetPropBOOL

GetPropChar

Fonction

Indique l'état actuel d'une propriété du type de données "char".

Syntaxe

char* GetPropChar(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

Valeur en retour

Pointeur sur une chaîne de caractères contenant la valeur de la propriété d'objet.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Exemple GetPropChar (Page 1531)

Exemple GetPropChar

GetPropDouble

Fonction

Indique l'état actuel d'une propriété du type de données "double".

Syntaxe

double GetPropDouble(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

Valeur de retour

Valeur de l'attribut dans le type de données "double"

GetPropWord

Fonction

Indique l'état actuel d'une propriété du type de données "long".

Syntaxe

long GetPropWord(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

Valeur de retour

Valeur de l'attribut dans le type "long"

state

State - brève description

Les fonctions du groupe State vous permettent de modifier ou d'interroger de différentes propriétés d'indicateurs d'état.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetBasePicReferenced

Fonction

Indique dans l'indicateur d'état si la vue de base est référencée.

Syntaxe

BOOL GetBasePicReferenced(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La vue de base n'a pas été enregistrée dans l'objet.

FALSE

La vue de base a été enregistrée dans l'objet.

GetBasePicTransColor

Fonction

Indique dans l'indicateur d'état la couleur transparente de la vue de base.

Syntaxe

long int GetBasePicTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur transparente de la vue de base sous forme de valeur en chiffre

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple GetBackColor (Page 1516)

Exemple GetBackColor

Tableau de couleurs

GetBasePicture

Fonction

Indique le nom de la vue de base pour l'indicateur d'état.

Syntaxe

char* GetBasePicture(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Nom de la vue de base pour l'indicateur d'état.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
   .......
}
```

GetBasePicUseTransColor

Fonction

Indique dans l'indicateur d'état si la couleur transparente est utilisée pour la vue de base.

Syntaxe

BOOL GetBasePicUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Couleur transparente est utilisée pour la vue de base

FALSE

Couleur transparente n'est pas utilisée pour la vue de base

GetFlashFlashPicture

Fonction

Indique dans l'indicateur si la vue clignotante est animée de façon dynamique ou statique.

Syntaxe

BOOL GetFlashFlashPicture(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La vue clignotante est animée de façon dynamique.

FALSE

La vue clignotante est animée de façon statique.

GetFlashPicReferenced

Fonction

Indique dans l'indicateur d'état si la vue clignotante est référencée.

Syntaxe

BOOL GetFlashPicReferenced(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La vue clignotante n'a pas été enregistrée dans l'objet.

FALSE

La vue clignotante a été enregistrée dans l'objet.

GetFlashPicTransColor

Fonction

Indique dans l'indicateur d'état la couleur transparente de la vue clignotante.

Syntaxe

long int GetFlashPicTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Couleur transparente de la vue clignotante sous forme de valeur en chiffre

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple GetBackColor (Page 1516)

Exemple GetBackColor

Tableau de couleurs

GetFlashPicture

Fonction

Indique dans l'indicateur d'état le nom de la vue clignotante.

Syntaxe

char* GetFlashPicture(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Nom de la vue clignotante (Nom de fichier du graphique).

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

GetFlashPicUseTransColor

Fonction

Indique dans l'indicateur d'état si la couleur transparente est utilisée pour la vue clignotante.

Syntaxe

BOOL GetFlashPicUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Couleur transparente est utilisée pour la vue clignotante

FALSE

Couleur transparente n'est pas utilisée pour la vue clignotante

GetFlashRateFlashPic

Fonction

Indique dans l'indicateur d'état la fréquence de clign. d'une vue clignotante.

Syntaxe

long int GetFlashRateFlashPic(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Fréquence de clign. d'une vue clignotante sous forme de valeur en chiffre

---

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence dépend du système et du matériel (nombre d'objets, processeur, mémoire, cycle de rafraîchissement, etc.).

Voir aussi

Fréquences de clignotement (Page 1583)

Exemple GetFlashRateFlashPic (Page 1520)

Fréquences de clignotement

Exemple GetFlashRateFlashPic

GetIndex

Fonction

Indique l'indice du point actuel pour des polygones ou des traits polygonaux.

Indique l'indice du champ actuel pour des cases à cocher et des cases d'option.

Syntaxe

long int GetIndex(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Indice du point ou champ actuel

style

Style - brève description

Les fonctions du groupe Style permettent de modifier ou d'interroger de différentes propriétés concernant la présentation d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

GetBackBorderWidth

Fonction

Indique la largeur de l'encadrement de bordures 3D et d'objets Slider.

Syntaxe

long int GetBackBorderWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Valeur en chiffre pour la largeur de l'encadrement de bordures 3D et d'objets Slider.

Voir aussi

Exemple GetBorderStyle (Page 1517)

Exemple GetBorderStyle

GetBorderEndStyle

Fonction

Indique le type de la fin de ligne.

Syntaxe

long int GetBorderEndStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Type de la fin de ligne sous forme de valeur en chiffre.

Voir aussi

Exemple GetBorderStyle (Page 1517)

Fins de trait (Page 1589)

Exemple GetBorderStyle

Fins de trait

GetBorderStyle

Fonction

Indique le type de ligne ou de bordure.

Syntaxe

long int GetBorderStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Type de ligne ou de bordure sous forme de valeur en chiffre

Voir aussi

Exemple GetBorderStyle (Page 1517)
Types de trait (Page 1588)
Exemple GetBorderStyle
Types de trait

GetBorderWidth

Fonction

Indique la largeur des lignes ou de la ligne de bordure.

Syntaxe

long int GetBorderWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Largeur des lignes ou de la ligne de bordure sous forme de valeur en chiffre

Voir aussi

Exemple GetBorderStyle (Page 1517)

Exemple GetBorderStyle

GetBoxAlignment

Fonction

Indique la disposition des éléments de commande (cadrés à gauche ou à droite) dans des cases à cocher ou des cases d'option.

Syntaxe

long int GetBoxAlignment(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Disposition des éléments de commande dans une case à cocher ou une case d'option sous forme de valeur en chiffre

Voir aussi

Exemple GetBorderStyle (Page 1517)

Sens du texte (Page 1591)

Exemple GetBorderStyle

Alignement du texte

GetFillStyle

Fonction

Indique le type du motif de rempl.

Syntaxe

long int GetFillStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Type du motif de rempl. sous forme de valeur en chiffre

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Motif de remplissage (Page 1587)

Exemple GetFillStyle (Page 1518)

Motif de remplissage

Exemple GetFillStyle

GetFillStyle2

Fonction

Indique, pour un affichage de barre, le motif de remplissage de la barre.

Syntaxe

long int GetFillStyle2(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Motif de rempl. de la barre sous forme de valeur en chiffre

Voir aussi

Motif de remplissage (Page 1587)

Exemple GetFillStyle (Page 1518)

Exemple GetFillStyle

Motif de remplissage

GetItemBorderStyle

Fonction

Indique, pour l'objet "liste de textes", le type de ligne de séparation.

Syntaxe

long int GetItemBorderStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

Type de ligne de séparation pour l'objet "liste de textes"

Voir aussi

Exemple GetBorderStyle (Page 1517)

Types de trait (Page 1588)

Exemple GetBorderStyle

Types de trait

GetItemBorderWidth

Fonction

Indique, pour l'obj. "liste de textes", la largeur de ligne de sép.

Syntaxe

long int GetItemBorderWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

Largeur de ligne de sép. pour l'objet "liste de textes" sous forme de valeur en chiffre

Voir aussi

Exemple GetBorderStyle (Page 1517)

Exemple GetBorderStyle

GetPressed

Fonction

Indique, pour les boutons ou bouton ronds, si la position du bouton est "enfoncé" ou "non enfoncé".

Syntaxe

BOOL GetPressed(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

TRUE

Position du bouton est "enfoncé"

FALSE

Position du bouton est "non enfoncé"

GetToggle

Fonction

Indique pour des boutons ou boutons ronds si le bouton est encliquetable ou non.

Syntaxe

BOOL GetToggle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur en retour

TRUE

Bouton est encliquetable

FALSE

Bouton n'est pas encliquetable

GetWindowsStyle

Fonction

Indique pour les boutons s'il sont représentés dans le style Windows.

Syntaxe

BOOL GetWindowsStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

Bouton est représenté de la manière habituelle sous Windows.

FALSE

Vous déterminez vous-même l'apparence du bouton.

set

axes

Axes - brève description

Les fonctions du groupe Axes sont seulement applicables aux objets de barre.

A l'aide de ces fonctions vous pouvez modifier ou interroger de différentes propriétés d'un objet de barre.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetAlignment

Fonction

Règle, pour des objets de barre, si le texte se trouve à droite ou à gauche de la barre.

Syntaxe

BOOL SetAlignment(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bAlignment);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bAlignment

Sens du texte

TRUE	Texte à droite de la barre
FALSE	Texte à gauche de la barre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetAxisSection

Fonction

Paramètre, pour les objets de barre, la section d'axe, c.-à-d. la différence de valeurs entre deux textes des axes voisins.

Syntaxe

BOOL SetAxisSection(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dAxisSection);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dAxisSection

Section d'axe

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

SetExponent

Fonction

Paramètre, pour les objets de barre, la représentation des textes des axes (exponentielle/décimale).

Syntaxe

BOOL SetExponent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bExponent);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bExponent

Textes d'axe

TRUE	Textes des axes en représentation exponentielle
FALSE	Textes des axes en représentation décimale

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetLeftComma

Fonction

Règle, pour les objets de barre, le nombre des chiffres avant la virgule dans les textes des axes.

Syntaxe

BOOL SetLeftComma(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lLeftComma);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lLeftComma

Nombre des chiffres avant la virgule

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetLongStrokesBold

Fonction

Règle, pour les objets de barre, si les grandes divisions sont représentées en gras ou normalement.

Syntaxe

BOOL SetLongStrokesBold(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bLongStrokesBold);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bLongStrokesBold

Grandes divisions sur l'échelle de la barre

TRUE	gras
FALSE	Normal

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetLongStrokesOnly

Fonction

Détermine, pour les objets de barre, si également des subdivisions sont utilisées sur l'échelle de barre.

Syntaxe

BOOL SetLongStrokesOnly(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bLongStrokesOnly);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bLongStrokesOnly

Grandes divisions uniquement oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetLongStrokesSize

Fonction

Paramètre, pour les objets de barre, la langueur des des grandes divisions sur l'échelle de barre.

Syntaxe

BOOL SetLongStrokesSize(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lLongStrokesSize);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lLongStrokesSize

Longueur des grandes divisions en pixels.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetRightComma

Fonction

Détermine, pour les objets de barre, le nombre des chiffres après la virgule dans les textes des axes.

Syntaxe

BOOL SetRightComma(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRightComma);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRightComma

Nombre des chiffres après la virgule

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetScaleTicks

Fonction

Règle, pour les objets de barre, la division d'échelle sous forme de nombre de sections d'échelle. La section d'échelle étant la zone délimitée par deux grandes divisions.

Syntaxe

BOOL SetScaleTicks(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lScaleTicks);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lScaleTicks

Nombre des sections d'échelle

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si le nombre des sections d'échelle est indiqué par 0, l'objet de barre calcule lui-même une division d'échelle appropriée.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetScaling

Fonction

Active ou désactive l'échelle de barre pour les objets de barre.

Syntaxe

BOOL SetScaling(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bScaling);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bScaling

Echelle activée/désactivée

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetScaling (Page 1565)

Exemple SetScaling

SetScalingType

Fonction

Détermine, pour les objets de barre, le type de la mise à l'échelle de la barre.

Syntaxe

BOOL SetScalingType(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lScalingType);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lScalingType

Type de la mise à l'échelle sous forme de valeur en chiffre.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Mise à l'échelle du bargraphe (Page 1583)

Exemple SetScaling (Page 1565)

Exemple SetScaling

Mise à l'échelle du bargraphe

color

Color- brève description

Les fonctions du groupe Color vous permettent de modifier ou d'interroger de différentes propriétés de couleur sur des objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetBackColor

Fonction

Paramètre la couleur d'arrière-plan d'un objet.

Syntaxe

BOOL SetBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackColor

Couleur d'arrière-plan de l'objet sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetBackColor2

Fonction

Paramètre la couleur de barre pour des objets de barre.

Syntaxe

BOOL SetBackColor2(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackColor2);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackColor2

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetBackColor3

Fonction

Paramètre la couleur d'arrière-plan de la barre pour des objets de barre.

Syntaxe

BOOL SetBackColor3(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackColor3);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackColor3

Couleur d'arrière-plan de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetBackColorBottom

Fonction

Paramètre, pour les objets slider, la couleur d'arrière-plan en bas à droite.

Syntaxe

BOOL SetBackColorBottom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackColorBottom);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackColorBottom

Couleur d'arrière-plan d'objets Slider sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetBackColorTop

Fonction

Paramètre, pour les objets slider, la couleur d'arrière-plan en haut à gauche.

Syntaxe

BOOL SetBackColorTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackColorTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackColorTop

Couleur d'arrière-plan d'objets Slider sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetBorderBackColor

Fonction

Règle la couleur d'arrière-plan des lignes ou de la bordure.

Syntaxe

BOOL SetBorderBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderBackColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderBackColor

Couleur d'arrière-plan des lignes ou de la bordure

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetBorderColor

Fonction

Règle la couleur des lignes ou de la bordure.

Syntaxe

BOOL SetBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderColor

Couleur des lignes ou de la bordure sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetBorderColorBottom

Fonction

Règle la couleur de la bordure 3D en bas.

Syntaxe

BOOL SetBorderColorBottom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderColorBottom);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderColorBottom

Couleur de bordure 3D sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetBorderColorTop

Fonction

Règle la couleur de la bordure 3D en haut.

Syntaxe

BOOL SetBorderColorTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderColorTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderColorTop

Couleur de bordure 3D sous forme de valeur en chiffre en haut

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetButtonColor

Fonction

Détermine, pour les objets slider, la couleur de bouton.

Syntaxe

BOOL SetButtonColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lButtonColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lButtonColor

Pour les objets slider, couleur de bouton sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetColorBottom

Fonction

Détermine, pour les objets slider, la couleur de la limite inférieure.

Syntaxe

BOOL SetColorBottom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorBottom);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorBottom

Pour les objets slider, couleur de la limite inférieure sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetColorTop

Fonction

Détermine, pour les objets slider, la couleur de la limite supérieure.

Syntaxe

BOOL SetColorTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorTop

Pour les objets slider, couleur de la limite supérieure sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetFillColor

Fonction

Détermine la couleur du motif de remplissage.

Syntaxe

BOOL SetFillColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFillColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFillColor

Couleur du motif de rempl. sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetForeColor

Fonction

Règle la couleur de police.

Syntaxe

BOOL SetForeColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lForeColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lForeColor

Couleur de police sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetItemBorderBackColor

Fonction

Règle la couleur d'arrière-plan de la ligne de séparation pour l'objet "liste de textes".

Syntaxe

BOOL SetItemBorderBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lItemBorderBackColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lItemBorderBackColor

Couleur d'arrière-plan de la ligne de sép. sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetItemBorderColor

Fonction

Détermine la couleur de ligne de séparation pour l'objet "liste de textes".

Syntaxe

BOOL SetItemBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lItemBorderColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lItemBorderColor

Couleur de ligne de séparation sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetScaleColor

Fonction

Paramètre la couleur d'échelle pour des objets de barre.

Syntaxe

BOOL SetScaleColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lScaleColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lScaleColor

Pour les obj. de barre, couleur d'échelle sous forme de valeur en chiffre.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetSelBGColor

Fonction

Détermine, pour l'objet "liste de textes", la couleur d'arrière-plan pour l'entrée sélectionnée.

Syntaxe

BOOL SetSelBGColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lSelBGColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lSelBGColor

Couleur d'arrière-plan pour l'entrée sélectionnée sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetSelTextColor

Fonction

Détermine, pour l'objet "liste de textes", la couleur de police pour une entrée sélectionnée.

Syntaxe

BOOL SetSelTextColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lSelTextColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lSelTextColor

Couleur de police pour l'entrée sélectionnée sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetTrendColor

Fonction

Paramètre la couleur de tendance pour des objets de barre.

Syntaxe

BOOL SetTrendColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lTrendColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lTrendColor

Couleur de tendance sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetUnselBGColor

Fonction

Détermine, pour l'objet "liste de textes", la couleur d'arrière-plan pour les entrées non sélectionnée.

Syntaxe

BOOL SetUnselBGColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lUnselBGColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lUnselBGColor

Couleur d'arrière-plan pour entrées non sélectionnées sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetUnselTextColor

Fonction

Détermine, pour l'objet "liste de textes", la couleur de police pour les entrées non sélectionnée.

Syntaxe

BOOL SetUnselTextColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lUnselTextColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lUnselTextColor

Couleur de police pour entrées non sélectionnées sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

fill

Fill - brève description

Les fonctions du groupe Fill pilotent le remplissage dynamique d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetFilling

Fonction

Active ou désactive le remplissage dynamique avec la couleur d'arrière-plan.

Syntaxe

BOOL SetFilling(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFilling);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFilling

Remplissage dynamique avec la couleur d'arrière-plan activé/désactivé.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Exemple SetFilling (Page 1557)

Exemple SetFilling

SetFillingIndex

Fonction

Règle le taux de remplissage.

Syntaxe

BOOL SetFillingIndex(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFillingIndex);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFillingIndex

Taux de rempl. sous forme de valeur en chiffre (0 – 100)

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Exemple SetFillingIndex (Page 1557)

Exemple SetFillingIndex

flash

Flash - brève description

Les fonctions du groupe Flash permettent de modifier ou d'interroger de différentes propriétés concernant le clignotement.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetBackFlashColorOff

Fonction

Détermine la couleur clignotante d'arrière-plan pour l'état désactivé.

Syntaxe

BOOL SetBackFlashColorOff(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackFlashColorOff);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackFlashColorOff

Couleur clignotante d'arrière-plan pour l'état désactivé sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetFlashBackColorOn (Page 1558)

Tableau de couleurs

Exemple SetFlashBackColorOn

SetBackFlashColorOn

Fonction

Détermine la couleur clignotante d'arrière-plan pour l'état activé.

Syntaxe

BOOL SetBackFlashColorOn(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackFlashColorOn);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackFlashColorOn

Couleur clignotante d'arrière-plan pour l'état activé sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetFlashBackColorOn (Page 1558)

Tableau de couleurs

Exemple SetFlashBackColorOn

SetBorderFlashColorOff

Fonction

Détermine la couleur clignotante de la bordure ou de ligne pour l'état désactivé.

Syntaxe

```
BOOL SetBorderFlashColorOff(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,
long int lBorderFlashColorOff);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderFlashColorOff

Couleur de bordure ou de ligne clignotante pour l'état désactivé sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Exemple SetFlashBackColorOn (Page 1558)

Tableau de couleurs (Page 1585)

Tableau de couleurs

Exemple SetFlashBackColorOn

SetBorderFlashColorOn

Fonction

Détermine la couleur clignotante de bordure ou de ligne pour l'état activé.

Syntaxe

BOOL SetBorderFlashColorOn(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderFlashColorOn);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderFlashColorOn

Couleur de bordure ou de ligne clignotante pour l'état activé sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetFlashBackColorOn (Page 1558)

Tableau de couleurs

Exemple SetFlashBackColorOn

SetFlashBackColor

Fonction

Active ou désactive le clignotement de l'arrière-plan.

Syntaxe

BOOL SetFlashBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFlashBackColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFlashBackColor

Clignotement de l'arrière-plan activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Exemple SetFlashBackColor (Page 1558)

Exemple SetFlashBackColor

SetFlashBorderColor

Fonction

Active ou désactive le clignotement de la bordure ou de la ligne.

Syntaxe

BOOL SetFlashBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFlashBorderColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFlashBorderColor

Clignotement de la bordure ou de la ligne activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Exemple SetFlashBackColor (Page 1558)

Tableau de couleurs

SetFlashForeColor

Fonction

Active ou désactive le clignotement de la police.

Syntaxe

BOOL SetFlashForeColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFlashForeColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFlashForeColor

Clignotement de la police activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Exemple SetFlashBackColor (Page 1558)

Exemple SetFlashBackColor

SetFlashRateBackColor

Fonction

Détermine la fréquence de clignotement de l'arrière-plan.

Syntaxe

BOOL SetFlashRateBackColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFlashRateBackColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFlashRateBackColor

Fréquence de clignotement de l'arrière-plan

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Fréquences de clignotement (Page 1583)

Exemple SetFlashBackColor (Page 1558)

Exemple SetFlashBackColor

Fréquences de clignotement

SetFlashRateBorderColor

Fonction

Détermine la fréquence de clignotement de la ligne ou de la bordure.

Syntaxe

BOOL SetFlashRateBorderColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFlashRateBorderColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFlashRateBorderColor

Fréquence du clignotement de la ligne ou de la bordure.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Fréquences de clignotement (Page 1583)

Exemple SetFlashBackColor (Page 1558)

Exemple SetFlashBackColor

Fréquences de clignotement

SetFlashRateForeColor

Fonction

Détermine la fréquence de clignotement de la police.

Syntaxe

BOOL SetFlashRateForeColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFlashRateForeColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFlashRateForeColor

Fréquence de clignotement de la police

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Fréquences de clignotement (Page 1583)

Exemple SetFlashBackColor (Page 1558)

Fréquences de clignotement

Exemple SetFlashBackColor

SetForeFlashColorOff

Fonction

Détermine la couleur clignotante de la police pour l'état désactivé.

Syntaxe

BOOL SetForeFlashColorOff(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lForeFlashColorOff);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lForeFlashColorOff

Couleur clignotante de la police pour l'état désactivé sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetFlashBackColorOn (Page 1558)

Exemple SetFlashBackColorOn

Tableau de couleurs

SetForeFlashColorOn

Fonction

Détermine la couleur clignotante de la police pour l'état activé.

Syntaxe

BOOL SetForeFlashColorOn(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lForeFlashColorOn);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lForeFlashColorOn

Couleur clignotante de la police pour l'état activé sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetFlashBackColorOn (Page 1558)

Exemple SetFlashBackColorOn

Tableau de couleurs

focus

Focus - abrégé descriptif

Les fonctions du groupe Focus permettent de placer le focus sur un objet ou bien interroger quel objet possède le focus.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Set_Focus

Fonction

Met le focus sur l'objet indiqué.

Syntaxe

```
BOOL Set_Focus(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetFocus (Page 1559)

Exemple SetFocus

font

Font - Brève description

Les fonctions du groupe Font permettent de modifier ou d'interroger les propriétés concernant le texte.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetAlignmentLeft

Fonction

Détermine le sens du texte horizontal (à gauche, centré, à droite).

Syntaxe

BOOL SetAlignmentLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lAlignmentLeft);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lAlignmentLeft

Sens du texte horizontal sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Sens du texte (Page 1591)

Exemple SetFontSize (Page 1560)

Alignement du texte

Exemple SetFontSize

SetAlignmentTop

Fonction

Détermine le sens du texte vertical (en haut, centré, en bas).

Syntaxe

BOOL SetAlignmentTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lAlignmentTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lAlignmentTop

Sens du texte vertical sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Sens du texte (Page 1591)

Exemple SetFontSize (Page 1560)

Alignement du texte

Exemple SetFontSize

SetFontBold

Fonction

Active ou désactive le style de police "gras".

Syntaxe

BOOL SetFontBold(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFontBold);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFontBold

Style de police "gras" activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetFontBold (Page 1559)

Exemple SetFontBold

SetFontItalic

Fonction

Active ou désactive le style de police "italique".

Syntaxe

BOOL SetFontItalic(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFontItalic);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFontItalic

Style de police "italique" activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetFontBold (Page 1559)

Exemple SetFontBold

SetFontName

Fonction

Règle un type de police.

Syntaxe

BOOL SetFontName(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, char* szFontName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

szFontName

Pointeur sur le nom du type de police.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetText (Page 1573)

Exemple SetText

SetFontSize

Fonction

Règle la taille de police.

Syntaxe

BOOL SetFontSize(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFontSize);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFontSize

Taille

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetFontSize (Page 1560)

Exemple SetFontSize

SetFontUnderline

Fonction

Active ou désactive le style de police "souligné".

Syntaxe

BOOL SetFontUnderline(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFontUnderline);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFontUnderline

Style de police "souligné" activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetFontBold (Page 1559)

Exemple SetFontBold

SetOrientation

Fonction

Détermine le sens du texte (vertical/horizontal).

Syntaxe

BOOL SetOrientation(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bOrientation);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bOrientation

Sens du texte

TRUE	vertical
FALSE	horizontal

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetFontBold (Page 1559)

Exemple SetFontBold

SetText

Fonction

Détermine la valeur de la propriété "Texte" pour des objets comme texte statique, case à cocher et case d'option.

Syntaxe

BOOL SetText(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, char* szText);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

szText

Pointeur sur un texte

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Pour les cases à cocher et les cases d'option, il faut déterminer l'élément à modifier via la fonction SetIndex avant l'appel de cette fonction.

Voir aussi

Exemple SetText (Page 1573)

Exemple SetText

geometry

Geometry - brève description

Les fonctions du groupe Geometry vous permettent de modifier ou d'interroger la taille, la position et d'autres propriétés géométriques d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetActualPointLeft

Fonction

Règle la valeur X du point actuel d'un polygone ou d'un trait polygonal.

Syntaxe

BOOL SetActualPointLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lActualPointLeft);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lActualPointLeft

Valeur X du point actuel d'un polygone ou d'un trait polygonal

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction SetIndex permet de régler le point actuel d'un polygone.

Voir aussi

Exemple SetLeft (Page 1561)

Exemple SetLeft

SetActualPointTop

Fonction

Règle la valeur Y du point actuel d'un polygone ou d'un trait polygonal.

Syntaxe

BOOL SetActualPointTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lActualPointTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lActualPointTop

Valeur Y du point actuel d'un polygone ou d'un trait polygonal

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

La fonction SetIndex permet de régler le point actuel d'un polygone.

Voir aussi

Exemple SetTop (Page 1574)

Exemple SetTop

SetBoxCount

Fonction

Détermine le nombre des champs dans une case à cocher ou une case d'option.

Syntaxe

BOOL SetBoxCount(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBoxCount);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBoxCount

Nombre des champs dans une case à cocher ou une case d'option

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetDirection

Fonction

Détermine l'orientation du bargraphe pour des objets de barre.

Syntaxe

BOOL SetDirection(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lDirection);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lDirection

Orientation du bargraphe sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Orientation du bargraphe (Page 1582)

Exemple SetTop (Page 1574)

Exemple SetTop

Orientation du bargraphe

SetEndAngle

Fonction

Paramètre l'angle final pour des segments de cercle et d'ellipse ainsi que pour des arcs de cercle et d'ellipse.

Syntaxe

BOOL SetEndAngle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lEndAngle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lEndAngle

Angle final pour des segments de cercle et d'ellipse ainsi que pour des arcs de cercle et d'ellipse.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTop (Page 1574)

Exemple SetTop

SetHeight

Fonction

Règle la hauteur du rectangle circonscrit d'un objet.

Syntaxe

BOOL SetHeight(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lHeight);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lHeight

Hauteur du rectangle circonscrit

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple SetHeight (Page 1560)

Exemple SetHeight

SetLeft

Fonction

Détermine la valeur X de l'angle gauche supérieur du rectangle circonscrit d'un objet.

Syntaxe

BOOL SetLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lLeft);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lLeft

Valeur X de l'angle gauche supérieur du rectangle circonscrit

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetLeft (Page 1561)

Exemple SetLeft

SetPointCount

Fonction

Règle le nombre de points d'angle d'un polygone ou d'un trait polygonal.

Syntaxe

BOOL SetPointCount(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lPointCount);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lPointCount

Nombre d'angles

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetLeft (Page 1561)

Exemple SetLeft

SetRadius

Fonction

Détermine le rayon d'un cercle, d'un segment de cercle ou d'un arc de cercle.

Syntaxe

BOOL SetRadius(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRadius);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRadius

Rayon

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHeight (Page 1560)

Exemple SetHeight

SetRadiusHeight

Fonction

Règle le rayon d'une ellipse, d'un segment d'ellipse ou d'un arc d'ellipse dans le sens vertical.

Syntaxe

BOOL SetRadiusHeight(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRadiusHeight);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRadiusHeight

Rayon dans le sens vertical

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHeight (Page 1560)

Exemple SetHeight

SetRadiusWidth

Fonction

Règle le rayon d'une ellipse, d'un segment d'ellipse ou d'un arc d'ellipse dans le sens horizontal.

Syntaxe

BOOL SetRadiusWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRadiusWidth);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRadiusWidth

Rayon dans le sens horizontal

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetWidth (Page 1574)

Exemple SetWidth

SetReferenceRotationLeft

Fonction

Règle la valeur X de la référence de rotation pour une ligne, un polygone et un trait polygonal (Point de rotation autour duquel l'objet peut être tourné).

Syntaxe

BOOL SetReferenceRotationLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lReferenceRotationLeft);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lReferenceRotationLeft

Valeur X de la référence de rotation

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetLeft (Page 1561)

Exemple SetLeft

SetReferenceRotationTop

Fonction

Règle la valeur Y de la référence de rotation pour une ligne, un polygone et un trait polygonal (Point de rotation autour duquel l'objet peut être tourné).

Syntaxe

BOOL SetReferenceRotationTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lReferenceRotationTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lReferenceRotationTop

Valeur Y de la référence de rotation

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTop (Page 1574)

Exemple SetTop

SetRotationAngle

Fonction

Règle l'angle de rotation autour du point de rotation pour une ligne, un polygone ou un trait polygonal.

Syntaxe

BOOL SetRotationAngle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRotationAngle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRotationAngle

Angle de rotation

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetLeft (Page 1561)

Exemple SetLeft

SetRoundCornerHeight

Fonction

Règle le rayon vertical de l'angle d'un rectangle arrondi.

Syntaxe

BOOL SetRoundCornerHeight(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRoundCornerHeight);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRoundCornerHeight

Rayon vertical

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHeight (Page 1560)

Exemple SetHeight

SetRoundCornerWidth

Fonction

Règle le rayon horizontal de l'angle d'un rectangle arrondi.

Syntaxe

BOOL SetRoundCornerWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRoundCornerWidth);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRoundCornerWidth

Rayon horizontal

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetWidth (Page 1574)

Exemple SetWidth

SetStartAngle

Fonction

Règle l'angle de départ pour des segments de cercle et d'ellipse ainsi que pour des arcs de cercle et d'ellipse.

Syntaxe

BOOL SetStartAngle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lStartAngle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lStartAngle

Angle de départ

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHeight (Page 1560)

Exemple SetHeight

SetTop

Fonction

Détermine la valeur Y de l'angle gauche supérieur du rectangle circonscrit d'un objet.

Syntaxe

BOOL SetTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lTop

Valeur Y de l'angle gauche supérieur du rectangle circonscrit

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTop (Page 1574)

Exemple SetTop

SetWidth

Fonction

Règle la largeur du rectangle circonscrit d'un objet.

Syntaxe

BOOL SetWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lWidth);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lWidth

Largeur du rectangle circonscrit

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple SetWidth (Page 1574)

Exemple SetWidth

SetZeroPoint

Fonction

Détermine l'origine pour des objets de barre.

Syntaxe

BOOL SetZeroPoint(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lZeroPoint);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lZeroPoint

Origine

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTop (Page 1574)

Exemple SetTop

i_o

i_o - brève description

Les fonctions du groupe i_o permettent de modifier ou d'interroger de différentes propriétés concernant les valeurs de sortie et d'entrée.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetAssumeOnExit

Fonction

Détermine, pour les champs E/S, si les valeurs entrées seront validées au moment de quitter le champ.

Syntaxe

BOOL SetAssumeOnExit(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bAssumeOnExit);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bAssumeOnExit

Valeurs sont validées au moment de quitter le champ oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHiddenInput (Page 1560)

Exemple SetHiddenInput

SetAssumeOnFull

Fonction

Détermine, pour les champs E/S, si les valeurs entrées seront validées lors de l'entrée complète.

Syntaxe

BOOL SetAssumeOnFull(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bAssumeOnFull);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bAssumeOnFull

Valeurs sont validées lors de l'entrée complète oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHiddenInput (Page 1560)

Exemple SetHiddenInput

SetBitNumber

Fonction

Détermine, dans le cas du type de liste "bit", le bit pertinent dans la valeur de sortie.

Syntaxe

BOOL SetBitNumber(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBitNumber);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBitNumber

Bit pertinent dans valeur de sortie dans le cas du type de liste "bit"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Types de liste (Page 1589)

Types de liste

SetClearOnError

Fonction

Règle, pour les champs E/S, si la suppression du contenu en cas d'entrée erronée est activée.

Syntaxe

BOOL SetClearOnError(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bClearOnError);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bClearOnError

Suppression de l'entrée en cas d'entrée erronée oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHiddenInput (Page 1560)

Exemple SetHiddenInput

SetClearOnNew

Fonction

Règle, pour les champs E/S, la suppression du contenu en cas de nouvelle entrée.

Syntaxe

BOOL SetClearOnNew(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bClearOnNew);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bClearOnNew

Suppression du contenu en cas de nouvelle entrée oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHiddenInput (Page 1560)

Exemple SetHiddenInput

SetHiddenInput

Fonction

Pilote l'entrée cachée pour les champs E/S

Syntaxe

BOOL SetHiddenInput(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bHiddenInput);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bHiddenInput

Entrée cachée oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetHiddenInput (Page 1560)

Exemple SetHiddenInput

SetNumberLines

Fonction

Détermine, pour l'objet "liste de textes", le nombre des lignes visibles.

Syntaxe

BOOL SetNumberLines(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lNumberLines);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lNumberLines

Nombre des lignes visibles

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si la quantité des textes configurées est supérieure au nombre des lignes visibles, l'objet "liste de textes" est complété par une barre de défilement verticale.

SetOutputValueChar

Fonction

Met, pour les champs E/S, un pointeur sur la valeur de sortie.

Syntaxe

BOOL SetOutputValueChar(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, char* szOutputValueChar);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

szOutputValueChar

Pointeur sur la valeur de sortie

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetOutputValueDouble

Fonction

Paramètre la valeur de sortie pour les champs E/S.

Syntaxe

BOOL SetOutputValueDouble(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dOutputValueDouble);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dOutputValueDouble

Valeur de sortie

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetOutputValueDouble (Page 1562)

Exemple SetOutputValueDouble

Limites

Limites - brève description

Les fonctions du groupe Limites permettent de modifier ou d'interroger de différentes propriétés concernant les valeurs limites.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetAlarmHigh

Fonction

Détermine le seuil d'alarme supérieur pour des objets de barre.

Syntaxe

BOOL SetAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dAlarmHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dAlarmHigh

Limite d'alarme supérieure

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetAlarmLow

Fonction

Détermine le seuil d'alarme inférieur pour des objets de barre.

Syntaxe

BOOL SetAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dAlarmLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dAlarmLow

Limite d'alarme inférieure

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetCheckAlarmHigh

Fonction

Pilote la surveillance du seuil d'alarme supérieur pour des objets de barre.

Syntaxe

BOOL SetCheckAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckAlarmHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckAlarmHigh

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckAlarmLow

Fonction

Pilote la surveillance du seuil d'alarme inférieur pour des objets de barre.

Syntaxe

BOOL SetCheckAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckAlarmLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckAlarmLow

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckLimitHigh4

Fonction

Pilote la surveillance de la valeur limite supérieur Réserve 4 pour des objets de barre.

Syntaxe

BOOL SetCheckLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckLimitHigh4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckLimitHigh4

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckLimitHigh5

Fonction

Pilote la surveillance de la valeur limite supérieur Réserve 5 pour des objets de barre.

Syntaxe

BOOL SetCheckLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckLimitHigh5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckLimitHigh5

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckLimitLow4

Fonction

Pilote la surveillance de la valeur limite inférieure Réserve 4 pour des objets de barre.

Syntaxe

BOOL SetCheckLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckLimitLow4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckLimitLow4

Surveillance oui/non.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckLimitLow5

Fonction

Pilote la surveillance de la valeur limite inférieure Réserve 5 pour des objets de barre.

Syntaxe

BOOL SetCheckLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckLimitLow5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckLimitLow5

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckToleranceHigh

Fonction

Pilote la surveillance du seuil de tolérance supérieur pour des objets de barre.

Syntaxe

```
BOOL SetCheckToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,
BOOL bCheckToleranceHigh);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckToleranceHigh

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckToleranceLow

Fonction

Pilote la surveillance du seuil de tolérance inférieur pour des objets de barre.

Syntaxe

BOOL SetCheckToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckToleranceLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckToleranceLow

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckWarningHigh

Fonction

Pilote la surveillance du seuil d'avertissement supérieur pour des objets de barre.

Syntaxe

BOOL SetCheckWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckWarningHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckWarningHigh

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetCheckWarningLow

Fonction

Pilote la surveillance du seuil d'avertissement inférieur pour des objets de barre.

Syntaxe

BOOL SetCheckWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCheckWarningLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCheckWarningLow

Surveillance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetColorAlarmHigh

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil d'alarme supérieur est atteint.

Syntaxe

BOOL SetColorAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorAlarmHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorAlarmHigh

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetColorAlarmLow

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil d'alarme inférieur est atteint.

Syntaxe

BOOL SetColorAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorAlarmLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorAlarmLow

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetColorLimitHigh4

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil supérieur Réserve 4 est atteint.

Syntaxe

BOOL SetColorLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorLimitHigh4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorLimitHigh4

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetColorLimitHigh5

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil supérieur Réserve 5 est atteint.

Syntaxe

BOOL SetColorLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorLimitHigh5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorLimitHigh5

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetColorLimitLow4

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil inférieur Réserve 4 est atteint.

Syntaxe

BOOL SetColorLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorLimitLow4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorLimitLow4

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetColorLimitLow5

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil inférieur Réserve 5 est atteint.

Syntaxe

BOOL SetColorLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorLimitLow5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorLimitLow5

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetColorToleranceHigh

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil de tolérance supérieur est atteint.

Syntaxe

BOOL SetColorToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorToleranceHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorToleranceHigh

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetColorToleranceLow

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil de tolérance inférieur est atteint.

Syntaxe

BOOL SetColorToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorToleranceLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorToleranceLow

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetColorWarningHigh

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil d'avertissement supérieur est atteint.

Syntaxe

BOOL SetColorWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorWarningHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorWarningHigh

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetColorWarningLow

Fonction

Détermine, pour les objets de barre, la couleur de barre lorsque le seuil d'avertissement inférieur est atteint.

Syntaxe

BOOL SetColorWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lColorWarningLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lColorWarningLow

Couleur de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s’est produite.

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetLimitHigh4

Fonction

Détermine, pour les objets de barre, le seuil supérieur pour Réserve 4.

Syntaxe

BOOL SetLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dLimitHigh4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dLimitHigh4

Seuil supérieur pour Réserve 4

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetLimitHigh5

Fonction

Détermine, pour les objets de barre, le seuil supérieur pour Réserve 5.

Syntaxe

BOOL SetLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dLimitHigh5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dLimitHigh5

Seuil supérieur pour Réserve 5

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetLimitLow4

Fonction

Détermine, pour les objets de barre, la valeur limite inférieure pour Réserve 4.

Syntaxe

BOOL SetLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dLimitLow4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dLimitLow4

Valeur limite inférieure pour Réserve 4

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetLimitLow5

Fonction

Détermine, pour les objets de barre, la valeur limite inférieure pour Réserve 5.

Syntaxe

BOOL SetLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dLimitLow5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dLimitLow5

Valeur limite inférieure pour Réserve 5

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetLimitMax

Fonction

Détermine le seuil supérieur pour les champs E/S.

Syntaxe

BOOL SetLimitMax(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dLimitMax);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dLimitMax

Seuil supérieur

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetLimitMin

Fonction

Détermine la valeur limite inférieure pour les champs E/S.

Syntaxe

BOOL SetLimitMin(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dLimitMin);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dLimitMin

Valeur limite inférieure

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetMarker

Fonction

Pilote, pour les objets de barre, l'affichage du marqueur de valeur limite.

Syntaxe

BOOL SetMarker(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bMarker);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bMarker

Marqueur de valeur limite activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetToleranceHigh

Fonction

Détermine le seuil de tolérance supérieur pour des objets de barre.

Syntaxe

BOOL SetToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dToleranceHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dToleranceHigh

Seuil de tolérance supérieur

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetToleranceLow

Fonction

Détermine le seuil de tolérance inférieur pour des objets de barre.

Syntaxe

BOOL SetToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dToleranceLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dToleranceLow

Seuil de tolérance inférieur

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetAlarmHigh (Page 1555)

Exemple SetAlarmHigh

SetTypeAlarmHigh

Fonction

Pilote, pour les objets de barre, si le seuil d'alarme supérieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeAlarmHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeAlarmHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeAlarmHigh

Limite d'alarme supérieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeAlarmLow

Fonction

Pilote, pour les objets de barre, si le seuil d'alarme inférieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeAlarmLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeAlarmLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeAlarmLow

Limite d'alarme inférieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeLimitHigh4

Fonction

Pilote, pour les objets de barre, si le seuil supérieur pour Réserve 4 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeLimitHigh4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeLimitHigh4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeLimitHigh4

Limite supérieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeLimitHigh5

Fonction

Pilote, pour les objets de barre, si le seuil supérieur pour Réserve 5 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeLimitHigh5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeLimitHigh5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeLimitHigh5

Limite supérieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeLimitLow4

Fonction

Pilote, pour les objets de barre, si le seuil inférieur pour Réserve 4 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeLimitLow4(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeLimitLow4);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeLimitLow4

Limite inférieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeLimitLow5

Fonction

Pilote, pour les objets de barre, si le seuil inférieur pour Réserve 5 est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeLimitLow5(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeLimitLow5);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeLimitLow5

Limite inférieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeToleranceHigh

Fonction

Pilote, pour les objets de barre, si le seuil de tolérance supérieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeToleranceHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeToleranceHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeToleranceHigh

Seuil de tolérance supérieur

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeToleranceLow

Fonction

Pilote, pour les objets de barre, si le seuil de tolérance inférieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeToleranceLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeToleranceLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeToleranceLow

Seuil de tolérance inférieur

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeWarningHigh

Fonction

Détermine, pour les objets de barre, si le seuil d'avertissement supérieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeWarningHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeWarningHigh

Limite d'avertissement supérieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetTypeWarningLow

Fonction

Pilote, pour les objets de barre, si le seuil d'avertissement inférieur est indiqué en pour cent ou en valeur absolue.

Syntaxe

BOOL SetTypeWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTypeWarningLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTypeWarningLow

Limite d'avertissement inférieure

TRUE	Indication en pour cent
FALSE	Indication en valeur absolue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetWarningHigh

Fonction

Détermine le seuil d'avertissement supérieur pour des objets de barre.

Syntaxe

BOOL SetWarningHigh(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dWarningHigh);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dWarningHigh

Limite d'avertissement supérieure

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

SetWarningLow

Fonction

Détermine le seuil d'avertissement inférieur pour des objets de barre.

Syntaxe

BOOL SetWarningLow(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dWarningLow);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dWarningLow

Limite d'avertissement inférieure

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetMarker (Page 1562)

Exemple SetMarker

lien

Lien - brève description

Les fonctions du groupe Lien permettent de créer ou d'interroger la liaison de variable d'une propriété.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetLink

Fonction

Création d'une liaison de variables de propriétés d'objets

Syntaxe

BOOL SetLink(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName, LPLINKINFO *pLink);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

pLink

Pointeur sur une structure du type : LINKINFO

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Définition de structure LINKINFO (Page 1598)

Exemple SetLink (Page 1561)

Définition de structure LINKINFO

Exemple SetLink

miscs

Miscs - brève description

Les fonctions du groupe Miscs vous permettent de modifier ou d'interroger de différentes propriétés d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetAverage

Fonction

Pilote, pour les objets de barre, le calcul de la moyenne.

Syntaxe

BOOL SetAverage(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bAverage);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bAverage

Calcul de moyenne oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetBoxType

Fonction

Détermine, pour un champ E/S, le type de champ (champ d'entrée, champ de sortie, champ d'entrée/sortie).

Syntaxe

BOOL SetBoxType(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBoxType);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBoxType

Type de champ

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur est survenue.

Voir aussi

Champ E/S, Type de champ (Page 1585)

Champ d'E/S, Type de champ

SetColorChangeType

Fonction

Détermine, pour des objets de barre, si le changement de couleur au cas où une valeur limite serait atteinte ne s'effectue que dans un segment de barre ou dans toute la barre.

Syntaxe

BOOL SetColorChangeType(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bColorChangeType);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bColorChangeType

Type du changement de couleur

TRUE	Changement de couleur ne s'effectue que dans un segment
FALSE	Changement de couleur s'effectue dans la barre entière

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetCursorControl

Fonction

Paramètre la commande du curseur pour les champs E/S

Syntaxe

BOOL SetCursorControl(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCursorControl);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCursorControl

Commande du curseur activée/désactivée

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetCursorMode

Fonction

Paramètre la commande du curseur pour des vues

Syntaxe

BOOL SetCursorMode(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bCursorMode);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bCursorMode

Mode curseur

TRUE	Pointeur
FALSE	Curseur alpha

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetCursorMode (Page 1557)

Exemple SetCursorMode

SetEditAtOnce

Fonction

Paramètre, pour les champs E/S, si la propriété "Entrée immédiate" est activée.

Syntaxe

BOOL SetEditAtOnce(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bEditAtOnce);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bEditAtOnce

Entrée immédiate oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetExtendedOperation

Fonction

Pilote, pour les objets Slider, la propriété "Commande étendue".

Syntaxe

BOOL SetExtendedOperation(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bExtendedOperation);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bExtendedOperation

Commande étendue oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetHysteresis

Fonction

Pilote, pour les objets de barre, si l'affichage s'effectue avec ou sans hystérésis.

Syntaxe

BOOL SetHysteresis(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bHysteresis);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bHysteresis

Affichage avec/sans hystérésis

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetHysteresisRange

Fonction

Paramètre, pour des objets de barre, la valeur de l'hystérésis à l'affichage.

Syntaxe

BOOL SetHysteresisRange(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dHysteresisRange);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dHysteresisRange

Valeur de l'hystérésis

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetMax

Fonction

Paramètre la valeur maximale pour des objets de barre ou des objets Slider.

Syntaxe

BOOL SetMax(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dMax);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dMax

Valeur maximale

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetMin

Fonction

Paramètre la valeur minimale pour des objets de barre ou des objets Slider.

Syntaxe

BOOL SetMin(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dMin);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dMin

Valeur minimale

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetOffsetLeft

Fonction

Paramètre, pour des fenêtres de vue, la distance horizontale de la vue du bord de fenêtre gauche.

Syntaxe

BOOL SetOffsetLeft(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lOffsetLeft);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lOffsetLeft

Distance de la vue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetOffsetTop

Fonction

Paramètre, pour des fenêtres de vue, la distance verticale de la vue du bord de fenêtre supérieur.

Syntaxe

BOOL SetOffsetTop(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lOffsetTop);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lOffsetTop

Distance de la vue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetOperation

Fonction

Commande la manipulabilité d'objets.

Syntaxe

BOOL SetOperation(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bOperation);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bOperation

Objet manipulable oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne l'objet de vue, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetOperationMessage

Fonction

Pilote la sortie d'une alarme lors de la commande des objets "Champ E/S", "Case à cocher", "Case d'option" et "Slider".

Syntaxe

BOOL SetOperationMessage(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bOperationMessage);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bOperationMessage

Sortie d'une alarme lors de commande oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetOperationReport

Fonction

Pilote la journalisation de la raison de la manipulation pour tous les objets, excepté les fenêtres d'application et de vue ainsi que OLE-Control.

Syntaxe

BOOL SetOperationReport(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bOperationReport);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bOperationReport

Journalisation de la raison de manipulation oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne l'objet de vue, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetPasswordLevel

Fonction

Paramètre le niveau d'autorisation pour la manipulation d'objets pour tous les objets, excepté les fenêtres d'application et de vue ainsi que OLE-Control.

Syntaxe

BOOL SetPasswordLevel(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lPasswordLevel);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lPasswordLevel

Niveau d'autorisation

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

SetPictureName

Fonction

Détermine le nom de la vue devant être affichée dans une fenêtre de vue ou dans un objet graphique.

Syntaxe

BOOL SetPictureName(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, char* szPictureName);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de la fenêtre de vue ou de l'objet graphique

szPictureName

Pointeur sur le nom de la vue

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetPictureName (Page 1563)

Exemple SetPictureName

SetProcess

Fonction

Détermine, pour les objets de barre et les objets Slider, le paramétrage par défaut pour la valeur à afficher.

Détermine les champs sélectionnés pour des cases à cocher et des cases d'option.

Syntaxe

BOOL SetProcess(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dProcess);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dProcess

- En Runtime, cette valeur est utilisée pour des objets de barre et des objets Slider si, lors de l'activation de la vue, la variable correspondante n'est ni raccordée, ni mise à jour.
- Les champs sélectionnés sont déterminés pour des cases à cocher et des cases d'option. Chaque champ est représenté par un bit dans le mot de 32 bits (Champ1 correspond à la valeur indicative Bit 0). Les champs sélectionnés sont marqués par un bit positionné. Des champs non existants sont assignés par 0.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetSmallChange

Fonction

Paramètre pour des objets Slider le nombre de pas qu'effectue le curseur lors d'un clic de souris.

Syntaxe

BOOL SetSmallChange(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lSmallChange);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lSmallChange

Nombre de pas de paramétrage

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetTagPrefix

Fonction

Cette fonction active, pour des fenêtres de vue, le préfixe de variable d'une fenêtre de vue :

Dans une fenêtre de vue, la variable "Temperatur" est demandée pour un objet. Si un préfixe de variable "Moteur1." est affecté à la fenêtre de vue, la variable "Moteur1.Température" sera alors demandée.

L'activation d'un préfixe de variable ne sera valide que lorsque le nom de vue est affecté de nouveau.

Vous devez donc soit mettre le préfixe avant la sélection de vue, soit affecter le nom de vue de nouveau, au cas où la vue ne serait pas changée.

Syntaxe

BOOL SetTagPrefix(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,char* szTagPrefix);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

szTagPrefix

Préfixe de variable à activer

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si le préfixe de variable est activé pour une fenêtre de vue, le préfixe de variable précédera toutes les variables contenues dans la vue à afficher. Cela est valable également lorsque la demande s'effectue dans une fonction. Si une variable doit être lue sans préfixe de variable, vous devez faire précéder le nom de la variable par le complément "@NOTP::".

La technique à vues modulables est fortement simplifiée en utilisant un préfixe de variable.

Voir aussi

Exemple SetTagPrefix (Page 1570)

Exemple SetTagPrefix

SetTrend

Fonction

Pilote, pour les objets de barre, l'affichage de la tendance.

Syntaxe

BOOL SetTrend(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bTrend);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bTrend

Afffichage de la tendance oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetVisible

Fonction

Commande l'affichage d'un objet.

Syntaxe

BOOL SetVisible(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bVisible);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bVisible

Affichage d'objet oui/non

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Exemple SetVisible (Page 1574)

Exemple SetVisible

SetZeroPointValue

Fonction

Paramètre, pour les objets de barre, la valeur absolue de l'origine.

Syntaxe

BOOL SetZeroPointValue(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, double dZeroPointValue);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

dZeroPointValue

Valeur absolue de l'origine

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetZoom

Fonction

Paramètre, pour une fenêtre de vue, le facteur de mise à l'échelle.

Syntaxe

BOOL SetZoom(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lZoom);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lZoom

Facteur de mise à l'échelle

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

ole_control

OLE_control - brève description

Les fonctions du groupe ole_Control sont seulement applicables aux objets OCX-Slider.

A l'aide de ces fonctions vous pouvez modifier ou interroger les différentes propriétés et paramétrages d'un objet OCX-Slider.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetPosition

Fonction

Paramètre position de la coulisse du curseur de l'objet OCX Slider.

Syntaxe

BOOL SetPosition(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lPosition);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lPosition

Position de la coulisse du curseur de l'objet OCX Slider

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetPosition (Page 1564)

Exemple SetPosition

SetRangeMax

Fonction

Détermine la plage de réglage "Max" pour l'objet OCX Slider.

Syntaxe

BOOL SetRangeMax(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRangeMax);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRangeMax

Plage de réglage "Max" pour l'objet OCX Slider

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetRangeMax (Page 1565)

Exemple SetRangeMax

SetRangeMin

Fonction

Détermine la plage de réglage "Min" pour l'objet OCX Slider.

Syntaxe

BOOL SetRangeMin(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lRangeMin);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lRangeMin

Plage de réglage "Min" pour l'objet OCX Slider

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetRangeMin (Page 1565)

Exemple SetRangeMin

pictures

Pictures - brève description

Les fonctions du groupe Pictures vous permettent de modifier ou d'interroger de différentes propriétés des vues d'objets graphiques ou de boutons ronds.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetPicDeactTransparent

Fonction

Détermine pour les boutons ronds la couleur transparente pour l'état "désactivé".

Syntaxe

BOOL SetPicDeactTransparent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lPicDeactTransparent);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lPicDeactTransparent

Couleur transparente pour l'état "désactivé"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetPicDeactUseTransColor

Fonction

Pilote pour les boutons ronds la couleur transparente pour l'état "désactivé".

Syntaxe

```
BOOL SetPicDeactUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,
BOOL bPicDeactUseTransColor);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bPicDeactUseTransColor

Couleur transparente oui/non.

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetPicDownTransparent

Fonction

Détermine pour les boutons ronds la couleur transparente pour l'état "activé/enfoncé".

Syntaxe

BOOL SetPicDownTransparent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lPicDownTransparent);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lPicDownTransparent

Couleur transparente pour l'état "activé/enfoncé"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetPicDownUseTransColor

Fonction

Pilote pour les boutons ronds la couleur transparente pour l'état "activé/enfoncé".

Syntaxe

BOOL SetPicDownUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bPicDownUseTransColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bPicDownUseTransColor

Couleur transparente oui/non.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetPicTransColor

Fonction

Détermine la couleur transparente de la vue d'arrière-plan d'un objet graphique.

Syntaxe

BOOL SetPicTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lPicTransColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lPicTransColor

Couleur transparente de la vue d'arrière-plan

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetPictureDeactivated

Fonction

Détermine pour les boutons ronds le nom de vue pour l'état "désactivé".

Syntaxe

BOOL SetPictureDeactivated(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, char* szPictureDeactivated);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

szPictureDeactivated

Nom de vue pour l'état "désactivé"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Des fichiers Bitmap (*.bmp, *.dib) ainsi que des métafichiers (*.emf, *.wmf) peuvent être intégrés.

Voir aussi

Exemple SetPictureDown (Page 1563)

Exemple SetPictureDown

SetPictureDown

Fonction

Détermine pour les boutons ronds le nom de vue pour l'état "activé/enfoncé".

Syntaxe

BOOL SetPictureDown(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, char* szPictureDown);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

szPictureDown

Nom de vue pour l'état "activé/enfoncé"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Des fichiers Bitmap (*.bmp, *.dib) ainsi que des métafichiers (*.emf, *.wmf) peuvent être intégrés.

Voir aussi

Exemple SetPictureDown (Page 1563)

Exemple SetPictureDown

SetPictureUp

Fonction

Détermine pour les boutons ronds le nom de vue pour l'état "désactivé/non enfoncé".

Syntaxe

BOOL SetPictureUp(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, char* szPictureUp);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

szPictureUp

Nom de vue pour l'état "désactivé/non enfoncé".

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Des fichiers Bitmap (*.bmp, *.dib) ainsi que des métafichiers (*.emf, *.wmf) peuvent être intégrés.

Voir aussi

Exemple SetPictureUp (Page 1563)

Exemple SetPictureUp

SetPicUpTransparent

Fonction

Détermine pour les boutons ronds la couleur transparente pour l'état "désactivé/non enfoncé".

Syntaxe

BOOL SetPicUpTransparent(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lPicUpTransparent);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lPicUpTransparent

Couleur transparente pour l'état "désactivé/non enfoncé".

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetPicUpUseTransColor

Fonction

Pilote pour les boutons ronds la couleur transparente pour l'état "désactivé/non enfoncé".

Syntaxe

```
BOOL SetPicUpUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,
BOOL bPicUpUseTransColor);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bPicUpUseTransColor

Couleur transparente oui/non.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetPicUseTransColor

Fonction

Pilote la couleur transparente de la vue d'arrière-plan d'un objet graphique.

Syntaxe

```
BOOL SetPicUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,
BOOL bPicUseTransColor);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bPicUseTransColor

Couleur transparente oui/non.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

property

Property - brève description

Les fonctions du groupe Property vous permettent de modifier ou d'interroger des propriétes d'objets qui ne possèdent pas de propre fonction.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetPropBOOL

Fonction

Détermine une propriété avec la valeur "bValue".

Syntaxe

BOOL SetPropBOOL(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName, BOOL bValue)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

bValue

Valeur dans le format de données BOOL

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction se réfère à l'objet d'image, le paramètre lpszObjectName = NULL doit être défini.

Voir aussi

Exemple SetPropBOOL (Page 1564)

SetPropChar

Fonction

Détermine une propriété avec la valeur sur laquelle est mis le pointeur "szValue".

Syntaxe

BOOL SetPropChar(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName, char* szValue)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

szValue

Pointeur sur la valeur

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction se réfère à l'objet d'image, le paramètre lpszObjectName = NULL doit être défini.

Voir aussi

Exemple GetPropChar (Page 1531)

SetPropDouble

Fonction

Détermine une propriété avec la valeur "dValue".

Syntaxe

BOOL SetPropDouble(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName, double dValue)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

dValue

Valeur dans le format de données "double".

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction se réfère à l'objet d'image, le paramètre lpszObjectName = NULL doit être défini.

SetPropWord

Fonction

Détermine une propriété avec la valeur "lValue".

Syntaxe

BOOL SetPropWord(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, LPCTSTR lpszPropertyName, long lValue)

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lpszPropertyName

Nom d'une propriété d'objet

lValue

Valeur dans le format de données "long".

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction se réfère à l'objet d'image, le paramètre lpszObjectName = NULL doit être défini.

state

State - brève description

Les fonctions du groupe State vous permettent de modifier ou d'interroger de différentes propriétés d'indicateurs d'état.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetBasePicTransColor

Fonction

Détermine dans l'indicateur d'état la couleur transparente de la vue de base.

Syntaxe

BOOL SetBasePicTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBasePicTransColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBasePicTransColor

Couleur transparente de la vue de base

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Tableau de couleurs

Exemple SetBackColor

SetBasePicUseTransColor

Fonction

Pilote la couleur transparente de la vue de base d'un indicateur d'état.

Syntaxe

```
BOOL SetBasePicUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName,
BOOL bBasePicUseTransColor);
```

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bBasePicUseTransColor

Couleur transparente oui/non.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetFlashFlashPicture

Fonction

Règle pour l'indicateur d'état si la vue clignotante est animée de façon dynamique ou statique.

Syntaxe

BOOL SetFlashFlashPicture(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFlashFlashPicture);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFlashFlashPicture

Type de la vue clignotante

TRUE	Vue clignotante animée de façon dynamique
FALSE	Vue clignotante animée de façon statique

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetFlashPicTransColor

Fonction

Détermine pour l'indicateur d'état la couleur transparente de la vue clignotante.

Syntaxe

BOOL SetFlashPicTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFlashPicTransColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFlashPicTransColor

Couleur transparente de la vue clignotante

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Cette fonction n'est valable que pour des graphiques Bitmap (*.bmp).

Voir aussi

Tableau de couleurs (Page 1585)

Exemple SetBackColor (Page 1555)

Exemple SetBackColor

Tableau de couleurs

SetFlashPicUseTransColor

Fonction

Pilote pour un indicateur d'état la couleur transparente de la vue clignotante.

Syntaxe

BOOL SetFlashPicUseTransColor(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bFlashPicUseTransColor);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bFlashPicUseTransColor

Couleur transparente oui/non.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetFlashRateFlashPic

Fonction

Détermine pour l'indicateur d'état la fréquence de clignotement de la vue clignotante.

Syntaxe

BOOL SetFlashRateFlashPic(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFlashRateFlashPic);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFlashRateFlashPic

Fréquence de clignotement de la vue clignotante

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence dépend du système et du matériel (nombre d'objets, processeur, mémoire, cycle de rafraîchissement, etc.).

Voir aussi

Fréquences de clignotement (Page 1583)

Exemple SetFlashRateFlashPic (Page 1559)

Fréquences de clignotement

SetIndex

Fonction

Met l'indice d'un polygone ou d'un trait polygonal et régle ainsi le point actuel de l'objet.

Syntaxe

BOOL SetIndex(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lIndex);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lIndex

Valeur d'indice

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

style

Style - brève description

Les fonctions du groupe Style permettent de modifier ou d'interroger de différentes propriétés concernant la présentation d'objets.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

SetBackBorderWidth

Fonction

Règle la largeur de l'encadrement de bordures 3D et d'objets Slider.

Syntaxe

BOOL SetBackBorderWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBackBorderWidth);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBackBorderWidth

Largeur de l'encadrement en pixels.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetBorderStyle (Page 1556)

Exemple SetBorderStyle

SetBorderEndStyle

Fonction

Règle le type de la fin de ligne.

Syntaxe

BOOL SetBorderEndStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderEndStyle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderEndStyle

Type de la fin de ligne sous forme de valeur en chiffre.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Fins de trait (Page 1589)

Exemple SetBorderEndStyle (Page 1555)

Exemple SetBorderEndStyle

Fins de trait

SetBorderStyle

Fonction

Règle le type de ligne ou de bordure.

Syntaxe

BOOL SetBorderStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderStyle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderStyle

Type de ligne ou de bordure sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Types de trait (Page 1588)

Exemple SetBorderStyle (Page 1556)

Exemple SetBorderStyle

Types de trait

SetBorderWidth

Fonction

Règle la largeur des lignes ou de la ligne de bordure.

Syntaxe

BOOL SetBorderWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBorderWidth);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBorderWidth

Largeur de ligne ou largeur de la ligne de bordure

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetBorderStyle (Page 1556)

Exemple SetBorderStyle

SetBoxAlignment

Fonction

Règle la disposition des éléments de commande (cadrés à gauche ou à droite) dans des cases à cocher ou des cases d'option.

Syntaxe

BOOL SetBoxAlignment(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lBoxAlignment);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lBoxAlignment

Disposition des éléments de commande

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Alignement d'élément dans des cases à cocher et des cases d'option (Page 1585)

Exemple SetBorderStyle (Page 1556)

Exemple SetBorderStyle

Alignement d'élément dans des cases à cocher et des cases d'option

SetFillStyle

Fonction

Règle le type du motif de rempl.

Syntaxe

BOOL SetFillStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFillStyle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFillStyle

Type du motif de rempl. sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Remarque

Si l'appel de la fonction concerne la vue complète, le paramètre lpszObjectName = NULL doit être réglé.

Voir aussi

Motif de remplissage (Page 1587)

Exemple SetFillStyle (Page 1558)

Motif de remplissage

Exemple SetFillStyle

SetFillStyle2

Fonction

Règle, pour un affichage de barre, le motif de remplissage de la barre.

Syntaxe

BOOL SetFillStyle2(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lFillStyle2);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lFillStyle2

Motif de rempl. de la barre sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Motif de remplissage (Page 1587)

Exemple SetFillStyle (Page 1558)

Motif de remplissage

Exemple SetFillStyle

SetItemBorderStyle

Fonction

Détermine le type de ligne de séparation pour l'objet "liste de textes".

Syntaxe

BOOL SetItemBorderStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lItemBorderStyle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lItemBorderStyle

Type de ligne de séparation sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Types de trait (Page 1588)

Exemple SetBorderStyle (Page 1556)

Types de trait

Exemple SetBorderStyle

SetItemBorderWidth

Fonction

Règle la largeur de ligne de séparation pour l'objet "liste de textes".

Syntaxe

BOOL SetItemBorderWidth(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, long int lItemBorderWidth);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

lItemBorderWidth

Largeur de ligne de sép. sous forme de valeur en chiffre

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetBorderStyle (Page 1556)

Exemple SetBorderStyle

SetPressed

Fonction

Règle, pour les boutons ou bouton ronds, si la position du bouton est "enfoncé" ou "non enfoncé".

Syntaxe

BOOL SetPressed(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bPressed);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bPressed

Position du bouton

TRUE	Position du bouton "enfoncé"
FALSE	Position du bouton "non enfoncé"

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetToggle

Fonction

Règle pour des boutons ou boutons ronds si le bouton peut être bloqué ou pas.

Syntaxe

BOOL SetToggle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bToggle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bToggle

Bouton peut/ne peut pas être bloqué

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

SetWindowsStyle

Fonction

Règle pour les boutons s'il sont représentés dans le style Windows.

Syntaxe

BOOL SetWindowsStyle(LPCTSTR lpszPictureName, LPCTSTR lpszObjectName, BOOL bWindowStyle);

Paramètres

lpszPictureName

Nom de la vue

lpszObjectName

Nom de l'objet

bWindowStyle

"Style Windows" activé/désactivé

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

OpenHomePicture

Fonction

Ouvre la vue d'accueil enregistrée

Syntaxe

BOOL OpenHomePicture();

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

OpenNextPicture

Fonction

WinCC enregistre les noms des vues ayant été ouvertes au moment de l'exécution par l'utilisateur ainsi que l'ordre dans lequel sont ouvertes ces vues.

Le nombre maximal de noms de vue enregistrés de cette façon est réglable dans WinCC Explorer, dans les propriétés de l'ordinateur sur l'onglet "Graphics Runtime", sous "Taille du tampon de vues".

La fonction OpenNextPicture ouvre maintenant la vue ayant été ouverte avant le dernier appel de OpenPrevPicture.

Syntaxe

BOOL OpenNextPicture();

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

OpenPrevPicture

Fonction

WinCC enregistre les noms des vues ayant été ouvertes au moment de l'exécution par l'utilisateur ainsi que l'ordre dans lequel sont ouvertes ces vues.

Le nombre maximal de noms de vue enregistrés de cette façon est réglable dans WinCC Explorer, dans les propriétés de l'ordinateur sur l'onglet "Graphics Runtime", sous "Taille du tampon de vues".

La fonction OpenPrevPicture ouvre maintenant la vue ayant été ouverte avant la vue ouverte actuellement.

Syntaxe

BOOL OpenPrevPicture();

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

OpenStoredPicture

Fonction

Ouvre la vue qui a été enregistrée avec la fonction StorePicture.

Syntaxe

BOOL OpenStoredPicture();

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

StorePicture

Fonction

Enregistre la vue actuelle; cette vue enregistrée peut être ouverte avec la fonction OpenStoredPicture.

Syntaxe

BOOL StorePicture();

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

2.15.3.5 tag

tag - brève description

Les fonctions du groupe Tag vous permettent de mettre ou d'interroger des variables.

GetTag ou GetTagWait ?

Les variables de process demandées avec GetTag sont intégrées dans une image. Etant donné que le rafraîchissement et la lecture de l'image sont deux processus se déroulant séparément, l'appel de GetTag n'est pas directement influencé par le couplage. Il peut ainsi être exécuté plus rapidement et plus indépendemment qu'un appel GetTagWait.

Les variables de process demandées avec GetTagWait ne sont pas intégrées dans l'image. Un appel GetTagWait lit explicitement la valeur depuis l'AS. Cela implique toujours un aller-retour via le couplage et le temps de réaction de l'AS. Pendant l'exécution, le traitement des actions en C est bloqué et le temps que durera cet appel n'est pas prévisible. Cette durée s'allonge si plusieurs variables sont lues.

Un appel GetTagWait est nécessaire si

- des processus d'écriture / de lecture rapides doivent être synchronisés
- une valeur est lue explicitement dans l'AS
- ou bien si une connexion dans l'image doit être sciemment évitée.

Evitez l'appel GetTagWait dans les actions cycliques en C étant donné qu'il s'agit là de la cause essentielle des problèmes de performance.

SetTag ou SetTagWait ?

L'appel SetTag attribue un travail d'écriture sans attendre de confirmation de l'AS.

L'appel SetTagWait attribue un travail d'écriture et attend une confirmation de l'AS. Cela implique toujours un aller-retour via le couplage et le temps de réaction de l'AS. Pendant l'exécution, le traitement des actions en C est bloqué et le temps que durera cet appel n'est pas prévisible. Cette durée s'allonge si plusieurs variables sont écrites.

Un appel SetTagWait est utilisé pour garantir que la valeur a été écrite avant que ne se poursuive l'action en C. Evitez l'appel SetTagWait dans les actions cycliques en C.

Remarque

La différence entre GetTag et GetTagWait existe également pour les variables internes. Ici, la différence est cependant moins significative étant donné qu'il n'y a pas de couplage. Pour synchroniser des processus d'écriture / de lecture rapides, il est nécessaire d'utiliser la fonction Wait respective également pour les variables internes.

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

---

get

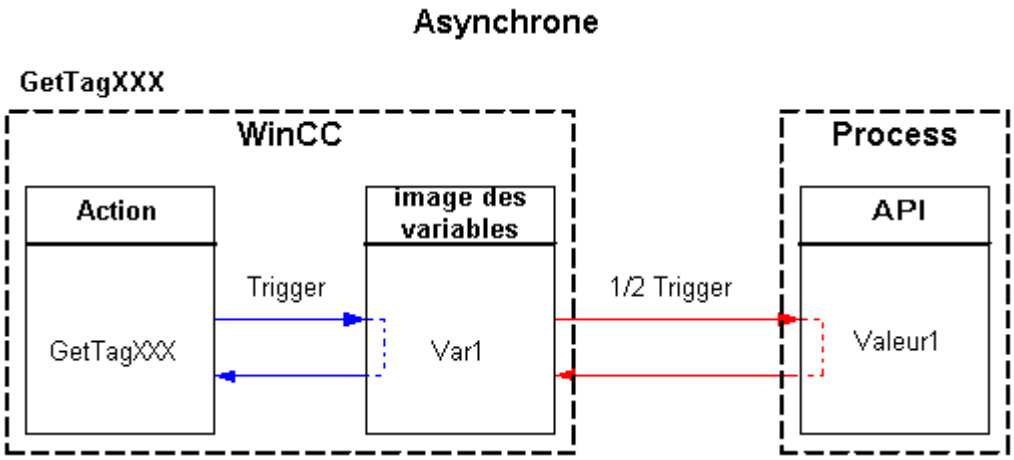
Mode de fonctionnement des fonctions GetTag

GetTagXXX

Par l'appel de la fonction, la variable est connectée et, à partir de là, elle est constamment demandée de façon cyclique depuis l'AS. Le cycle de connexion y dépend du Trigger (voir la description ci-après). Pour les appels GetTagXXX, la valeur se trouvant dans WinCC est fournie. En cas de désélection de vue, les variables sont à nouveau déconnectées.

L'appel se caractérise ainsi de la façon suivante :

- La valeur est lue depuis l'image de variable de WinCC.
- L'appel est plus vite comparé à GetTagXXXWait (sauf lors du premier appel, celui dure en principe plus longtemps étant donné que la valeur doit être lue depuis l'AS et ensuite connectée).
- La durée de l'appel ne dépend pas de la charge du bus ou de l'AS.
- La fonction ne fournit aucune information concernant l'état des variables



Remarque

Si une variable est demandée dans une action Global Script, elle reste connectée pendant toute la durée d'exécution Runtime de WinCC.

Dans les fonctions Callback, il faut utiliser la fonction GetTagXXXWait correspondante.

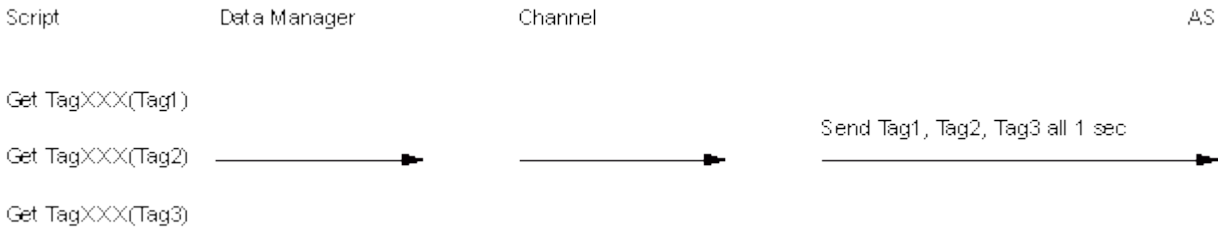
Comportement dans les actions avec Trigger de variable (recommandé) :

Toutes les variables contenues dans le Trigger de variable sont déjà connues lors de la sélection de vues et sont connectées avec le temps de surveillance spécifié.

Etant donné que toutes les variables sont demandées à la fois, l'optimisation la meilleure possible peut être obtenue du canal. Si une variable contenue dans le Trigger est exigée avec GetTagXXX() dans une action en C, la valeur existe déjà et est directement transférée à l'appel (performant).

Déclaration de variables dans les actions avec déclenchement par variable

Comme les variables sont déjà connues à la sélection de la vue, elles peuvent être transférées au gestionnaire de données par une tâche et être déclarées au canal de manière groupée.



Remarque

Si une variable n'existant pas dans le Trigger est demandée, le comportement correspond à celui pour le Trigger standard.

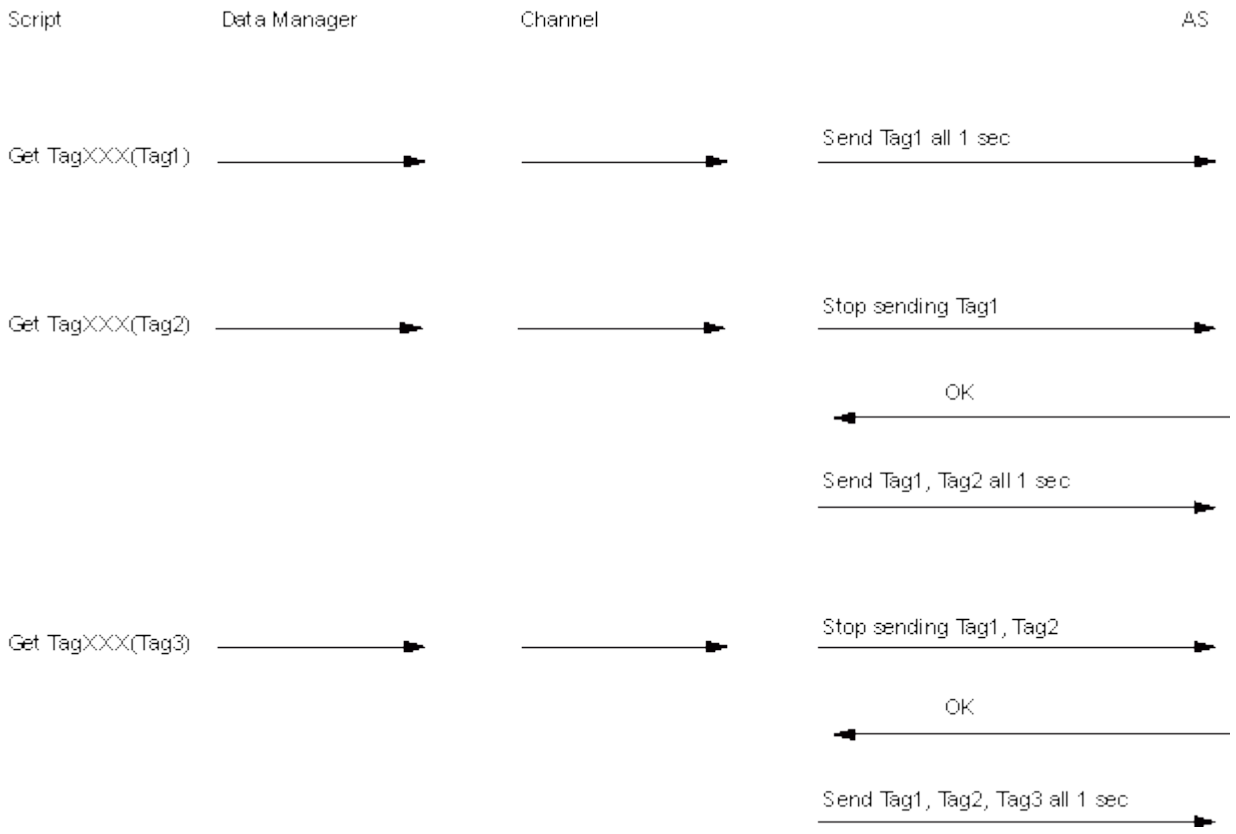
Comportement dans des actions avec Trigger standard :

Lors du premier appel, la variable est connectée avec la moitié du temps de cycle. La valeur est alors présente à chaque nouvel appel.

Déclaration de variables dans les actions avec déclenchement standard et déclenchement par événement

Ce n'est qu'à l'exécution des différentes actions que les variables requises dans la vue sont identifiées. Les variables dans les nombreuses tâches individuelles sont ainsi déclarées au canal. Lors de la sélection d'une vue avec des actions cycliques, la communication peut être soumise à une forte charge par la réorganisation permanente.

Exemple : Le canal prend en charge sa propre formation de cycle. En général, la formation de cycle par le canal est effectuée directement par l'AS. Les ressources pour ces cycles sont limitées par l'AS. Par conséquent, le canal stoppe les tâches actuelles pour ce cycle et configure à nouveau le cycle dans l'AS.



Comportement dans des actions déclenchées par des événements :

Lors du premier appel, la variable est connectée dans le mode "En cas de modification". Les variables de process qui sont connectées dans le mode "En cas de modification", correspondent à un travail de lecture cyclique ayant un temps de cycle de 1s.

Remarque

Si une valeur est demandée avec un événement, p. ex. un "clic de souris" par GetTagXXX(), la variable est intégrée dans l'image de variable. La variable est demandée cycliquement à partir de ce moment depuis l'AS, ce qui augmente la charge de base.

Pour éviter cette augmentation de la charge de base, la valeur peut être demandée par GetTagXXXWait(). L'appel GetTagXXXWait() n'entraîne certes qu'une seule fois une augmentation de la charge de communication, mais la variable n'est pas intégrée dans l'image de variable.

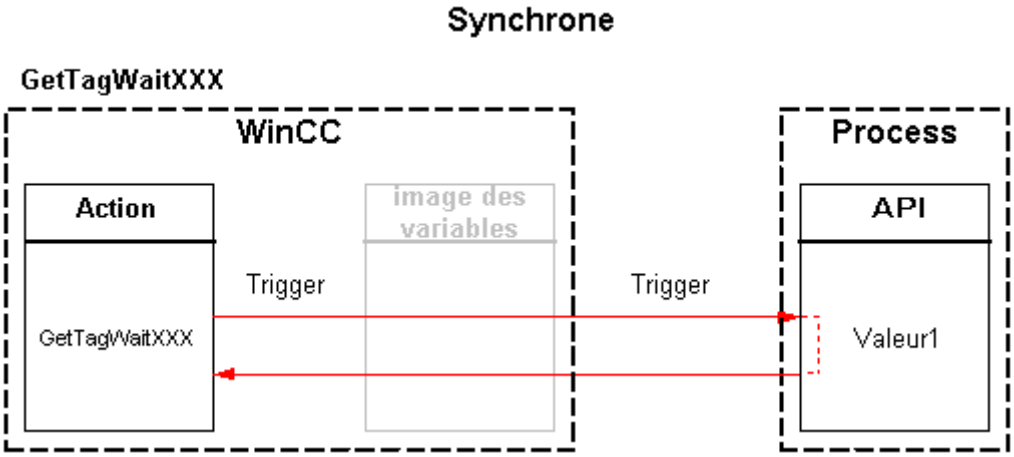
GetTagXXXWait

La fonction renvoie la valeur actuelle. La variable n'est pas connectée cycliquement, mais la valeur est demandée une seule fois depuis l'AS.

L'appel se caractérise ainsi de la façon suivante :

- La valeur est lue explicitement dans l'AS.
- L'appel dure plus longtemps que GetTagXXX.

- La durée de l'appel dépend entre autres de la charge du bus et de l'AS.
- La fonction ne fournit aucune information concernant l'état de la variable.



GetTagXXXState

La fonction GetTagXXXState possède les mêmes caractéristiques que GetTagXXX, mais fournit en plus des informations concernant l'état de la variable. Etant donné que l'état est toujours donné sur le plan interne, on ne peut constater aucune différence avec la performance de GetTagXXX.

GetTagXXXStateWait

La fonction GetTagXXXStateWait possède les mêmes caractéristiques que GetTagXXXWait, mais fournit en plus des informations concernant l'état de la variable. Etant donné que l'état est toujours donné sur le plan interne, on ne peut constater aucune différence avec la performance de GetTagXXXWait.

La différence entre les fonctions GetTagXXXStateWait et GetTagXXXState correspond à la différence entre GetTagXXXWait et GetTagXXX. Dans le cas des variables de process, étant donné que la lecture se fait explicitement depuis l'AS, il se peut que la valeur ou l'état soient plus actuels que pour GetTagXXXState.

GetTagXXXStateQC

La fonction GetTagXXXStateQC possède les mêmes caractéristiques que GetTagXXXState, mais fournit en plus des informations concernant le code qualité de la variable.

GetTagXXXStateQCWait

La fonction GetTagXXXStateQCWait possède les mêmes caractéristiques que GetTagXXXStateWait, mais fournit en plus des informations concernant le code qualité de la variable.

GetTagMultiWait

La fonction GetTagMultiWait possède les mêmes caractéristiques que GetTagXXXWait. Elle offre toutefois la possibilité de demander plusieurs variables regroupées dans un seul travail. Cela permet dans la plupart des cas d'optimiser les travaux de lecture vers l'AS, de sorte qu'un seul travail est envoyé à l'AS.

GetTagMultiStateWait

La fonction GetTagMultiStateWait possède les mêmes caractéristiques que GetTagMultiWait, mais fournit en plus des informations concernant les états des variables.

GetTagMultiStateQCWait

La fonction GetTagMultiStateQCWait possède les mêmes caractéristiques que GetTagMultiStateWait, mais fournit en plus des informations concernant le code qualité de la variable.

state

wait

getTagBitStateWait

Fonction

Calcule la valeur d'une variable du type de données "Variable binaire". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

BOOL GetTagBitStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "BOOL"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagBitStateWait (Page 1537)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Exemple GetTagBitStateWait

Mode de fonctionnement des fonctions GetTag

GetTagByteStateWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 8 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

BYTE GetTagByteStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "OCTET"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateWait (Page 1550)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateWait

GetTagCharStateWait

Fonction

Calcule la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

char* GetTagCharStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur en retour

Pointeur sur la valeur des variables dans le type de données "char".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable (Page 1591)

Exemple GetTagCharStateWait (Page 1539)

Exemple GetTagCharStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagDoubleStateWait

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

double GetTagDoubleStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "double"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagFloatStateWait (Page 1541)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagFloatStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagDWordStateWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 32 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

DWORD GetTagDWordStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "DWORD"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateWait (Page 1550)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWordStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagFloatStateWait

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

```
float GetTagFloatStateWait(Tag Tag_Name, PDWORD lp_dwstate);
```

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "float"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagFloatStateWait (Page 1541)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagFloatStateWait

Etats de variable

Mode de fonctionnement des fonctions GetTag

GetTagMultiStateWait

Fonction

Les valeurs et états de plusieurs variables sont déterminés et écrits à l'adresse correspondante dans le format indiqué. Les valeurs sont lues explicitement dans l'AS.

Il est nécessaire de transmettre à la fonction une 'DWORD Array' dans les membres de laquelle se trouvent les états des diverses variables après appel de la fonction. Sélectionnez cette 'Array' suffisamment grande pour que ces états disposent d'un espace mémoire suffisant.

Syntaxe

```
BOOL GetTagMultiStateWait(DWORD* pdwState, const char* pFormat)
```

Paramètres

pdwState

Champ dans lequel sont enregistrés les états des variables

pFormat

Description du format pour toutes les variables demandées ainsi que pour chaque variable le nom et l'adresse de la valeur

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur est survenue.

Voir aussi

Descripteur de format (Page 1586)

Etats de variable (Page 1591)

Exemple GetTagMultiStateWait (Page 1543)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Descripteur de format

Mode de fonctionnement des fonctions GetTag

Exemple GetTagMultiStateWait

GetTagRawStateWait

Fonction

Calcule la valeur d'une variable du type de données "Type de données brutes". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

BOOL GetTagRawStateWait(Tag Tag_Name, BYTE pValue, DWORD size, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagRawStateWait (Page 1546)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagRawStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagSByteStateWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 8 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

signed char GetTagSByteStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "signed char"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateWait (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByteStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagSDWordStateWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 32 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

long GetTagSDWordStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "long"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateWait (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByteStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagSWordStateWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 16 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

short GetTagSWordStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "short"

Voir aussi

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable (Page 1591)

Exemple GetTagSByteStateWait (Page 1548)

Exemple GetTagSByteStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagWordStateWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 16 bits". La valeur est lue explicitement dans l'AS. L'état de la variable est en outre retransmis.

Syntaxe

WORD GetTagWordStateWait(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "WORD"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateWait (Page 1550)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWordStateWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagBitState

Fonction

Calcule la valeur d'une variable du type de données "Variable binaire". L'état de la variable est en outre retransmis.

Syntaxe

BOOL GetTagBitState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "BOOL"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagBitStateWait (Page 1537)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Exemple GetTagBitStateWait

Mode de fonctionnement des fonctions GetTag

GetTagByteState

Fonction

Calcule la valeur d'une variable du type de données "Non signé 8 bits". L'état de la variable est en outre retransmis.

Syntaxe

BYTE GetTagByteState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "OCTET"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateWait (Page 1550)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateWait

GetTagCharState

Fonction

Calcule la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". L'état de la variable est en outre retransmis.

Syntaxe

char* GetTagCharState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur en retour

Pointeur sur la valeur des variables dans le type de données "char".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagCharStateWait (Page 1539)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagCharStateWait

GetTagDoubleState

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". L'état de la variable est en outre retransmis.

Syntaxe

double GetTagDoubleState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "double"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagFloatStateWait (Page 1541)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagFloatStateWait

GetTagDWordState

Fonction

Calcule la valeur d'une variable du type de données "Non signé 32 bits". L'état de la variable est en outre retransmis.

Syntaxe

DWORD GetTagDWordState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "DWORD"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateWait (Page 1550)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateWait

GetTagFloatState

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". L'état de la variable est en outre retransmis.

Syntaxe

float GetTagFloatState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "float"

Voir aussi

Exemple GetTagFloatStateWait (Page 1541)

Etats de variable (Page 1591)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagFloatStateWait

Etats de variable

Mode de fonctionnement des fonctions GetTag

GetTagRawState

Fonction

Calcule la valeur d'une variable du type de données "Type de données brutes". L'état de la variable est en outre retransmis.

Syntaxe

BOOL GetTagRawState(Tag Tag_Name, BYTE* pValue, DWORD size, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagRawStateWait (Page 1546)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagRawStateWait

GetTagSByteState

Fonction

Calcule la valeur d'une variable du type de données "Signé 8 bits". L'état de la variable est en outre retransmis.

Syntaxe

signed char GetTagSByteState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "signed char"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateWait (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagSByteStateWait

GetTagSDWordState

Fonction

Calcule la valeur d'une variable du type de données "Signé 32 bits". L'état de la variable est en outre retransmis.

Syntaxe

long GetTagSDWordState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "long"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateWait (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagSByteStateWait

GetTagSWordState

Fonction

Calcule la valeur d'une variable du type de données "Signé 16 bits". L'état de la variable est en outre retransmis.

Syntaxe

short GetTagSWordState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "short"

Voir aussi

Etats de variable (Page 1591)
Exemple GetTagSByteStateWait (Page 1548)
Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagSByteStateWait

GetTagWordState

Fonction

Calcule la valeur d'une variable du type de données "Non signé 16 bits". L'état de la variable est en outre retransmis.

Syntaxe

WORD GetTagWordState(Tag Tag_Name, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "WORD"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateWait (Page 1550)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateWait

stateqc

wait

GetTagBitStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Variable binaire". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

BOOL GetTagBitStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur de la variable dans le type de données "BOOL"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateQCWait (Page 1549)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Etats de variable

Exemple GetTagWordStateQCWait

GetTagByteStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 8 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

BYTE GetTagByteStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur de la variable dans le type de données "BYTE".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateQCWait (Page 1549)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWordStateQCWait

Etats de variable

Mode de fonctionnement des fonctions GetTag

GetTagCharStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

char* GetTagCharStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur en retour

Pointeur sur la valeur des variables dans le type de données "char".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagCharStateQCWait (Page 1539)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagCharStateQCWait

GetTagDoubleStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

```
double GetTagDoubleStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD
pdwQualityCode);
```

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "double".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagFloatStateQCWait (Page 1541)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Etats de variable

Exemple GetTagFloatStateQCWait

GetTagDWordStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 32 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

DWORD GetTagDWordStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "DWORD".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateQCWait (Page 1549)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateQCWait

GetTagFloatStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

```
float GetTagFloatStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD
pdwQualityCode);
```

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "float".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagFloatStateQCWait (Page 1541)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagFloatStateQCWait

GetTagMultiStateQCWait

Fonction

Les valeurs, états et codes qualité de plusieurs variables sont déterminés et écrits à l'adresse correspondante dans le format indiqué. Les valeurs sont lues explicitement dans l'AS.

Il est nécessaire de transmettre à la fonction deux 'DWORD Array' dans les membres desquelles se trouvent les états et codes qualité des diverses variables après appel de la fonction. Sélectionnez ces 'Array' suffisamment grande pour que ces états et ces codes de qualité disposent d'un espace mémoire suffisant.

Syntaxe

BOOL GetTagMultiStateQCWait(DWORD* pdwState, DWORD* pdwQualityCode, const char* pFormat)

Paramètres

pdwState

Champ dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Champ dans lequel est écrit le code qualité de la variable après exécution de la fonction.

pFormat

Description de format pour toutes les variables demandées ainsi que pour chaque variable le nom et l'adresse de la valeur

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur est survenue.

Voir aussi

Descripteur de format (Page 1586)

Etats de variable (Page 1591)

Exemple GetTagMultiStateQCWait (Page 1542)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagMultiStateQCWait

GetTagRawStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Type de données brutes". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

BOOL GetTagRawStateQCWait(Tag Tag_Name, BYTE pValue, DWORD size, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes.

size

Taille du champ d'octets en octets

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagRawStateQCWait (Page 1545)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagRawStateQCWait

GetTagSByteStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 8 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

signed char GetTagSByteStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "signed char".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateQCWait (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagSByteStateQCWait

GetTagSDWordStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 32 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

long GetTagSDWordStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "long".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateQCWait (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagSByteStateQCWait

GetTagSWordStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 16 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

short GetTagSWordStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "short".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateQCWait (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagSByteStateQCWait

GetTagValueStateQCWait

Fonction

Permet la remise d'une valeur sous forme de Variant. Détermine le pointeur sur la structure de résultat qui contient la valeur. La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

```
BOOL GetTagValueStateQCWait(LPDM_VARKEY lpdmVarKey,
LPDM_VAR_UPDATE_STRUCTEX lpdmresult, LPCMN_ERROR lpdmError);
```

Paramètres

lpdmVarKey

Pointeur sur une structure du type de données "DM_VARKEY"

lpdmresult

Pointeur sur la valeur du type de données "DM_VAR_UPDATE_STRUCTEX"

lpdmError

Pointeur sur la structure contenant la description d'erreurs

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Définition de structure CMN_ERROR (Page 1594)

Définition de structure DM_VAR_UPDATE_STRUCTEX (Page 1597)

Définition de structure DM_VARKEY (Page 1598)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Définition de structure CMN_ERROR

Définition de structure DM_VAR_UPDATE_STRUCTEX

Définition de structure DM_VARKEY

GetTagWordStateQCWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 16 bits". La valeur est lue explicitement dans l'AS. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

WORD GetTagWordStateQCWait(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "WORD".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateQCWait (Page 1549)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateQCWait

GetTagBitStateQC

Fonction

Calcule la valeur d'une variable du type de données "Variable binaire". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

BOOL GetTagBitStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur de la variable dans le type de données "BOOL"

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagBitStateQC (Page 1536)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagBitStateQC

GetTagByteStateQC

Fonction

Calcule la valeur d'une variable du type de données "Non signé 8 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

BYTE GetTagByteStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur de la variable dans le type de données "BYTE".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateQCWait (Page 1549)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateQCWait

GetTagCharStateQC

Fonction

Calcule la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

char* GetTagCharStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur en retour

Pointeur sur la valeur des variables dans le type de données "char".

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagCharStateQCWait (Page 1539)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagCharStateQCWait

GetTagDoubleStateQC

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

double GetTagDoubleStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "double".

Voir aussi

Exemple GetTagFloatStateQCWait (Page 1541)

Etats de variable (Page 1591)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagFloatStateQCWait

Etats de variable

Mode de fonctionnement des fonctions GetTag

GetTagDWordStateQC

Fonction

Calcule la valeur d'une variable du type de données "Non signé 32 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

DWORD GetTagDWordStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "DWORD".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateQCWait (Page 1549)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWordStateQCWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagFloatStateQC

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

```
float GetTagFloatStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD
pdwQualityCode);
```

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "float".

Voir aussi

Exemple GetTagFloatStateQCWait (Page 1541)

Etats de variable (Page 1591)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagFloatStateQCWait

Etats de variable

Mode de fonctionnement des fonctions GetTag

GetTagRawStateQC

Fonction

Calcule la valeur d'une variable du type de données "Type de données brutes". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

BOOL GetTagRawStateQC(Tag Tag_Name, BYTE* pValue, DWORD size, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes.

size

Taille du champ d'octets en octets

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagRawStateQCWait (Page 1545)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagRawStateQCWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagSByteStateQC

Fonction

Calcule la valeur d'une variable du type de données "Signé 8 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

signed char GetTagSByteStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type des données "signed char".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateQCWait (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByteStateQCWait

Etats de variable

Mode de fonctionnement des fonctions GetTag

GetTagSDWordStateQC

Fonction

Calcule la valeur d'une variable du type de données "Signé 32 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

long GetTagSDWordStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "long".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateQCWait (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByteStateQCWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagSWordStateQC

Fonction

Calcule la valeur d'une variable du type de données "Signé 16 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

short GetTagSWordStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "short".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagSByteStateQCWait (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByteStateQCWait

Mode de fonctionnement des fonctions GetTag

Etats de variable

GetTagValueStateQC

Fonction

Permet la remise d'une valeur sous forme de Variant. Détermine le pointeur sur la structure de résultat qui contient la valeur. L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

```
BOOL GetTagValueStateQC(LPDM_VARKEY lpdmVarKey,
LPDM_VAR_UPDATE_STRUCTEX lpdmresult, LPCMN_ERROR lpdmError);
```

Paramètres

lpdmVarKey

Pointeur sur une structure du type de données "DM_VARKEY"

lpdmresult

Pointeur sur la valeur du type de données "DM_VAR_UPDATE_STRUCTEX"

lpdmError

Pointeur sur la structure contenant la description d'erreurs

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Définition de structure CMN_ERROR (Page 1594)

Définition de structure DM_VAR_UPDATE_STRUCTEX (Page 1597)

Définition de structure DM_VARKEY (Page 1598)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Définition de structure DM_VARKEY

Définition de structure DM_VAR_UPDATE_STRUCTEX

Définition de structure CMN_ERROR

GetTagWordStateQC

Fonction

Calcule la valeur d'une variable du type de données "Non signé 16 bits". L'état et le code qualité de la variable sont en outre retransmis.

Syntaxe

WORD GetTagWordStateQC(Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Paramètres

Tag_Name

Nom de la variable.

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

pdwQualityCode

Pointeur sur un DWORD dans lequel est écrit le code qualité de la variable après exécution de la fonction.

Valeur de retour

Valeur des variables dans le type de données "WORD".

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagWordStateQCWait (Page 1549)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Etats de variable

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWordStateQCWait

wait

GetTagBitWait

Fonction

Calcule la valeur d'une variable du type de données "Variable binaire". La valeur est lue explicitement dans l'AS.

Syntaxe

BOOL GetTagBitWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "BOOL"

Voir aussi

Exemple GetTagBit (Page 1536)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Exemple GetTagBit

GetTagByteWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 8 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

BYTE GetTagByteWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "OCTET"

Voir aussi

Exemple GetTagWord (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Exemple GetTagWord

GetTagCharWait

Fonction

Calcule la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

char* GetTagCharWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur en retour

Pointeur sur une chaîne de caractères contenant la valeur des variables.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
   .......
}
```

Voir aussi

Exemple GetTagChar (Page 1538)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Exemple GetTagChar

GetTagDoubleWait

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

double GetTagDoubleWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "double"

Voir aussi

Exemple GetTagFloat (Page 1540)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Exemple GetTagFloat

GetTagDWordWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 32 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

DWORD GetTagDWordWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type des données "DWORD"

Voir aussi

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWord (Page 1548)

Exemple GetTagWord

Mode de fonctionnement des fonctions GetTag

GetTagFloatWait

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

float GetTagFloatWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "float"

Voir aussi

Exemple GetTagFloat (Page 1540)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagFloat

Mode de fonctionnement des fonctions GetTag

GetTagMultiWait

Fonction

Les valeurs de plusieurs variables sont déterminées et écrites à l'adresse correspondante dans le format indiqué. La valeur est lue explicitement dans l'AS. La mémoire pour la valeur de variable est créée par la fonction SysMalloc.

Syntaxe

BOOL GetTagMultiWait(const char* pFormat,...)

Paramètres

pFormat

Description du format pour toutes les variables demandées ainsi que pour chaque variable le nom et l'adresse de la valeur

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Descripteur de format (Page 1586)

Exemple GetTagMultiWait (Page 1544)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Exemple GetTagMultiWait

Descripteur de format

GetTagRawWait

Fonction

Calcule la valeur d'une variable du type de données "Type de données brutes". La valeur est lue explicitement dans l'AS.

Syntaxe

BOOL GetTagRawWait(Tag Tag_Name , BYTE pValue, DWORD size);

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur est survenue.

Voir aussi

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagRaw (Page 1545)

Mode de fonctionnement des fonctions GetTag

Exemple GetTagRaw

GetTagSByteWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 8 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

signed char GetTagSByteWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "signed char"

Voir aussi

Exemple GetTagSByte (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Exemple GetTagSByte

GetTagSDWordWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 32 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

long GetTagSDWordWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type des données "long"

Voir aussi

Exemple GetTagSByte (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByte

Mode de fonctionnement des fonctions GetTag

GetTagSWordWait

Fonction

Calcule la valeur d'une variable du type de données "Signé 16 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

short GetTagSWordWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type des données "short"

Voir aussi

Exemple GetTagSByte (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByte

Mode de fonctionnement des fonctions GetTag

GetTagValueWait

Fonction

Permet la remise d'une valeur sous forme de Variant. Détermine le pointeur sur la structure de résultat qui contient la valeur. La valeur est lue directement dans l'AS.

Syntaxe

BOOL GetTagValueWait(LPDM_VARKEY lpdmVarKey, LPDM_VAR_UPDATE_STRUCT lpdmresult, LPCMN_ERROR lpdmError);

Paramètres

lpdmVarKey

Pointeur sur une structure du type de données "DM_VARKEY"

lpdmresult

Pointeur sur la valeur du type de données "DM_VAR_UPDATE_STRUCT"

lpdmError

Pointeur sur la structure contenant la description d'erreurs

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Définition de structure CMN_ERROR (Page 1594)

Définition de structure DM_VAR_UPDATE_STRUCT (Page 1596)

Définition de structure DM_VARKEY (Page 1598)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Définition de structure DM_VARKEY

Définition de structure DM_VAR_UPDATE_STRUCTEX

Définition de structure CMN_ERROR

GetTagWordWait

Fonction

Calcule la valeur d'une variable du type de données "Non signé 16 bits". La valeur est lue explicitement dans l'AS.

Syntaxe

WORD GetTagWordWait(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "WORD"

Voir aussi

Exemple GetTagWord (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWord

Mode de fonctionnement des fonctions GetTag

GetTagBit

Fonction

Calcule la valeur d'une variable du type de données "Variable binaire".

Syntaxe

BOOL GetTagBit(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "BOOL"

Voir aussi

Exemple GetTagBit (Page 1536)
Mode de fonctionnement des fonctions GetTag (Page 1394)
Exemple GetTagBit
Mode de fonctionnement des fonctions GetTag

GetTagByte

Fonction

Calcule la valeur d'une variable du type de données "Non signé 8 bits".

Syntaxe

BYTE GetTagByte(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "OCTET"

Voir aussi

Exemple GetTagWord (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWord

Mode de fonctionnement des fonctions GetTag

GetTagChar

Fonction

Calcule la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits".

Syntaxe

char* GetTagChar(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur en retour

Pointeur sur une chaîne de caractères contenant la valeur des variables.

La validité de la valeur en retour doit être contrôlée afin d'éviter une exception de pointeur zéro, par ex. pour la fonction "GetText()" :

```
pszValue = GetText(lpszPictureName,"Text1");
if(pszValue != NULL)
{
  .......
}
```

Voir aussi

Exemple GetTagChar (Page 1538)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagChar

Mode de fonctionnement des fonctions GetTag

GetTagDouble

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits".

Syntaxe

double GetTagDouble(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "double"

Voir aussi

Exemple GetTagFloat (Page 1540)
Mode de fonctionnement des fonctions GetTag (Page 1394)
Exemple GetTagFloat
Mode de fonctionnement des fonctions GetTag

GetTagDWord

Fonction

Calcule la valeur d'une variable du type de données "Non signé 32 bits".

Syntaxe

DWORD GetTagDWord(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type des données "DWORD"

Voir aussi

Exemple GetTagWord (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWord

Mode de fonctionnement des fonctions GetTag

GetTagFloat

Fonction

Calcule la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits".

Syntaxe

```
float GetTagFloat(Tag Tag_Name);
```

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "float".

Voir aussi

Exemple GetTagFloat (Page 1540)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagFloat

Mode de fonctionnement des fonctions GetTag

GetTagRaw

Fonction

Calcule la valeur d'une variable du type de données "Type de données brutes".

Syntaxe

```
BOOL GetTagRaw(Tag Tag_Name, BYTE* pValue, DWORD size);
```

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple GetTagRaw (Page 1545)
Mode de fonctionnement des fonctions GetTag (Page 1394)
Mode de fonctionnement des fonctions GetTag
Exemple GetTagRaw

GetTagSByte

Fonction

Calcule la valeur d'une variable du type de données "Signé 8 bits".

Syntaxe

signed char GetTagSByte(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "signed char"

Voir aussi

Exemple GetTagSByte (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByte

Mode de fonctionnement des fonctions GetTag

GetTagSDWord

Fonction

Calcule la valeur d'une variable du type de données "Signé 32 bits".

Syntaxe

long GetTagSDWord(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type des données "long"

Voir aussi

Exemple GetTagSByte (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByte

Mode de fonctionnement des fonctions GetTag

GetTagSWord

Fonction

Calcule la valeur d'une variable du type de données "Signé 16 bits".

Syntaxe

short GetTagSWord(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type des données "short"

Voir aussi

Exemple GetTagSByte (Page 1547)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagSByte

Mode de fonctionnement des fonctions GetTag

GetTagValue

Fonction

Permet la remise d'une valeur sous forme de Variant. Détermine le pointeur sur la structure de résultat qui contient la valeur.

Syntaxe

BOOL GetTagValue(LPDM_VARKEY lpdmVarKey, LPDM_VAR_UPDATE_STRUCT lpdmresult, LPCMN_ERROR lpdmError);

Paramètres

lpdmVarKey

Pointeur sur une structure du type de données "DM_VARKEY"

lpdmresult

Pointeur sur la valeur du type de données "DM_VAR_UPDATE_STRUCT"

lpdmError

Pointeur sur la structure contenant la description d'erreurs

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Définition de structure CMN_ERROR (Page 1594)

Définition de structure DM_VAR_UPDATE_STRUCT (Page 1596)

Définition de structure DM_VARKEY (Page 1598)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Mode de fonctionnement des fonctions GetTag

Définition de structure DM_VARKEY

Définition de structure DM_VAR_UPDATE_STRUCTEX

Définition de structure CMN_ERROR

GetTagWord

Fonction

Calcule la valeur d'une variable du type de données "Non signé 16 bits".

Syntaxe

WORD GetTagWord(Tag Tag_Name);

Paramètres

Tag_Name

Nom des variables

Valeur de retour

Valeur des variables dans le type de données "WORD"

Voir aussi

Exemple GetTagWord (Page 1548)

Mode de fonctionnement des fonctions GetTag (Page 1394)

Exemple GetTagWord

Mode de fonctionnement des fonctions GetTag

set

Fonctionnement des fonctions SetTag

SetTagXXX

La fonction SetTagXXX donne l'ordre d'écrire une valeur et retourne ensuite immédiatement à l'appelant. Le système n'attend pas que la valeur ait été effectivement écrite.

L'appel se caractérise ainsi de la façon suivante :

- L'appel est rapide.
- L'appelant ne sait pas quand la valeur est effectivement écrite.
- La fonction ne fournit aucune information concernant l'état du travail d'écriture.

SetTagXXXWait

La fonction SetTagXXXWait donne l'ordre d'écrire une valeur et ne retourne à l'appelant de la fonction que lorsque la valeur a effectivement été écrite.

L'appel se caractérise ainsi de la façon suivante :

- L'appel dure plus longtemps qu'avec SetTagXXX. La durée dépend entre autres du canal et de l'AS.
- Après l'appel, la valeur est écrite.
- La fonction ne fournit aucune information concernant l'état du travail d'écriture.

SetTagXXXState

La fonction SetTagXXXState possède les mêmes caractéristiques que SetTagXXX, mais fournit en plus des informations concernant l'état de l'ordre d'écriture.

Etant donné que l'état est toujours donné sur le plan interne, on ne peut constater aucune différence avec la performance de SetTagXXX.

SetTagXXXStateWait

La fonction SetTagXXXStateWait possède les mêmes caractéristiques que SetTagXXXWait, mais fournit en plus des informations concernant l'état du travail d'écriture.

Etant donné que l'état est toujours donné sur le plan interne, on ne peut constater aucune différence avec la performance de SetTagXXXWait.

La différence entre les fonctions SetTagXXXStateWait et SetTagXXXState correspond à la différence entre SetTagXXXWait et SetTagXXX.

Notez que certains états ne peuvent être constitués que lorsque la procédure d'écriture est terminée.

SetTagMultiWait

La fonction SetTagMultiWait possède les mêmes caractéristiques que SetTagXXXWait. Elle offre toutefois la possibilité de distribuer plusieurs travaux d'écriture regroupés dans un seul travail.

state

wait

SetTagBitStateWait

Fonction

Met à un la valeur d'une variable du type de données "Variable binaire". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

BOOL SetTagBitStateWait(Tag Tag_Name, short value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "short"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagBitStateWait (Page 1566)

Etats de variable

Exemple SetTagBitStateWait

SetTagByteStateWait

Fonction

Met à un la valeur d'une variable du type de données "Non signé 8 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

BOOL SetTagByteStateWait(Tag Tag_Name, BYTE value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "OCTET"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagWordStateWait (Page 1573)

Etats de variable

Exemple SetTagWordStateWait

SetTagCharStateWait

Fonction

Met à un la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

```
BOOL SetTagCharStateWait(Tag Tag_Name, LPSTR value, PDWORD lp_dwstate);
```

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "LPSTR"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagCharStateWait (Page 1567)

Etats de variable

Exemple SetTagCharStateWait

SetTagDoubleStateWait

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

```
BOOL SetTagDoubleStateWait(Tag Tag_Name, double value, PDWORD lp_dwstate);
```

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "double"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagFloatStateWait (Page 1568)

Etats de variable

Exemple SetTagFloatStateWait

SetTagDWordStateWait

Fonction

Met à un la valeur d'une variable du type de données "Non signé 32 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

```
BOOL SetTagDWordStateWait(Tag Tag_Name, DWORD value, PDWORD lp_dwstate);
```

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "DWORD"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagWordStateWait (Page 1573)

Etats de variable

Exemple SetTagWordStateWait

SetTagFloatStateWait

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

```
BOOL SetTagFloatStateWait(Tag Tag_Name, float value, PDWORD lp_dwstate);
```

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "float"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagFloatStateWait (Page 1568)

Etats de variable

Exemple SetTagFloatStateWait

SetTagMultiStateWait

Fonction

Met à un les valeurs de plusieurs variables. La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Il est nécessaire de transmettre à la fonction une 'DWORD Array' dans les membres de laquelle se trouvent les états des diverses variables après appel de la fonction. Sélectionnez cette 'Array' suffisamment grande pour que ces états disposent d'un espace mémoire suffisant.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

BOOL SetTagMultiStateWait(DWORD* pdwState, const char* pFormat,...)

Paramètres

pdwState

Champ dans lequel sont enregistrés les états des variables

pFormat

Description de format pour toutes les variables demandées ainsi que pour chaque variable le nom et la valeur

Descripteur de formatBEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur est survenue.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagMultiStateWait (Page 1568)

Etats de variable

Exemple SetTagMultiStateWait

SetTagRawStateWait

Fonction

Met à un la valeur d'une variable du type de données "Type de données brutes". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

BOOL SetTagRawStateWait(Tag Tag_Name, BYTE pValue, DWORD size, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur est survenue.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagRawStateWait (Page 1571)

Etats de variable

Exemple SetTagRawStateWait

SetTagSByteStateWait

Fonction

Met à un la valeur d'une variable du type de données "Signé 8 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

```
BOOL SetTagSByteStateWait(Tag Tag_Name, signed char value, PDWORD lp_dwstate);
```

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "signed char"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagSByteStateWait (Page 1572)

Etats de variable

Exemple SetTagSByteStateWait

SetTagSDWordStateWait

Fonction

Met à un la valeur d'une variable du type de données "Signé 32 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

BOOL SetTagSDWordStateWait(Tag Tag_Name, long value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "long"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagSByteStateWait (Page 1572)

Exemple SetTagSByteStateWait

Etats de variable

SetTagSWordStateWait

Fonction

Met à un la valeur d'une variable du type de données "Signé 16 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

BOOL SetTagSWordStateWait(Tag Tag_Name, short value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "short"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagSByteStateWait (Page 1572)

Etats de variable

Exemple SetTagSByteStateWait

SetTagWordStateWait

Fonction

Met à un la valeur d'une variable du type de données "Non signé 16 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur. L'état de la variable est en outre retransmis.

Fonctionnement des fonctions SetTag
BEISPIELE_INTERNE_FUNKTIONEN_TAG_STATEWAIT_23_130

Syntaxe

BOOL SetTagWordStateWait(Tag Tag_Name, WORD value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "WORD"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagWordStateWait (Page 1573)

Exemple SetTagWordStateWait

Etats de variable

SetTagBitState

Fonction

Met à un la valeur d'une variable du type de données "Variable binaire". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagBitState(Tag Tag_Name, short int value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "short int"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagBitStateWait (Page 1566)

Fonctionnement des fonctions SetTag (Page 1460)

Etats de variable

Exemple SetTagBitStateWait

Mode de fonctionnement des fonctions SetTag

SetTagByteState

Fonction

Met à un la valeur d'une variable du type de données "Non signé 8 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagByteState(Tag Tag_Name, BYTE value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "OCTET"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagWordStateWait (Page 1573)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Etats de variable

Exemple SetTagWordStateWait

SetTagCharState

Fonction

Met à un la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagCharState(Tag Tag_Name, LPSTR value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "LPSTR"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagCharStateWait (Page 1567)

Fonctionnement des fonctions SetTag (Page 1460)

Etats de variable

Mode de fonctionnement des fonctions SetTag

Exemple SetTagCharStateWait

SetTagDoubleState

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagDoubleState(Tag Tag_Name, double value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "double"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagFloatStateWait (Page 1568)

Fonctionnement des fonctions SetTag (Page 1460)

Etats de variable

Mode de fonctionnement des fonctions SetTag

Exemple SetTagFloatStateWait

SetTagDWordState

Fonction

Met à un la valeur d'une variable du type de données "Non signé 32 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagDWordState(Tag Tag_Name, DWORD value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "DWORD"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagWordStateWait (Page 1573)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagWordStateWait

Etats de variable

Mode de fonctionnement des fonctions SetTag

SetTagFloatState

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagFloatState(Tag Tag_Name, float value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "float"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagFloatStateWait (Page 1568)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagFloatStateWait

Etats de variable

Mode de fonctionnement des fonctions SetTag

SetTagRawState

Fonction

Met à un la valeur d'une variable du type de données "Type de données brutes". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagRawState(Tag Tag_Name, BYTE* pValue, DWORD size, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple GetTagRaw (Page 1545)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Etats de variable

Exemple GetTagRaw

SetTagSByteState

Fonction

Met à un la valeur d'une variable du type de données "Signé 8 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagSByteState(Tag Tag_Name, signed char value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "signed char"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagSByteStateWait (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagSByteStateWait

Mode de fonctionnement des fonctions SetTag

Etats de variable

SetTagSDWordState

Fonction

Met à un la valeur d'une variable du type de données "Signé 32 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagSDWordState(Tag Tag_Name, long value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "long"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagSByteStateWait (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagSByteStateWait

Mode de fonctionnement des fonctions SetTag

Etats de variable

SetTagSWordState

Fonction

Met à un la valeur d'une variable du type de données "Signé 16 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagSWordState(Tag Tag_Name, short value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "short"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagSByteStateWait (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagSByteStateWait

Mode de fonctionnement des fonctions SetTag

Etats de variable

SetTagWordState

Fonction

Met à un la valeur d'une variable du type de données "Non signé 16 bits". L'état de la variable est en outre retransmis.

Syntaxe

BOOL SetTagWordState(Tag Tag_Name, WORD value, PDWORD lp_dwstate);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "short"

lp_dwstate

Pointeur sur un DWORD dans lequel est écrit l'état de la variable après exécution de la fonction.

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur. Pour cela, l'état de la variable doit être évalué.

FALSE

Une erreur s'est produite.

Voir aussi

Etats de variable (Page 1591)

Exemple SetTagWordStateWait (Page 1573)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagWordStateWait

Etats de variable

Mode de fonctionnement des fonctions SetTag

wait

SetTagBitWait

Fonction

Met à un la valeur d'une variable du type de données "Variable binaire". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagBitWait(Tag Tag_Name, short value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "short"

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagBit (Page 1566)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagBit

SetTagByteWait

Fonction

Met à un la valeur d'une variable du type de données "Non signé 8 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagByteWait(Tag Tag_Name, BYTE value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "OCTET"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur est survenue.

Voir aussi

Exemple SetTagWord (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagWord

SetTagCharWait

Fonction

Met à un la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagCharWait(Tag Tag_Name, LPSTR value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "LPSTR"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagChar (Page 1567)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagChar

SetTagDoubleWait

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagDoubleWait(Tag Tag_Name, double value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "double"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagFloat (Page 1568)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagFloat

SetTagDWordWait

Fonction

Met à un la valeur d'une variable du type de données "Non signé 32 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagDWordWait(Tag Tag_Name, DWORD value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "DWORD"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagWord (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagWord

Mode de fonctionnement des fonctions SetTag

SetTagFloatWait

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagFloatWait(Tag Tag_Name, float value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "float"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagFloat (Page 1568)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagFloat

Mode de fonctionnement des fonctions SetTag

SetTagMultiWait

Fonction

Les valeurs de plusieurs variables sont mises au format indiqué. La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagMultiWait(const char* pFormat,...)

Paramètres

pFormat

Description de format pour toutes les variables demandées ainsi que pour chaque variable le nom et la valeur

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Descripteur de format (Page 1586)

Exemple SetTagMultiWait (Page 1569)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Descripteur de format

Exemple SetTagMultiWait

SetTagRawWait

Fonction

Met à un la valeur d'une variable du type de données "Type de données brutes". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagRawWait(Tag Tag_Name, BYTE pValue, DWORD size);

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagRaw (Page 1570)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagRaw

SetTagSByteWait

Fonction

Met à un la valeur d'une variable du type de données "Signé 8 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagSByteWait(Tag Tag_Name, signed char value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "signed char"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagSByte (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagSByte

SetTagSDWordWait

Fonction

Met à un la valeur d'une variable du type de données "Signé 32 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagSDWordWait(Tag Tag_Name, long value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "long"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagSByte (Page 1572)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagSByte

SetTagSWordWait

Fonction

Met à un la valeur d'une variable du type de données "Signé 16 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagSWordWait(Tag Tag_Name, short value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "short"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagSByte (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagSByte

Mode de fonctionnement des fonctions SetTag

SetTagValueWait

Fonction

Permet le transfert d'une valeur sous forme de Variant et positionne le pointeur sur la valeur du type de données "Variant". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagValueWait(LPDM_VARKEY lpdmVarKey, LPVARIANT lpdmValue, PDWORD dwState, LPCMN_ERROR lpdmError);

Paramètres

lpdmVarKey

Pointeur sur une structure du type de données "DM_VARKEY"

lpdmValue

Pointeur sur la valeur du type de données "Variant". Pour la description du type de données VARIANT, veuillez vous référer aux ouvrages spécialisés.

dwState

Etat de la variable qui est renvoyé après exécution de la fonction

lpdmError

Pointeur sur la structure contenant la description d'erreurs

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur

FALSE

Une erreur s'est produite.

Voir aussi

Définition de structure CMN_ERROR (Page 1594)

Etats de variable (Page 1591)

Définition de structure DM_VAR_UPDATE_STRUCT (Page 1596)

Définition de structure DM_VARKEY (Page 1598)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Définition de structure DM_VARKEY

Définition de structure DM_VAR_UPDATE_STRUCTEX

Définition de structure CMN_ERROR

SetTagWordWait

Fonction

Met à un la valeur d'une variable du type de données "Non signé 16 bits". La fonction n'est terminée qu'après que l'AS ait certifié la validation de la valeur.

Syntaxe

BOOL SetTagWordWait(Tag Tag_Name, WORD value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "WORD"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagWord (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagWord

Mode de fonctionnement des fonctions SetTag

SetTagBit

Fonction

Met à un la valeur d'une variable du type de données "Variable binaire".

Syntaxe

BOOL SetTagBit(Tag Tag_Name, short int value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "short int"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagBit (Page 1566)

Exemple SetTagBit

Mode de fonctionnement des fonctions SetTag

SetTagByte

Fonction

Met à un la valeur d'une variable du type de données "Non signé 8 bits".

Syntaxe

BOOL SetTagByte(Tag Tag_Name, BYTE value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "OCTET"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagWord (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagWord

Mode de fonctionnement des fonctions SetTag

SetTagChar

Fonction

Met à un la valeur d'une variable du type de données "Variable de texte 8 bits" ou "Variable de texte 16 bits".

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "LPSTR"

Valeur en retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagChar (Page 1567)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagChar

SetTagDouble

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 64 bits".

Syntaxe

BOOL SetTagDouble(Tag Tag_Name, double value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "double"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagFloat (Page 1568)
Fonctionnement des fonctions SetTag (Page 1460)
Mode de fonctionnement des fonctions SetTag
Exemple SetTagFloat

SetTagDWord

Fonction

Met à un la valeur d'une variable du type de données "Non signé 32 bits".

Syntaxe

BOOL SetTagDWord(Tag Tag_Name, DWORD value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "DWORD"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagWord (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagWord

Mode de fonctionnement des fonctions SetTag

SetTagFloat

Fonction

Met à un la valeur d'une variable du type de données "Nombre à virgule flottante 32 bits".

Syntaxe

BOOL SetTagFloat(Tag Tag_Name, float value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "float"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagFloat (Page 1568)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagFloat

Mode de fonctionnement des fonctions SetTag

SetTagRaw

Fonction

Met à un la valeur d'une variable du type de données "Type de données brutes".

Syntaxe

BOOL SetTagRaw(Tag Tag_Name, BYTE* pValue, DWORD size);

Paramètres

Tag_Name

Nom des variables

pValue

Pointeur sur un champ d'octets qui contient la valeur des variables des données brutes

size

Taille du champ d'octets en octets

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagRaw (Page 1570)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagRaw

SetTagSByte

Fonction

Met à un la valeur d'une variable du type de données "Signé 8 bits".

Syntaxe

BOOL SetTagSByte(Tag Tag_Name, signed char value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "signed char"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagSByte (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Mode de fonctionnement des fonctions SetTag

Exemple SetTagSByte

SetTagSDWord

Fonction

Met à un la valeur d'une variable du type de données "Signé 32 bits".

Syntaxe

BOOL SetTagSDWord(Tag Tag_Name, long value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "long"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagSByte (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagSByte

Mode de fonctionnement des fonctions SetTag

SetTagSWord

Fonction

Met à un la valeur d'une variable du type de données "Signé 16 bits".

Syntaxe

BOOL SetTagSWord(Tag Tag_Name, short value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type des données "short"

size

Taille du champ d'octets en octets

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagSByte (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagSByte

Mode de fonctionnement des fonctions SetTag

SetTagValue

Fonction

Permet le transfert d'une valeur sous forme de Variant et positionne le pointeur sur la valeur du type de données "Variant".

Syntaxe

BOOL SetTagValue(LPDM_VARKEY lpdmVarKey, LPVARIANT lpdmValue, PDWORD dwState, LPCMN_ERROR lpdmError);

Paramètres

lpdmVarKey

Pointeur sur une structure du type de données "DM_VARKEY"

lpdmValue

Pointeur sur la valeur du type de données "Variant". Pour la description du type de données VARIANT, veuillez vous référer aux ouvrages spécialisés.

dwState

Etat de la variable qui est renvoyé après exécution de la fonction

lpdmError

Pointeur sur la structure contenant la description d'erreurs

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Fonctionnement des fonctions SetTag (Page 1460)

Définition de structure CMN_ERROR (Page 1594)

Etats de variable (Page 1591)

Définition de structure DM_VAR_UPDATE_STRUCT (Page 1596)

Définition de structure DM_VARKEY (Page 1598)

Mode de fonctionnement des fonctions SetTag

Définition de structure CMN_ERROR

Définition de structure DM_VAR_UPDATE_STRUCTEX

Définition de structure DM_VARKEY

SetTagWord

Fonction

Met à un la valeur d'une variable du type de données "Non signé 16 bits".

Syntaxe

BOOL SetTagWord(Tag Tag_Name, WORD value);

Paramètres

Tag_Name

Nom des variables

value

Valeur des variables dans le type de données "WORD"

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

Il n'est toutefois pas contrôlé si la variable a également pu être écrite sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple SetTagWord (Page 1572)

Fonctionnement des fonctions SetTag (Page 1460)

Exemple SetTagWord

Mode de fonctionnement des fonctions SetTag

2.15.3.6 WinCC

WinCC - brève description

Les fonctions du groupe WinCC vous permettent d'entreprendre de différents paramétrages dans Runtime.

Les fonctions du sous-groupe Système vous permettent d'influencer le WinCC Runtime.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

system

DeactivateRTProject

Fonction

Désactive le projet activé.

Remarque

Si Runtime est terminé sur un serveur ou un Client, seulement cet ordinateur en est concerné.

Un projet activé pour lequel WinCC Explorer n'est pas lancé, doit être fermé avec la fonction interne "ExitWinCC".

Si le projet activé a été quitté avec la fonction interne "DeactivateRTProject", le projet WinCC reste ouvert à l'arrière-plan. Afin de terminer ce projet, WinCC Explorer doit être ouvert et ensuite être fermé via le menu "Fichier" > "Quitter".

Syntaxe

BOOL DeactivateRTProject ();

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple DeactivateRTProject (Page 1515)

Exemple DeactivateRTProject

ExitWinCC

Fonction

Désactive Runtime et ferme WinCC sur l'ordinateur sur lequel est exécutée la fonction.

Remarque

Si Runtime est terminé sur un serveur ou un Client, seulement cet ordinateur en est concerné.

Un projet activé pour lequel WinCC Explorer n'est pas lancé, doit être fermé avec la fonction interne "ExitWinCC".

Si le projet activé a été quitté avec la fonction interne "DeactivateRTProject", le projet WinCC reste ouvert à l'arrière-plan. Afin de terminer ce projet, WinCC Explorer doit être ouvert et ensuite être fermé via le menu "Fichier" > "Quitter".

Syntaxe

BOOL ExitWinCC ();

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Exemple ExitWinCC (Page 1515)

Exemple ExitWinCC

GetLanguage

Fonction

Détermine la langue Runtime actuelle.

Syntaxe

DWORD GetLanguage();

Valeur en retour

La langue Runtime actuelle avec le code respectif de la langue est renvoyée.

Remarque

Vous trouverez un tableau détaillé "Language code" dans la documentation "Bases du VBScript" sous l'entrée d'index "Language code".

Voir aussi

Exemple GetLanguage (Page 1523)

Exemple GetLanguage

InquireLanguage

Fonction

Détermine toutes les langues qui sont configurées dans la bibliothèque de texte pour le temps d'exécution.

Avec dwCount, vous indiquez où doit être écrit le nombre des identificateurs de langue déterminés.

Syntaxe

DWORD* InquireLanguage (DWORD* dwCount);

Paramètres

dwCount

Pointeur sur le nombre des identificateurs de langue déterminés

Valeur en retour

Les langues configurées avec les codes de langue respectifs sont renvoyées.

Remarque

Vous trouverez un tableau détaillé "Language code" dans la documentation "Bases du VBScript" sous l'entrée d'index "Language code".

Voir aussi

Exemple InquireLanguage (Page 1552)

Exemple InquireLanguage

SetLanguage

Fonction

Modifie le réglage de langue dans Runtime.

Syntaxe

BOOL SetLanguage(DWORD dwLocaleID);

Paramètres

dwLocaleID

Identificateur de langue de la langue à régler

Valeur de retour

TRUE

La fonction a été exécutée sans erreur.

FALSE

Une erreur s'est produite.

Voir aussi

Identificateurs de langue (Page 1589)

Exemple SetLanguage (Page 1561)

Exemple SetLanguage

Identificateurs de langue

FillDiagnoseInTags

Fonction

Active ou désactive l'enregistrement d'informations de diagnostic dans variables.

Etant donné que le remplissage des variables constitue une charge supplémentaire au système, ceci ne devrait être activé que pendant une période courte pour les informations de diagnostic.

Syntaxe

void FillDiagnoseInTags(BOOL bfill);

Paramètres

bFill

Enregistrement d'informations de diagnostic dans variables activé/désactivé

TRUE	Activer affectation des variables de diagnostic
FALSE	Désactiver affectation des variables de diagnostic

Variables de diagnostic de GlobalScript

@SCRIPT_COUNT_TAGS

Cette variable contient le nombre actuel des variables demandées par Script.

@SCRIPT_COUNT_REQUEST_IN_QUEUES

Cette variable contient le nombre actuel de travaux.

@SCRIPT_COUNT_ACTIONS_IN_QUEUES

Cette variable contient le nombre actuel d'actions.

GetServerTagPrefix

Fonction

Afin de pouvoir accéder, dans un système réparti, à des variables du serveur correspondant depuis un client WinCC, il faut ajouter aux noms de variable le préfixe de serveur.

Si l'accès aux variables s'effectue avec les fonctions GetTagxx ou SetTagxx, la commande d'actions effectue l'ajout nécessaire.

Si des fonction de WinCC API sont utilisées pour l'accès, les noms de variable doivent être complétés par l'utilisateur. La fonction GetServerTagPrefix en fournit ces préfixes.

A chaque fois est renvoyé un pointeur du type "char" sur ServerPrefix, TagPrefix et WindowPrefix.

L'utilisateur ne doit ni modifier (strcat non plus) ni débloquer la mémoire.

Syntaxe

void GetServerTagPrefix(char** ppszServerPrefix, char** ppszTagPrefix, char** ppszWindowPrefix);

Paramètres

ppszServerPrefix

Pointeur sur un pointeur qui fait référence au préfixe de serveur.

ppszTagPrefix

Pointeur sur un pointeur qui fait référence au préfixe de Tag.

ppszWindowPrefix

Pointeur sur un pointeur qui fait référence au préfixe Window.

Voir aussi

Exemple GetServerTagPrefix (Page 1535)

Exemple GetServerTagPrefix

TraceText

Fonction

La valeur indiquée dans <Paramètre> est listée dans APDiag lorsque le niveau de diagnostic indiqué est atteint.

Syntaxe

void TraceText(DWORD dwTraceLevel, char* pszFormat, <Parameter>);

Paramètres

dwTraceLevel

Niveau de diagnostic

pszFormat

Format de sortie (correspondant à la fonction printf)

<Paramètres>

Valeur à lister

Remarque

Dans le dialogue de paramétrage pour cette fonction, la sélection de variables, objets graphique et vues est offerte.

TraceTime

Fonction

La valeur indiquée dans <Paramètre> est listée dans APDiag lorsque le niveau de diagnostic indiqué est atteint.

En outre, le temps à partir de AP lancement de diagnostic est sorti en millisecondes afin de permettre des mesures de performance.

Syntaxe

void TraceTime(DWORD dwTraceLevel, char* pszFormat, <Parameter>);

Paramètres

dwTraceLevel

Niveau de diagnostic

pszFormat

Format de sortie (correspondant à la fonction printf)

<Paramètres>

Valeur à consigner

Remarque

Dans le dialogue de paramétrage pour cette fonction, la sélection de variables, objets graphique et vues est offerte.

2.15.4 Exemples

2.15.4.1 Exemples - A à G

Exemple AcknowledgeMessage

```
{
//Acknowledge the AlarmLogging message which is selected
AcknowledgeMessage(GetTagWord("U08i_MsgNr"));
}
```

Indiquer le numéro d'alarme à acquitter. Celui-ci y est lu dans une variable.

Exemple AXC_OnBtnMsgFirst

```
{
// jump to the first message in the WinCC Alarm Control
AXC_OnBtnMsgFirst("gs_alarm_00","Control1");
}
```

Paramètres de la fonction AXC_OnBtnMsgFirst:

"gs_alarm_00" est le nom de la vue dans laquelle le WinCC Alarm Control a été configuré.

Control1 est le nom d'objet du WinCC Alarm Control.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Exemple AXC_OnBtnMsgLast

```
{
// jump to the last message in the WinCC Alarm Control
AXC_OnBtnMsgLast("gs_alarm_00","Control1");
}
```

Paramètres de la fonction AXC_OnBtnMsgLast :

"gs_alarm_00" est le nom de la vue dans laquelle le WinCC Alarm Control a été configuré.

Control1 est le nom d'objet du WinCC Alarm Control.

Exemple AXC_OnBtnScroll

```
{
// activate/deactivate the scroll function
AXC_OnBtnScroll("gs_alarm_00","Control1");
}
```

Paramètres de la fonction AXC_OnBtnScroll:

"gs_alarm_00" est le nom de la vue dans laquelle le WinCC Alarm Control a été configuré.

Control1 est le nom d'objet du WinCC Alarm Control.

Exemple AXC_OnBtnSinglAckn

```
{
// acknowledge the active message
AXC_OnBtnSinglAckn("gs_alarm_00","Control1");
}
```

Paramètres de la fonction AXC_OnBtnSinglAckn:

"gs_alarm_00" est le nom de la vue dans laquelle le WinCC Alarm Control a été configuré.

Control1 est le nom d'objet du WinCC Alarm Control.

Exemple AXC_SetFilter

```
{
BOOL ret;
MSG_FILTER_STRUCT Filter;
CMN_ERROR Error;

//Reset the filter struct
memset( &Filter, 0, sizeof( MSG_FILTER_STRUCT ) );

//Set the filter name
```

```
strcpy( Filter.szFilterName, "Control1");

// Sélectionner éléments de sélection
Filter.dwFilter = MSG_FILTER_NR_FROM | MSG_FILTER_NR_TO;

// Numéro d'alarme de
Filter.dwMsgNr[0] = 2;
// Numéro d'alarme à
Filter.dwMsgNr[1] = 2;

ret = AXC_SetFilter("gs_alarm_00","Control1",&Filter,&Error);
}
```

1. Désigner le filtre.
2. Sélectionner le type du filtre.
3. Indiquer critères de filtrage.
4. Placer le filtre.

Remarque

Il faut adapter le type de filtre et les critères de filtrage, tous les autres types de filtre sont décrits dans la structure de filtre.

Exemple DeactivateRTProject

```
{
//deactivate the runtime
DeactivateRTProject ();
}
```

Cette fonction désactive le WinCC Runtime.

Exemple ExitWinCC

```
{
//exit wincc
ExitWinCC ();
}
```

Cette fonction quitte WinCC.

2.15.4.2 Exemples - GetAlarmHigh à GetPropChar

Exemple GetAlarmHigh

```
{
double dAlarmHigh;
//Get the Alarm High Limit
dAlarmHigh = GetAlarmHigh(lpszPictureName,"Balken1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetAlarmHigh :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

1. Lire seuil d'alarme supérieur et enregistrer dans le presse-papiers dAlarmHigh.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetBackColor

```
{
long int bk_color;

//Get the backgroundcolor
bk_color = GetBackColor(lpszPictureName,"StatischerText1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetBackColor :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

1. Lire couleur d'arrière-plan actuelle et enregistrer dans le presse-papiers bk_color.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetBorderStyle

```
{
long int lstyle;

//Get the current border style
lstyle = GetBorderStyle(lpszPictureName,"Rechteck1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetBorderStyle :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

1. Lire type de trait actuel de l'objet et enregistrer dans le presse-papiers lstyle.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetFilling

```
{
BOOL bfilling;

//Get the actual state of dynamic filling
bfilling = GetFilling(lpszPictureName,"Rechteck1");

if(bfilling)
{
  // User defined code if the
  // dynamic filling is activated
  ...
}
    Else
{
  // User defined code if the
  // dynamic filling is deactivated
  ...
}
}
```

Paramètres de la fonction GetFilling :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

1. Lire si le remplissage dynamique est activé ou pas, et enregistrer dans le presse-papiers bfilling.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetFillingIndex

```
{
long int filling_index;

//Get the actual filling index of the object
filling_index = GetFillingIndex(lpszPictureName,"Rechteck1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetFillingIndex :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

1. Lire taux de remplissage actuel de l'objet et enregistrer dans le presse-papiers filling_index.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetFillStyle

```
{
long int lstyle;

//Get the current fill style
lstyle = GetFillStyle(lpszPictureName,"Rechteck1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetFillStyle :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

1. Lire motif de remplissage actuel de l'objet et enregistrer dans le presse-papiers lstyle.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetFlashBackColor

```
{
BOOL bflash_col;

//Get if the flashing is on or off
bflash_col = GetFlashBackColor(lpszPictureName,"Gruppe1");

if(bflash_col)
{
  // User defined code if the
  // flashing is activated
  ...
}
    Else
{
  // User defined code if the
  // flashing is deactivated
  ...
}
}
```

Paramètres de la fonction GetFlashBackColor :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Gruppe1" est le nom de l'objet.

1. Lire si le clignotement de la couleur d'arrière-plan est activé ou pas, et enregistrer dans le presse-papiers bflash_col.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetFlashBackColorOn

```
{
long int flashcol_on;

//Get the BackFlashColor
```

```
flashcol_on = GetBackFlashColorOn(lpszPictureName,"Gruppe1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetBackFlashColorOn :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Gruppe1" est le nom de l'objet.

1. Lire la couleur de clignotement d'arrière-plan pour l'état "On" de l'objet et enregistrer dans le presse-papiers flashcol_on.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetFlashRateFlashPic

```
{
long lFlashRate;

//Get the flashrate
lFlashRate = GetFlashRateFlashPic(lpszPictureName,"Zustandsanzeige1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetFlashRateFlashPic :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Zustandsanzeige1" est le nom de l'objet.

1. Lire fréquence de clignotement actuelle de l'objet et enregistrer dans le presse-papiers lFlashRate.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetFocus

```
{
char* pszValue = NULL;
```

```
char szValue[_MAX_PATH+1];


//Get the Object which has the focus
pszValue = Get_Focus();

//Copy the string
if(pszValue != NULL)
{
  strncpy(szValue,pszValue,_MAX_PATH);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

1. Lire sur quel objet se trouve le focus et enregistrer dans le presse-papiers pszValue.
2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szValue. _MAX_PATH caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetFontBold

```
{
BOOL bbold;

//Get if the text is bold
bbold = GetFontBold(lpszPictureName,"StatischerText1");

if(bbold)
{
  // User defined code if the
  // font is bold
  ...
}
    Else
{
  // User defined code if the
  // font is not bold
  ...
}
}
```

Paramètres de la fonction GetFontBold :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

1. Lire si le texte est en gras ou pas, et enregistrer dans le presse-papiers bbold.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetFontSize

```
{
long int fontsize;

//Get the actual Font size
fontsize = GetFontSize(lpszPictureName,"StatischerText1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetFontSize :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

1. Lire taille de police actuelle et enregistrer dans le presse-papiers fontsize.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetHeight

```
{
long lHeight;

//Get the height of the object
lHeight = GetHeight(lpszPictureName,"WinCCLogo");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetHeight :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

1. Lire la hauteur actuelle de l'objet et enregistrer dans le presse-papiers lHeight.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetHiddenInput

```
{
BOOL bHiddenInput;

//Get the state of hidden input
bHiddenInput = GetHiddenInput(lpszPictureName,"EAFeld1");

if(bHiddenInput)
{
  // User defined code if the
  // hidden input is activated
  ...
}
    Else
{
  // User defined code if the
  // hidden input is activated
  ...
}
}
```

Paramètres de la fonction GetHiddenInput :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAFeld1" est le nom de l'objet.

1. Lire si le texte est en gras ou pas, et enregistrer dans le presse-papiers bHiddenInput.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetLanguage

```
{
DWORD rt_language;

//Get the current language
rt_language = GetLanguage ();

//User defined code where the
//user can do something with the returnvalue
...
}
```

1. Lire la langue Runtime actuelle et enregistrer dans le presse-papiers rt_language.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetLeft

```
{
long lPos;

//Get the x-position of the object
lPos = GetLeft(lpszPictureName,"WinCCLogo");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetLeft :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

1. Lire la position X actuelle de l'objet et enregistrer dans le presse-papiers lPos.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetLink

```
{
LINKINFO linkinfo;

//Get the linked Tag
GetLink(lpszPictureName,"Balken1","Process",&linkinfo);

// linkinfo.szLinkName is the tag name
// linkinfo.dwCycle is the update cycle
// linkinfo.LinkType is the type of the connection


//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetLink :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

"Process" est la propriété qui est reliée à une variable.

"&linkinfo" est l'adresse de la structure linkinfo.

1. Remplit la structure transmise linkinfo avec les informations de la liaison de variable.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetLinkedVariable

```
{
char* pszVarName = NULL;
char szVarName[_MAX_PATH+1];

//Get the TagName
pszVarName = GetLinkedVariable("gs_stand_graph_00","StatischerText6","Visible");

//Copy the string
if (strcmp (pszVarName,"")!= 0)
{
  strncpy(szVarName,pszVarName,_MAX_PATH);
}
else printf("Das Attribut 'visible' ist nicht dynamisiert\r\n");
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetLinkedVariable :

"gs_stand_graph_00" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText6" est le nom de l'objet.

"Visible" est la propriété qui est reliée à une variable.

1. Enregistrer la valeur de retour de la fonction GetLinkedVariable dans le presse-papiers pszVarName.
2. Si une valeur valable a été renvoyée, enregistrez la valeur de retour dans szVarName. _MAX_PATH caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetLocalPicture

```
{
char* pszPicName = NULL;
char szPicName[_MAX_PATH+1];

//Get the Local Picture
pszPicName = GetLocalPicture(lpszPictureName);

//Copy the string
if (pszPicName != NULL)
{
  strncpy(szPicName,pszPicName,_MAX_PATH);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

1. Enregistrer la valeur de retour de la fonction GetLocalPicture dans le presse-papiers pszPicName.
2. Si une valeur valable a été renvoyée, enregistrez la valeur de retour dans szPicName. _MAX_PATH caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetMarker

```
{
BOOL bmarker;

//Get the state of the Marker
bmarker = GetMarker(lpszPictureName,"Balken1");

if(bmarker)
{
  // User defined code if the
  // marker is activated
  ...
}
    Else
{
  // User defined code if the
  // marker is deactivated
  ...
}
}
```

Paramètres de la fonction GetMarker :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

1. Lire si le Marker est affiché ou pas, et enregistrer dans le presse-papiers bmarker.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetOutputValueDouble

```
{
double doutput;

//Get the output value of the EA Field 1
doutput = GetOutputValueDouble(lpszPictureName,"EAFeld1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetOutputValueDouble :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAFeld1" est le nom de l'objet.

1. Lire la valeur de sortie et enregistrer dans le presse-papiers doutput.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetParentPicture

```
{
char* pszPicName = NULL;
char szPicName[_MAX_PATH+1];

//Get the parent picture
pszPicName = GetParentPicture(lpszPictureName);

//Copy the string
if (pszPicName != NULL)
{
  strncpy(szPicName,pszPicName,_MAX_PATH);
}
//User defined code where the
//user can do something with the returnvalue
```

```
...
}
```

1. Enregistrer la valeur de retour de la fonction GetParentPicture dans le presse-papiers pszPicName.
2. Si une valeur valable a été renvoyée, enregistrez la valeur de retour dans szPicName. _MAX_PATH caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetPictureDown

```
{
char* pszPicName = NULL;
char szPicName[_MAX_PATH+1];

//Get the current picture name
pszPicName = GetPictureDown(lpszPictureName,"Rundbutton1");

if (pszPicName != NULL)
{
//Copy the string
  strncpy(szPicName,pszPicName,_MAX_PATH);
}

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetPictureDown :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rundbutton1" est le nom de l'objet.

1. Lire le nom de la vue affichée dans le Rundbutton1, et enregistrer dans le presse-papiers pszPicName.
2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szPicName. _MAX_PATH caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetPictureName

```
{
char* pszPictureName = NULL;
char szPictureName[_MAX_PATH + 1];

//Get the current PictureName
pszPictureName = GetPictureName(lpszPictureName,"GraphikObjekt1");

if(pszPictureName != NULL)
{
//copy the string
strncpy(szPictureName,pszPictureName,_MAX_PATH);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetPictureName :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"GraphikObjekt1" est le nom de l'objet.

1. Lire le nom de la vue affichée dans le GraphikObjekt1, et enregistrer dans le presse-papiers pszPictureName.
2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szPictureName. _MAX_PATH caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetPictureUp

```
{
char* pszPicName = NULL;
char szPicName[_MAX_PATH+1];

//Get the current picture name
pszPicName = GetPictureUp(lpszPictureName,"Rundbutton1");

if (pszPicName != NULL)
{
//Copy the string
  strncpy(szPicName,pszPicName,_MAX_PATH);
}
```

```
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetPictureUp:

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rundbutton1" est le nom de l'objet.

1. Lire le nom de la vue affichée dans le Rundbutton1, et enregistrer dans le presse-papiers pszPicName.
2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szPicName. _MAX_PATH caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetPosition

```
{
long int lpos;

//Get the actual position of the Slider
lpos = GetPosition(lpszPictureName,"Control1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetPosition :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Control1" est le nom de l'objet.

1. Lire la position du curseur actuelle et enregistrer dans le presse-papiers lpos.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetPropBOOL

```
{
BOOL bProp;

//Get the property Visible
bProp = GetPropBOOL("gs_graph_eafield","EAFeld1","Visible");

if(bProp)
{
  // User defined code if the
  // objekt is visible
  ...
}
else
{
  // User defined code if the
  // objekt is not visible
  ...
}
}
```

Paramètres de la fonction GetVisible :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAFeld1" est le nom de l'objet.

"Visible" est la propriété de l'objet.

1. Lire si l'objet est visible ou pas, et enregistrer dans le presse-papiers bProp.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetPropChar

```
{
char* pszProp = NULL;
char szProp[14];

//Get the property Tooltiptext
pszProp = GetPropChar("lpszPictureName","EAFeld1","Tooltiptext");

if(pszProp != NULL)
{
//Copy the string
strncpy(szProp,pszProp,13);
}
//User defined code where the
//user can do something with the returnvalue
```

```
...
}
```

Paramètres de la fonction GetPropChar :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAFeld1" est le nom de l'objet.

"Tooltiptext" est la propriété de l'objet.

1. Lire le texte info-bulle de l'objet et enregistrer dans le presse-papiers pszProp.
2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szProp. 13 caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

2.15.4.3 Exemples - GetRangeMax à GetWidth

Exemple GetRangeMax

```
{
long int lrange;

//Get the upper scale Limit
lrange = GetRangeMax(lpszPictureName,"Control1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetRangeMax :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Control1" est le nom de l'objet.

1. Lire la limite supérieure actuelle de l'objet et enregistrer dans le presse-papiers lrange.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetRangeMin

```
{
long int lrange;

//Get the lower scale Limit
lrange = GetRangeMin(lpszPictureName,"Control1");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetRangeMin :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Control1" est le nom de l'objet.

1. Lire la limite inférieure actuelle de l'objet et enregistrer dans le presse-papiers lrange.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetScaling

```
{
BOOL bscaling;

//Get the Scaling state
bscaling = GetScaling(lpszPictureName,"Balken1");

if (bscaling)
{
  // User defined code if the
  // bar object has an additional scale
  ...
}
    Else
{
  // User defined code if the
  // bar object has no additional scale
  ...
}
}
```

Paramètres de la fonction GetScaling :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

1. Lire si l'échelle de la barre est affichée ou pas, et enregistrer dans le presse-papiers bscaling.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetServerTagPrefix

```
{
char* pszServerPrefix;
char* pszTagPrefix;
char* pszWindowPrefix;
int nServerPrefixLen = 0;
int nTagPrefixLen = 0;
int nTagLen = 0;
char myTagName[MAX_DM_VAR_NAME+1];

//Initialize the return value
memset(myTagName,0,MAX_DM_VAR_NAME + 1);

//Get the serverprefix the tagprefix and the windowprefix
GetServerTagPrefix(&pszServerPrefix, &pszTagPrefix, &pszWindowPrefix);

//If a serverprefix exists
if (pszServerPrefix)
{
  //Get the length of the string
  nServerPrefixLen = strlen(pszServerPrefix);
}
    Else
{
  printf("No server prefix was returned.");
return;
}

//If a tagprefix exists
if (pszTagPrefix)
{
  //Get the length of the string
  nTagPrefixLen = strlen(pszTagPrefix);
}

//Get the length of the tag
nTagLen = strlen("TagName");

//Check if the lenght of the
//ServerPrefix+TagPrefix+VarName + the double points < MAX_DM_VAR_NAME)
if (nServerPrefixLen + nTagPrefixLen + nTagLen+2 < MAX_DM_VAR_NAME)
{
  sprintf(myTagName,"%s::%s%s",pszServerPrefix,pszTagPrefix,"TagName");
//User defined code where the
```

```
//user can do something with the returnvalue
  ...
}
    Else
{
  printf("The resulting string is too long.");
return;
}
}
```

1. Initialiser la variable myTagName.
2. Lire le préfixe serveur, le préfixe de variable et le préfixe Window.
3. Si aucun préfixe de serveur a été renvoyé, un texte est sorti et la fonction est quittée.
4. Si un préfixe de serveur a été renvoyé, déterminer sa langueur et enregistrer dans le presse-papiers nServerPrefixLen.
5. Si un préfixe de variable a été renvoyé, déterminer sa langueur et enregistrer dans le presse-papiers nTagPrefixLen.
6. Déterminer la longueur du nom de variable et enregistrer dans le presse-papiers nVarLen.
7. Si la langueur admissible pour des noms de variable est dépassée, un texte est sorti et la fonction est quittée.
8. Si la langueur admissible pour des noms de variable n'est pas dépassée, le nom de variable nécessaire à un environnement Client est constitué.
9. Exécution du code personnalisé pour traitement des valeurs de retour.

Exemple GetServerTagPrefix

```
{
char* pszServerPrefix;
char* pszTagPrefix;
char* pszWindowPrefix;

//Get the serverprefix and the tagprefix
GetServerTagPrefix(&pszServerPrefix, &pszTagPrefix, &pszWindowPrefix);
//User defined code where the
//user can do something with the returnvalue
...

}
```

Paramètres de la fonction GetServerTagPrefix :

"pszServerPrefix" est la variable dans laquelle est écrit le préfixe de serveur.

"pszTagPrefix" est la variable dans laquelle est écrit le préfixe de variable.

"pszWindowPrefix" est la variable dans laquelle est écrit le préfixe Window.

1. Lire le préfixe serveur, le préfixe de variable et le préfixe Window.
2. Dans pszServerPrefix est écrit le préfixe de serveur renvoyé.
3. Dans pszTagPrefix est écrit le préfixe de variable renvoyé.
4. Dans pszWindowPrefix est écrit le préfixe Window renvoyé.
5. Exécution du code personnalisé pour traitement des valeurs de retour.

Exemple GetTagBit

```
{
BOOL bstate;

//Get the current state of the tag
bstate = GetTagBit("gs_tag_bit");

if(bstate)
{
  // User defined code if the
  // value of the tag is true
  ...
}
else
{
  // User defined code if the
  // value of the tag is false
  ...
}
}
```

Paramètres de la fonction GetTagBit :

"gs_tag_bit" est le nom de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers bstate.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetTagBitStateQC

```
{
```

```
DWORD dwState;
DWORD dwQC;
BOOL bValue;

dwState = 0xFFFFFFFF;

//Get the tag value
//dwstate is the tag state
bValue = GetTagBitStateQCWait("gs_tag_bit",&dwState,&dwQC);

//Create a string which includes the tag value
if (bValue)
{
  // User defined code if the
  // value of the tag is true
  ...
}
else
{
  // User defined code if the
  // value of the tag is false
  ...
}
}
```

Paramètres de la fonction GetTagBitStateQC :

"gs_tag_bit" est le nom de la variable.

"&dwState" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

"&dwQC" est l'adresse de la variable dans laquelle doit être inscrit le code qualité.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers bValue. La fonction écrit l'état de la variable dans dwState et le code qualité dans dwQC.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetTagBitStateWait

```
{
DWORD dwstate;
BOOL bValue;

dwstate = 0xFFFFFFFF;

//Get the tag value
//dwstate is the tag state
bValue = GetTagBitStateWait("gs_tag_bit",&dwstate);

//Create a string which includes the tag value
```

```
if (bValue)
{
  // User defined code if the
  // value of the tag is true
  ...
}
else
{
  // User defined code if the
  // value of the tag is false
  ...
}
}
```

Paramètres de la fonction GetTagBitStateWait :

"gs_tag_bit" est le nom de la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers bstate. La fonction écrit l'état de variable dans dwstate.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetTagChar

```
{
char* pszValue = NULL;
char szValue[13];

//Get the current value of the tag
pszValue = GetTagChar("gs_tag_char");

if(pszValue != NULL)
{
//Copy the string
strncpy(szValue,pszValue,12);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagChar :

"gs_tag_char" est le nom de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers pszValue.

2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szValue. 12 caractères sont enregistrés au maximum.

3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagCharStateQCWait

```
{
DWORD dwState;
DWORD dwQC;
char* pszRetValue = NULL;
char szRetValue[13];

dwState = 0xFFFFFFFF;

//Get the tag value
pszRetValue = GetTagCharStateQCWait("gs_tag_char",&dwState, &dwQC);

if (pszRetValue != NULL)
{
//Copy the string
strncpy(szRetValue,pszRetValue,12);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagCharStateQCWait :

"gs_tag_char" est le nom de la variable.

"&dwState" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

"&dwQC" est l'adresse de la variable dans laquelle doit être inscrit le code qualité.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers pszRetValue. La fonction écrit l'état de la variable dans dwState et le code qualité dans dwQC.

2. Lorsqu'une valeur valable a été restituée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szRetValue. 12 caractères sont enregistrés au maximum.

3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagCharStateWait

```
{
DWORD dwstate;
char szValue[11];
char* pszRetValue = NULL;
```

```
char szRetValue[13];

dwstate = 0xFFFFFFFF;
//Get the tag value
//dwstate is the tag state
pszRetValue = GetTagCharStateWait("gs_tag_char",&dwstate);

if (pszRetValue != NULL)
{
//Copy the string
strncpy(szRetValue,pszRetValue,12);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagCharStateWait :

"gs_tag_char" est le nom de la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers pszRetValue. La fonction écrit l'état de variable dans dwstate.
2. Lorsqu'une valeur valable a été restituée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szRetValue. 12 caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagFloat

```
{
float fValue;

//Get the current value of the tag
fValue = GetTagFloat("gs_tag_float");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagFloat :

"gs_tag_float" est le nom de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers fValue.

2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagFloatStateQCWait

```
{
DWORD dwState;
DWORD dwQC;
float fValue;

dwState = 0xFFFFFFFF;

//Get the tag value
fValue = GetTagFloatStateQCWait("gs_tag_float",&dwState, &dwQC);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagFloatStateQCWait :

"gs_tag_float" est le nom de la variable.

"&dwState" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

"&dwQC" est l'adresse de la variable dans laquelle doit être inscrit le code qualité.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers fValue. La fonction écrit l'état de la variable dans dwState et le code qualité dans dwQC.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagFloatStateWait

```
{
DWORD dwstate;
float fValue;

dwstate = 0xFFFFFFFF;
//Get the tag value
//dwstate is the tag state
fValue = GetTagFloatStateWait("gs_tag_float",&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagFloatStateWait :

"gs_tag_float" est le nom de la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers fValue. La fonction écrit l'état de variable dans dwstate.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagMultiStateQCWait

```
{
#define DATA_SIZE 5
DWORD dwState[DATA_SIZE];
DWORD dwQC[DATA_SIZE];

//define all Datas
BOOL lValue1;
long lValue2 ;
char* szValue3;
double dblValue4 ;
WORD lValue5 ;


//Set the tags
GetTagMultiStateQCWait(dwState,dwQC,"%d%d%s%f%d",
  "gs_tag_bit",&lValue1,
  "gs_tag_SByte",&lValue2,
  "gs_tag_char",&szValue3,
  "gs_tag_float",&dblValue4,
  "gs_tag_word",&lValue5);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagMultiStateWait :

"dwState" est la DWord-Array dans laquelle les états de variable sont enregistrés.

"dwState" est la DWord-Array dans laquelle les états de variable sont enregistrés.

"%d%d%s%f%d" sont les descriptions de type de la variable à lire.

"gs_tag_bit" est la variable devant être lue.

"&lValue1" est l'adresse de la variable dans laquelle la valeur de la variable gs_tag_bit doit être écrite.

"gs_tag_SByte" est la variable devant être lue.

"&lValue2" est l'adresse de la variable dans laquelle la valeur de la variable gs_tag_SByte doit être écrite.

Traiter les autres paramètres de façon similaire aux autres, comme décrit plus haut.

1. Création d'une DWord-Array de taille requise (nombre de variables).
2. Lire et enregistrer les valeurs des variables dans le presse-papiers. La valeur des variables gs_tag_bit est enregistrée dans le presse-papiers lValue1. La valeur des variables gs_tag_SByte est enregistrée dans le presse-papiers lValue2.
3. Exécution du code personnalisé pour traitement des valeurs de retour.

Exemple GetTagMultiStateWait

```
{
#define DATA_SIZE 5
DWORD dwData[DATA_SIZE];

//define all Datas
BOOL lValue1;
long lValue2 ;
char* szValue3;
double dblValue4 ;
WORD lValue5 ;


//Set the tags
GetTagMultiStateWait(dwData,"%d%d%s%f%d",
  "gs_tag_bit",&lValue1,
  "gs_tag_SByte",&lValue2,
  "gs_tag_char",&szValue3,
  "gs_tag_float",&dblValue4,
  "gs_tag_word",&lValue5);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagMultiStateWait :

"dwState" est la DWord-Array dans laquelle les états de variable sont enregistrés.

"%d%d%s%f%d" sont les descriptions de type de la variable à lire.

"gs_tag_bit" est la variable devant être lue.

"&lValue1" est l'adresse de la variable dans laquelle la valeur de la variable gs_tag_bit doit être écrite.

"gs_tag_SByte" est la variable devant être lue.

"&lValue2" est l'adresse de la variable dans laquelle la valeur de la variable gs_tag_SByte doit être écrite.

Traiter les autres paramètres de façon similaire aux autres, comme décrit plus haut.

1. Création d'une DWord-Array de taille requise (nombre de variables).
2. Lire et enregistrer les valeurs des variables dans le presse-papiers. La valeur des variables gs_tag_bit est enregistrée dans le presse-papiers lValue1. La valeur des variables gs_tag_SByte est enregistrée dans le presse-papiers lValue2.
3. Exécution du code personnalisé pour traitement des valeurs de retour.

Exemple GetTagMultiWait

```
DWORD dwVar1Value;
char* szVar2Value;
//Speicher für den Variablenwert wird
//durch die Funktion mit SysMalloc angelegt
double dbVar3Value;

BOOL ok;

ok=GetTagMultiWait("%d%s%f", "Ernie_word", &dwVar1Value,
  "Ernie_char", &szVar2Value,
  "Ernie_double", &dbVar3Value);

printf("Word %d, String %s, Double %f\r\n",
  dwVar1Value, szVar2Value, dbVar3Value);
```

Exemple GetTagPrefix

```
{
char* pszTagPrefix = NULL;
char szTagPrefix[7];

//Get the current tag prefix
pszTagPrefix = GetTagPrefix(lpszPictureName,"Bildfenster1");

if(pszTagPrefix != NULL)
{
//Copy the string
strncpy(szTagPrefix,pszTagPrefix,6);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagPrefix :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Bildfenster1" est le nom de l'objet.

1. Lire préfixe de variable actuel de fenêtre de vue Bildfenster1 et enregistrer dans le presse-papiers pszTagPrefix.
2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szTagPrefix. 6 caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagRaw

```
{
#define DATA_SIZE 3

BYTE byData[DATA_SIZE];

//Get the current values of the tag
GetTagRaw("gs_tag_raw",byData,DATA_SIZE);

//Use the values received in the array byData
...
}
```

Paramètres de la fonction GetTagRaw :

"gs_tag_raw" est le nom de la variable.

"byData" est le Byte-Array, dans lequel sont enregistrées les valeurs des variables de données brutes.

"DATA_SIZE" est le nombre de valeurs qui sont lues.

1. Lecture des valeurs des variables et enregistrement dans le presse-papiers byData.
2. Exécution du code personnalisé pour traitement des valeurs de retour.

Exemple GetTagRawStateQCWait

```
{
#define DATA_SIZE 3
DWORD dwState;
DWORD dwQC;
```

```
BYTE byData[DATA_SIZE];

dwState = 0xFFFFFFFF;

//Get the values of the tag
GetTagRawStateQCWait("gs_tag_raw",byData,DATA_SIZE,&dwState,&dwQC);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagRawStateQCWait :

"gs_tag_raw" est le nom de la variable.

"byData" est le Byte-Array, dans lequel sont enregistrées les valeurs des variables de données brutes.

"DATA_SIZE" est le nombre de valeurs qui sont lues.

"&dwState" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

"&dwQC" est l'adresse de la variable dans laquelle doit être inscrit le code qualité.

1. Lecture des valeurs des variables et enregistrement dans le presse-papiers byData.
2. Exécution du code personnalisé pour traitement des valeurs de retour.

Exemple GetTagRawStateWait

```
{
#define DATA_SIZE 3
DWORD dwstate;
BYTE byData[DATA_SIZE];
char szValue[11];

//Load dwState with default values
dwstate = 0xFFFFFFFF;

//Get the values of the tag
//dwstate is the tag state
GetTagRawStateWait("gs_tag_raw",byData,DATA_SIZE,&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagRawStateWait :

"gs_tag_raw" est le nom de la variable.

"byData" est le Byte-Array, dans lequel sont enregistrées les valeurs des variables de données brutes.

"DATA_SIZE" est le nombre de valeurs qui sont lues.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Lecture des valeurs des variables et enregistrement dans le presse-papiers byData.
2. Exécution du code personnalisé pour traitement des valeurs de retour.

Exemple GetTagSByte

```
{
long lValue;

//Get the current value of the tag
lValue = GetTagSByte("gs_tag_SByte");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagSByte :

"gs_tag_SByte" est le nom de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers lValue.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagSByteStateQCWait

```
{
DWORD dwState;
DWORD dwQC;
long lValue;

dwState = 0xFFFFFFFF;

//Get the tag value
lValue = GetTagSByteStateQCWait("gs_tag_SByte",&dwState, &dwQC);

//User defined code where the
//user can do something with the returnvalue
...
```

```
}
```

Paramètres de la fonction GetTagSByteStateQCWait :

"gs_tag_SByte" est le nom de la variable.

"&dwState" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

"&dwQC" est l'adresse de la variable dans laquelle doit être inscrit le code qualité.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers lValue. La fonction écrit l'état de la variable dans dwState et le code qualité dans dwQC.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagSByteStateWait

```
{
DWORD dwstate;
long lValue;

dwstate = 0xFFFFFFFF;
//Get the tag value
//dwstate is the tag state
lValue = GetTagSByteStateWait("gs_tag_SByte",&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagSByteStateWait :

"gs_tag_SByte" est le nom de la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers lValue. La fonction écrit l'état de variable dans dwstate.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagWord

```
{
WORD wValue;
```

```
//Get the current value of the tag
wValue = GetTagWord("gs_tag_word");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagWord :

"gs_tag_word" est le nom de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers wValue.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagWordStateQCWait

```
{
DWORD dwState;
DWORD dwQC;
WORD wValue;

dwState = 0xFFFFFFFF;

//Get the tag value
wValue = GetTagWordStateQCWait("gs_tag_word",&dwState, &dwQC);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagWordStateQCWait :

"gs_tag_word" est le nom de la variable.

"&dwState" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

"&dwQC" est l'adresse de la variable dans laquelle doit être inscrit le code qualité.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers wValue. La fonction écrit l'état de la variable dans dwState et le code qualité dans dwQC.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTagWordStateWait

```
{
DWORD dwstate;
WORD wValue;

dwstate = 0xFFFFFFFF;
//Get the tag value
//dwstate is the tag state
wValue = GetTagWordStateWait("gs_tag_word",&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTagWordStateWait :

"gs_tag_word" est le nom de la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Lecture de la valeur de la variable et enregistrement dans le presse-papiers wValue. La fonction écrit l'état de variable dans dwstate.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetText

```
{
char* pszValue = NULL;
char szValue[13];

//Get the Text which is actually set
pszValue = GetText(lpszPictureName,"StatischerText1");

if(pszValue != NULL)
{
//Copy the string
strncpy(szValue,pszValue,12);
}
//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetText :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

1. Lecture du texte qui se trouve dans l'objet StatischerText1 et enregistrement dans le presse-papiers pszValue.
2. Lorsqu'une valeur valable a été renvoyée, enregistrez la valeur de retour de la fonction dans la chaîne de caractères locale szValue. 12 caractères sont enregistrés au maximum.
3. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetTop

```
{
long lPos;

//Get the y-Position of the Object
lPos = GetTop(lpszPictureName,"WinCCLogo");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetTop :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

1. Lire la position Y actuelle de l'objet et enregistrer dans le presse-papiers lPos.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

Exemple GetVisible

```
{
BOOL bVisible;

//Get the visibility
bVisible = GetVisible(lpszPictureName,"GraphikObjekt1");

if(bVisible)
{
  // User defined code if the
  // objekt is visible
  ...
}
else
```

```
{
  // User defined code if the
  // objekt is not visible
  ...
}
}
```

Paramètres de la fonction GetVisible :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"GraphikObjekt1" est le nom de l'objet.

1. Lire si l'objet est visible ou pas, et enregistrer dans le presse-papiers bVisible.
2. Exécution du code personnalisé, selon la valeur de retour de la fonction.

Exemple GetWidth

```
{
long lWidth;

//Get the width of the object
lWidth = GetWidth(lpszPictureName,"WinCCLogo");

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction GetWidth :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

1. Lire la largeur actuelle de l'objet et enregistrer dans le presse-papiers lWidth.
2. Exécution du code personnalisé pour traitement de la valeur de retour.

2.15.4.4 Exemples - H à S

Exemple InquireLanguage

```
{
```

```
DWORD count;
DWORD* language;
int i;

//Count the installed languages
language = InquireLanguage(&count);

printf("##################### INQUIRE LANGUAGE ####################");
//Print out the count of languages
printf ( "\r\nCount Languages=%d\r\n", count );

//print out which languages are installed
for (i=1;i<=count; i++)
{
printf ("\r\n%d.language=%x", i,*language++);
}
}
```

1. Déterminer les langues configurées pour le temps d'exécution. Dans language, les identificateurs de langue sont enregistrés en mémoire intermédiaire. Dans count, le nombre des langues est enregistré en mémoire intermédiaire.
2. Le nombre des langues déterminées est sorti.
3. Tous les identificateurs de langue déterminés sont sortis.

Exemple ProgramExecute

```
{
//start the program calc.exe
ProgramExecute("C:\\Winnt\\system32\\calc.exe");
}
```

Le fichier avec son chemin est à indiquer comme paramètre.

Exemple ResetFilter

```
{
BOOL ret;
MSG_FILTER_STRUCT Filter;
CMN_ERROR Error;

//delete the whole Filter struct
memset( &Filter, 0, sizeof( MSG_FILTER_STRUCT ) );

//set an empty filter struct
AXC_SetFilter("gs_alarm_00","Control1",&Filter,&Error);
```

```
}
```

1. Supprimer la structure de filtre.
2. Décrire la structure de filtre avec des valeurs inoccupées.

Exemple RPTJobPreview

```
{
//Start the print preview of the specified print job
RPTJobPreview("Documentation Text Library");
}
```

Paramètres de la fonction "RPTJobPreview" :

"Documentation Text Library" est le nom de la tâche d'impression.

Exemple RPTJobPrint

```
{
//Print the specified print job out
RPTJobPrint("@Text library (compact)");
}
```

Paramètres de la fonction RPTJobPrint :

@Text library (compact) est le nom du travail d'impression.

Exemple SysMalloc

```
char* main(...);
{
char* returnwert;
char text[17];
returnwert=SysMalloc(17);
strcpy(returnwert,&text[0];
return returnwert;
}
```

2.15.4.5 Exemples - SetAlarmHigh à SetPropChar

Exemple SetAlarmHigh

```
{
//Set the upper limit for the warning
SetAlarmHigh(lpszPictureName,"Balken1",3.0);
}
```

Paramètres de la fonction SetAlarmHigh :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

"3.0" est la valeur sur laquelle est mis le seuil d'alarme supérieur.

Exemple SetBackColor

```
{
//Set the back color blue
SetBackColor(lpszPictureName,"StatischerText1",CO_BLUE);
}
```

Paramètres de la fonction SetBackColor :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

"CO_BLUE" est la constante pour la couleur bleu.

Remarque

Au lieu de la constante pour la valeur de couleur, vous pouvez également exprimer la couleur par une valeur hexadécimale.

Exemple SetBorderEndStyle

```
{
SetBorderEndStyle(lpszPictureName,"Line", (2|393216));
}
```

Met la fin de ligne gauche comme flèche pleine et la fin de ligne droite comme cercle plein. La fin de ligne gauche est enregistrée dans les deux octets inférieurs, la fin de ligne droite dans les deux octets supérieurs. Les paramètres sont transmis à l'aide de valeurs chiffrées.

Exemple pour mettre les fins de trait par des mnémoniques

```
{
SetBorderEndStyle(lpszPictureName,"Line", (LE_FULL_ARROW|(LE_FULL_CIRCLE <<16));
}
```

Met la fin de ligne gauche comme flèche pleine et la fin de ligne droite comme cercle plein. La fin de ligne gauche est enregistrée dans les deux octets inférieurs, la fin de ligne droite dans les deux octets supérieurs. Pour adresser la fin de ligne droite, le mnémonique "LE_FULL_CIRCLE" est décalé de 2 octets ou 16 bits dans les octets supérieurs.

Exemple SetBorderStyle

```
{
//Change the Border style
SetBorderStyle(lpszPictureName,"Rechteck1",3);
}
```

Paramètres de la fonction SetBorderStyle :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

"3" est le type de trait qui est réglé pour l'objet.

Exemple SetColorAlarmHigh

```
{
//Set the Color for the alarm high limit to red
SetColorAlarmHigh(lpszPictureName,"Balken1",CO_RED);
}
```

Paramètres de la fonction SetColorAlarmHigh :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

"CO_RED" est la constante pour la couleur rouge.

Remarque

Au lieu de la constante pour la valeur de couleur, vous pouvez également exprimer la couleur par une valeur hexadécimale.

Exemple SetCursorMode

```
{
//Set the Cursor Mode to Alpha cursor
SetCursorMode(lpszPictureName,"GraphikObjekt1",FALSE);
}
```

Paramètres de la fonction SetCursorMode :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"GraphikObjekt1" est le nom de l'objet.

"FALSE" signifie : Le mode de curseur "Alpha-Cursor" est paramétré.

Exemple SetFilling

```
{
//Set the dynamic filling true
SetFilling(lpszPictureName,"Rechteck1",TRUE);
}
```

Paramètres de la fonction SetFilling :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

"TRUE" signifie : Activer le remplissage dynamique.

Exemple SetFillingIndex

```
{
//Set the Filling of the Rechteck1 to 10
SetFillingIndex(lpszPictureName,"Rechteck1",10);
}
```

Paramètres de la fonction SetFillingIndex :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

"10" est le taux de remplissage qui est affecté à l'objet.

Exemple SetFillStyle

```
{
//Change the fill style
SetFillStyle(lpszPictureName,"Rechteck1",196617);
}
```

Paramètres de la fonction SetFillStyle :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rechteck1" est le nom de l'objet.

"196617" est le motif de remplissage (mur en briques) qui est réglé pour l'objet.

Exemple SetFlashBackColor

```
{
//Set the flashing to True
SetFlashBackColor(lpszPictureName,"Gruppe1",TRUE);
}
```

Paramètres de la fonction SetFlashBackColor :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Gruppe1" est le nom de l'objet.

"TRUE" signifie : Activer le clignotement de la couleur d'arrière-plan.

Exemple SetFlashBackColorOn

```
{
//Set the Flasch color for the state on to red
SetBackFlashColorOn(lpszPictureName,"Gruppe1",CO_RED);
}
```

Paramètres de la fonction SetBackFlashColorOn :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Gruppe1" est le nom de l'objet.

"CO_Red" est la constante pour la couleur rouge.

Remarque

Au lieu de la constante pour la valeur de couleur, vous pouvez également exprimer la couleur par une valeur hexadécimale.

Exemple SetFlashRateFlashPic

```
{
//Set the flash rate to 0
SetFlashRateFlashPic(lpszPictureName,"Zustandsanzeige1",0);
}
```

Paramètres de la fonction SetFlashRateFlashPic :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Zustandsanzeige1" est le nom de l'objet.

"0" est la fréquence de clignotement avec laquelle doit clignoter l'objet.

Exemple SetFocus

```
{
//Set the Focus on the Object Button 1
Set_Focus(lpszPictureName,"Button1");
}
```

Paramètres de la fonction Set_Focus :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Button1" est le nom de l'objet sur lequel est mis le focus.

Exemple SetFontBold

```
{
//Set the displayed Text bold
SetFontBold(lpszPictureName,"StatischerText1",TRUE);
}
```

Paramètres de la fonction SetFontBold :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

"TRUE" signifie : Ecrire le texte en gras.

Exemple SetFontSize

```
{
//Set Font Size to 12
SetFontSize(lpszPictureName,"StatischerText1",12);
}
```

Paramètres de la fonction SetFontSize :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

"12" est la taille de police réglée pour le texte.

Exemple SetHeight

```
{
//Set the height of the object to 100
SetHeight(lpszPictureName,"WinCCLogo",100);
}
```

Paramètres de la fonction SetHeight :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

"100" est la hauteur réglée pour l'objet.

Exemple SetHiddenInput

```
{
//Set the hidden input true
SetHiddenInput(lpszPictureName,"EAFeld1",TRUE);
}
```

Paramètres de la fonction SetHiddenInput :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAField1" est le nom de l'objet.

"TRUE" signifie : Activer l'entrée cachée.

Exemple SetLanguage

```
{
//German
SetLanguage(0x0407);
}
```

La langue Runtime est réglée à Allemand.

Exemple SetLeft

```
{
//Set the x-position to 0
SetLeft(lpszPictureName,"WinCCLogo",0);
}
```

Paramètres de la fonction SetLeft:

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

"0" est la position X réglée pour l'objet.

Exemple SetLink

```
{
LINKINFO linkinfo;

//Set the link type
linkinfo.LinkType = 1;

//Set the update cycle
linkinfo.dwCycle = 0;
```

```
//set the Structmember
strcpy(linkinfo.szLinkName, "U08i_link_00");

//Set the connection to the tag
SetLink(lpszPictureName,"Balken1","Process",&linkinfo);
}
```

Paramètres de la fonction SetLink :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

"Process" est la propriété qui est reliée à une variable.

"&linkinfo" est l'adresse de la structure linkinfo

1. Régler à Liaison directe le type de liaison à la propriété Process.
2. Mettre le cylcle de rafraîchissement à "En cas de modification".
3. Mettre le nom de variable à U08i_link_00.

Exemple SetMarker

```
{
//Set the marker visible
SetMarker(lpszPictureName,"Balken1",TRUE);
}
```

Paramètres de la fonction SetMarker :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

"TRUE" signifie : Le Marker est affiché.

Exemple SetOutputValueDouble

```
{
//Set the outputvalue of the EA Filed to 55.5
SetOutputValueDouble(lpszPictureName,"EAFeld1",55.5);
}
```

Paramètres de la fonction SetOutputValueDouble :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAField1" est le nom de l'objet.

"55.5" est la valeur qui est sortie.

Exemple SetPictureDown

```
{
//Set the picture name to activated.bmp
SetPictureDown(lpszPictureName,"Rundbutton1","activated.bmp");
}
```

Paramètres de la fonction SetPictureDown :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rundbutton1" est le nom de l'objet.

"activated.bmp" est le nom de la vue qui doit être affichée dans Rundbutton1.

Exemple SetPictureName

```
{
//Set the picture name cool_man.bmp
SetPictureName(lpszPictureName,"GraphikObjekt1","cool_man.bmp");
}
```

Paramètres de la fonction SetPictureName :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"GraphikObjekt1" est le nom de l'objet.

"cool_man.bmp" est le nom de la vue qui doit être affichée dans GraphikObjekt1.

Exemple SetPictureUp

```
{
//Set the picture name to deactivated.bmp
SetPictureUp(lpszPictureName,"Rundbutton1","deactivated.bmp");
}
```

Paramètres de la fonction SetPictureUp :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Rundbutton1" est le nom de l'objet.

"deactivated.bmp" est le nom de la vue qui doit être affichée dans Rundbutton1.

Exemple SetPosition

```
{
//Set the Slider Position to 30
SetPosition(lpszPictureName,"Control1",30);
}
```

Paramètres de la fonction SetPosition:

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Control1" est le nom de l'objet.

"30" est la position à laquelle doit être réglé le curseur.

Exemple SetPropBOOL

```
{
//Set the visibility TRUE
SetPropBOOL("lpszPictureName","EAFeld1","Visible",TRUE);
}
```

Paramètres de la fonction SetVisible :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAFeld1" est le nom de l'objet.

"TRUE" signifie : L'objet doit être visible.

Exemple SetPropChar

```
{
//Set the property Tooltiptext
SetPropChar("gs_graph_eafield","EAFeld1","ToolTipText","Tooltiptext1 ");
}
```

Paramètres de la fonction SetPropChar:

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"EAFeld1" est le nom de l'objet.

"Tooltiptext" est la propriété de l'objet.

"Tooltiptext 1" est la valeur à laquelle doit être réglée la propriété.

2.15.4.6 Exemples - SetRangeMax à SetWidth

Exemple SetRangeMax

```
{
//Set the Upper Scale Limit
SetRangeMax(lpszPictureName,"Control1",80);
}
```

Paramètres de la fonction SetRangeMax :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Control1" est le nom de l'objet.

"80" est le seuil supérieur qui doit être affecté à l'objet.

Exemple SetRangeMin

```
{
//Set the lower Scale Limit
SetRangeMin(lpszPictureName,"Control1",0);
}
```

Paramètres de la fonction SetRangeMin :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Control1" est le nom de l'objet.

"0" est le seuil inférieur qui doit être affecté à l'objet.

Exemple SetScaling

```
{
//Set the Scaling Visible
SetScaling(lpszPictureName,"Balken1",TRUE);

}
```

Paramètres de la fonction SetScaling:

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Balken1" est le nom de l'objet.

"TRUE" signifie : Rendre visible la graduation.

Exemple SetTagBit

```
{
//Set the tag to true
SetTagBit("gs_tag_bit",TRUE);
}
```

Paramètres de la fonction SetTagBit :

"gs_tag_bit" est le nom de la variable.

"TRUE" est la valeur avec laquelle doit être décrite la variable.

Exemple SetTagBitStateWait

```
{
DWORD dwstate;

//Load dwState with default values
dwstate = 0xFFFFFFFF;

//Set the value of the tag to TRUE
//dwstate is the tag state
SetTagBitStateWait("gs_tag_bit",TRUE,&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction SetTagBitStateWait :

"gs_tag_bit" est le nom de la variable.

"TRUE" est la valeur avec laquelle doit être décrite la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Mettre la variable à la valeur indiquée.

2. Exécution du code personnalisé pour traitement de l'état de variable.

Exemple SetTagChar

```
{
//Set the tag to Beispieltext
SetTagChar("gs_tag_char","Example Text");
}
```

Paramètres de la fonction SetTagChar :

"gs_tag_char" est le nom de la variable.

"Example Text" est la valeur avec laquelle doit être décrite la variable.

Exemple SetTagCharStateWait

```
{
DWORD dwstate;

//Load dwState with default values
dwstate = 0xFFFFFFFF;

//Set the tag to Example Text
//dwstate is the tag state
SetTagCharStateWait("gs_tag_char","Example Text",&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction SetTagCharStateWait :

"gs_tag_char" est le nom de la variable.

"Example Text" est la valeur avec laquelle doit être décrite la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Mettre la variable à la valeur indiquée.
2. Exécution du code personnalisé pour traitement de l'état de variable.

Exemple SetTagFloat

```
{
//Set the tag to 55.4711
SetTagFloat("gs_tag_float",55.4711);
}
```

Paramètres de la fonction SetTagFloat :

"gs_tag_float" est le nom de la variable.

"55.4711" est la valeur avec laquelle doit être décrite la variable.

Exemple SetTagFloatStateWait

```
{
DWORD dwstate;
char szValue[9];

//Load dwState with default values
dwstate = 0xFFFFFFFF;

//Set the tag to 55.4711
//dwstate is the tag state
SetTagFloatStateWait("gs_tag_float",55.4711,&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction SetTagFloatStateWait :

"gs_tag_float" est le nom de la variable.

"55.4711" est la valeur avec laquelle doit être décrite la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Mettre la variable à la valeur indiquée.
2. Exécution du code personnalisé pour traitement de l'état de variable.

Exemple SetTagMultiStateWait

```
{
```

```
#define DATA_SIZE 5
DWORD dwData[DATA_SIZE];

//define all tags
BOOL lValue1;
long lValue2 ;
char szValue3[_MAX_PATH];
float lValue4;
char lValue5;

// Fill the tags with the values
// you want to set into the WinCC tags
...

//Set the WinCC tags
SetTagMultiStateWait(dwData,"%d%d%s%f%d","gs_tag_bit",lValue1,
 "gs_tag_SByte",lValue2,
 "gs_tag_char",szValue3,
 "gs_tag_float",lValue4,
 "gs_tag_word",lValue5);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction SetTagMultiStateWait :

"dwState" est la DWord-Array dans laquelle les états de variable sont enregistrés.

"%d%d%s%f%d" sont les descriptions de type de la variable à décrire.

"gs_tag_bit" est la variable WinCC devant être décrite.

"lValue1" est la variable à la valeur de laquelle doit être mise la variable WinCC gs_tag_bit.

"gs_tag_SByte" est la variable WinCC devant être décrite.

"&lValue2" est la variable à la valeur de laquelle doit être mise la variable WinCC gs_tag_SByte.

Traiter les autres paramètres de façon similaire aux autres, comme décrit plus haut.

1. Création d'une DWord-Array de taille requise (nombre de variables).
2. Création de variables avec les valeurs desquelles doivent être décrites les variables WinCC.
3. Décrire les variables WinCC avec les valeurs des variables créées et remplies auparavant.
4. Exécution du code personnalisé pour traitement de l'état de variable.

Exemple SetTagMultiWait

```
BOOL ok;
```

```
ok=SetTagMultiWait("%d%s%f", "Ernie_word", 16,
  "Ernie_char", "Hallo Welt",
  "Ernie_double", 55.4711);
```

Exemple SetTagPrefix

```
{
//Set the TagPrefix to Struct1.
SetTagPrefix(lpszPictureName,"Bildfenster1","Struct1.");

//Set the picture name again to update the tag prefix
SetPictureName(lpszPictureName,"Bildfenster1","gs_graph_eafield");
}
```

Paramètres de la fonction SetTagPrefix :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"Bildfenster1" est le nom de l'objet.

"Struct1." est le préfixe de variable qui doit être mis sur Bildfenster1.

1. Mettre le préfixe de variable de l'objet "Bildfenster1" à "Struct1.".
2. Mettre de nouveau le nom de la vue qui est affichée dans la fenêtre de vue afin que la mise du préfixe de variable devienne valable.

Exemple SetTagRaw

```
{
#define DATA_SIZE 3

BYTE byData[DATA_SIZE];

// Fill the Byte array with the values
// you want to set into the raw data tag
...

//Set the tag to the default values
SetTagRaw("gs_tag_raw",byData,DATA_SIZE);
}
```

Paramètres de la fonction SetTagRaw :

"gs_tag_raw" est le nom de la variable.

"byData" est le Byte-Array avec les valeurs duquel est décrite la variable de données brutes.

"DATA_SIZE" est le nombre des valeurs qui sont écrites.

1. Création d'un BYTE-Array de taille requise (taille de la variable de données brutes).
2. Tirer le Byte-Array avec les valeurs à écrire.
3. Décrire la variable de données brutes avec les valeurs du BYTE-Array.

Exemple SetTagRawStateWait

```
{
#define DATA_SIZE 3

BYTE byData[DATA_SIZE];
DWORD dwstate;
char szValue[9];

//Load dwState with default values
dwstate = 0xFFFFFFFF;

// Fill the Byte array with the values
// you want to set into the raw data tag
...

//Set the tag to the default values
//dwstate is the tag state
SetTagRawStateWait("gs_tag_raw",byData,DATA_SIZE,&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction SetTagRawStateWait :

"gs_tag_raw" est le nom de la variable.

"byData" est le Byte-Array avec les valeurs duquel est décrite la variable de données brutes.

"DATA_SIZE" est le nombre des valeurs qui sont écrites.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Création d'un BYTE-Array de taille requise (taille de la variable de données brutes).
2. Tirer le Byte-Array avec les valeurs à écrire.
3. Décrire la variable de données brutes avec les valeurs du BYTE-Array.
4. Exécution du code personnalisé pour traitement de l'état de variable.

Exemple SetTagSByte

```
{
//Set the tag to 50
SetTagSByte("gs_tag_SByte",50);
}
```

Paramètres de la fonction SetTagSByte :

"gs_tag_SByte" est le nom de la variable.

"50" est la valeur avec laquelle doit être décrite la variable.

Exemple SetTagSByteStateWait

```
{
DWORD dwstate;
char szValue[9];

//Load dwState with default values
dwstate = 0xFFFFFFFF;

//Set the tag to 50
//dwstate is the tag state
SetTagSByteStateWait("gs_tag_SByte",50,&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction SetTagSByteStateWait :

"gs_tag_SByte" est le nom de la variable.

"50" est la valeur avec laquelle doit être décrite la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Mettre la variable à la valeur indiquée.
2. Exécution du code personnalisé pour traitement de l'état de variable.

Exemple SetTagWord

```
{
```

```
//Set the tag to 50
SetTagWord("gs_tag_word",50);
}
```

Paramètres de la fonction SetTagWord :

"gs_tag_word" est le nom de la variable.

"50" est la valeur avec laquelle doit être décrite la variable.

Exemple SetTagWordStateWait

```
{
DWORD dwstate;
char szValue[9];

//Load dwState with default values
dwstate = 0xFFFFFFFF;

//Set the tag to 50
//dwstate is the tag state
SetTagWordStateWait("gs_tag_word",50,&dwstate);

//User defined code where the
//user can do something with the returnvalue
...
}
```

Paramètres de la fonction SetTagWordStateWait :

"gs_tag_word" est le nom de la variable.

"50" est la valeur avec laquelle doit être décrite la variable.

"&dwstate" est l'adresse de la variable dans laquelle doit être inscrit l'état de la variable.

1. Mettre la variable à la valeur indiquée.
2. Exécution du code personnalisé pour traitement de l'état de variable.

Exemple SetText

```
{
//Set the text Beispieltext on the StaticTextfield
SetText(lpszPictureName,"StatischerText1","ExampleText");
}
```

Paramètres de la fonction SetText :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"StatischerText1" est le nom de l'objet.

"ExampleText" est le texte qui doit être sorti.

Exemple SetTop

```
{
//Set the y-position to 0
SetTop(lpszPictureName,"WinCCLogo",140);
}
```

Paramètres de la fonction SetTop :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

"140" est la position Y réglée pour l'objet.

Exemple SetVisible

```
{
//Set the Object visible
SetVisible(lpszPictureName,"GraphikObjekt1",TRUE);
}
```

Paramètres de la fonction SetVisible :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"GraphikObjekt1" est le nom de l'objet.

"TRUE" signifie : L'objet doit être visible.

Exemple SetWidth

```
{
//Set the width of the object to 400
SetWidth(lpszPictureName,"WinCCLogo",400);
}
```

Paramètres de la fonction SetWidth :

"lpszPictureName" est le nom de la vue dans laquelle l'objet a été configuré.

"WinCCLogo" est le nom de l'objet.

"400" est la largeur réglée pour l'objet.

2.15.4.7 Exemples - T à Z

Exemple TlgGetNumberOfColumns

```
{
char text[5];
long int columns

//get number of Columns
columns = GetNumberOfColumns("TableControl_01");

//convert long int to char
sprintf(text,"%d",columns);

//set text on TextField5
SetText(lpszPictureName,"StatischerText5",text);
}
```

Paramètres de la fonction TlgGetNumberOfColumns :

"TableControl_01" est le nom du WinCC Table Contol.

1. Enregistrer la valeur de retour de la fonction TlgGetNumberOfColumns dans le presse-papiers columns.
2. Enregistrer la valeur de retour dans le presse-papiers Chaîne de caractères text
3. Sortir la valeur de retour dans un champ de texte statique.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Exemple TlgGetNumberOfColumns

```
{
char text[5];
long int columns

//get number of Columns
columns = GetNumberOfColumns("TableControl_01");

//convert long int to char
sprintf(text,"%d",columns);

//set text on TextField5
SetText(lpszPictureName,"StatischerText5",text);
}
```

Paramètres de la fonction TlgGetNumberOfColumns :

"TableControl_01" est le nom du WinCC Table Contol.

1. Enregistrer la valeur de retour de la fonction TlgGetNumberOfColumns dans le presse-papiers columns.
2. Enregistrer la valeur de retour dans le presse-papiers Chaîne de caractères text
3. Sortir la valeur de retour dans un champ de texte statique.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Exemple TlgGetNumberOfRows

```
{
char text[5];
long int rows;

//get number of rows
rows = TlgGetNumberOfRows("TableControl_01");

//convert long int to char
sprintf(text,"%d",rows);
```

```
//set text on TextField5
SetText(lpszPictureName,"StatischerText5",text);
}
```

Paramètres de la fonction TlgGetNumberOfRows :

"TableControl_01" est le nom du WinCC Table Contol.

1. Enregistrer la valeur de retour de la fonction TlgGetNumberOfRows dans le presse-papiers rows.
2. Enregistrer la valeur de retour dans le presse-papiers Chaîne de caractères text
3. Sortir la valeur de retour dans un champ de texte statique.

Exemple TlgGetRulerTimeTrend

```
{
SYSTEMTIME systime;
WORD wHour;
WORD wMin;
WORD wSec;

char szTime[10];

//Get the current systemtime
systime = TlgGetRulerTimeTrend("TrendControl_01",0);

//Get the hour
wHour = systime.wHour;
//Get the minute
wMin = systime.wMinute;
//Get the second
wSec = systime.wSecond;

//
sprintf(szTime,"%d:%d:%d",wHour,wMin,wSec);

//output the variable name
SetText(lpszPictureName,"StatischerText7",szTime);
}
```

1. Lire l'heure de système actuelle.
2. Lire l'heure, la minute et la seconde de la structure SYSTEMTIME.
3. Créer une chaîne de caractères qui contient l'heure.
4. Sortir l'heure actuelle.

Exemple TlgGetRulerVariableNameTrend

```
{
char* pszVarName = NULL;
char szVarName[20];

//Get the ruler variable name
pszVarName = TlgGetRulerVariableNameTrend("TrendControl_01",0);

if (pszVarName != NULL)
{
//Copy the string
strncpy(szVarName,pszVarName,19);
}
//output the variable name
SetText(lpszPictureName,"StatischerText6",szVarName);
}
```

Paramètres de la fonction TlgGetRulerVariableNameTrend :

"TrendControl_01" est le nom du WinCC Trend Control.

"0" est le numéro de la courbe.

1. Enregistrer la valeur de retour de la fonction TlgGetRulerVariableNameTrend dans le presse-papiers pszVarName.
2. Si une valeur a été renvoyée, copier la valeur de retour dans szVarName.
3. Sortir la valeur de retour dans un champ de texte statique.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Exemple TlgTrendWindowPressOpenDlgButton

```
{
//Opens the Property Dialog
TlgTrendWindowPressOpenDlgButton("TrendControl_01");
}
```

Paramètres de la fonction TlgTrendWindowPressOpenDlgButton :

"TrendControl_01" est le titre de fenêtre du WinCC Trend Controls.

Exemple TlgTrendWindowPressStartStopButton

```
{
//start/stop the actualisation
TlgTrendWindowPressStartStopButton("TrendControl_01");
}
```

Paramètres de la fonction TlgTrendWindowPressStartStopButton :

"TrendControl_01" est le titre de fenêtre du WinCC Trend Controls.

Remarque

De divers exemples sont proposés pour les descriptions des fonctions. Pour les fonctions ayant une syntaxe similaire, on utilise comme modèle dans l'exemple une fonction sélectionnée. Il vous faut éventuellement adapter cet exemple.

Exemple TlgTrendWindowPressZoomInButton

```
{
//zoom in
TlgTrendWindowPressZoomInButton("TrendControl_01");
}
```

Paramètres de la fonction TlgTrendWindowPressZoomInButton :

"TrendControl_01" est le titre de fenêtre du WinCC Trend Controls.

Exemple TlgTrendWindowPressZoomOutButton

```
{
// zoom out
TlgTrendWindowPressZoomOutButton("TrendControl_01");
}
```

Paramètres de la fonction TlgTrendWindowPressZoomOutButton :

"TrendControl_01" est le titre de fenêtre du WinCC Trend Controls.

2.15.4.8 Exemples de contrôles WinCC

Comment ajouter des éléments à un WinCC OnlineTableControl

Introduction

L'exemple suivant vous permet d'ajouter des colonnes de variables avec des propriétés dans un WinCC OnlineTableControl vide et de relier les colonnes aux variables d'archive.

Conditions requises

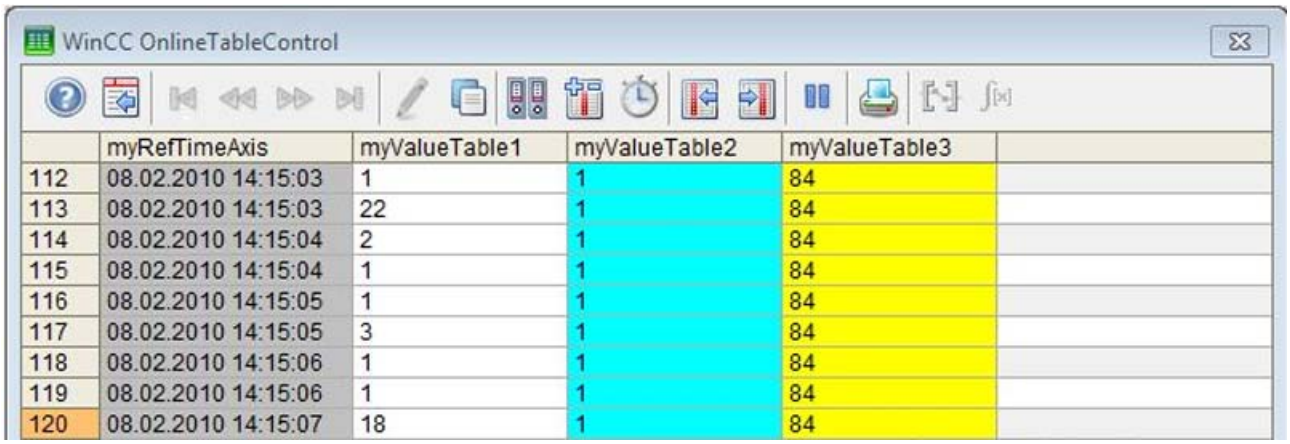
- Une archive est créée dans l'éditeur Tag Logging avec trois variables d'archive.
- Un "WinCC OnlineTableControl" est ajouté dans le synoptique au niveau du Graphics Designer avec le nom "Control2".
- Un bouton est ajouté dans le Graphics Designer. Vous avez par ex. configuré l'événement "Clic de souris" avec une action en C et le script suivant.

Exemple

```
//enable BackColor
SetPropBOOL(lpszPictureName, "Control2", "UseColumnBackColor",TRUE);

//add new new TimeColumn and assign column length
SetPropChar(lpszPictureName, "Control2", "TimeColumnAdd", "myRefTimeColumn");
SetPropWord(lpszPictureName, "Control2", "TimeColumnLength", 20);

//add new ValueColumn and assign propertys
SetPropChar(lpszPictureName, "Control2", "ValueColumnAdd", "myValueTable1");
SetPropWord(lpszPictureName, "Control2", "ValueColumnProvider", 1);
SetPropChar(lpszPictureName, "Control2", "ValueColumnTagName", "Process value archive\
\PDL_ZT_1");
SetPropWord(lpszPictureName, "Control2", "ValueColumnBackColor", RGB(255,255,255));
SetPropChar(lpszPictureName, "Control2", "ValueColumnTimeColumn", "myRefTimeColumn");

//add new ValueColumn and assign propertys
SetPropChar(lpszPictureName, "Control2", "ValueColumnAdd", "myValueTable2");
SetPropWord(lpszPictureName, "Control2", "ValueColumnProvider", 1);
SetPropChar(lpszPictureName, "Control2", "ValueColumnTagName", "Process value archive\
\PDL_ZT_2");
SetPropWord(lpszPictureName, "Control2", "ValueColumnBackColor", RGB(0,255,255));
SetPropChar(lpszPictureName, "Control2", "ValueColumnTimeColumn", "myRefTimeColumn");

//add new ValueColumn and assign propertys
SetPropChar(lpszPictureName, "Control2", "ValueColumnAdd", "myValueTable3");
```

```
SetPropWord(lpszPictureName, "Control2", "ValueColumnProvider", 1);
SetPropChar(lpszPictureName, "Control2", "ValueColumnTagName", "Process value archive\
\PDL_ZT_3");
SetPropWord(lpszPictureName, "Control2", "ValueColumnBackColor", RGB(255,255,0));
SetPropChar(lpszPictureName, "Control2", "ValueColumnTimeColumn", "myRefTimeColumn");
```

Résultat



	myRefTimeAxis	myValueTable1	myValueTable2	myValueTable3
112	08.02.2010 14:15:03	1	1	84
113	08.02.2010 14:15:03	22	1	84
114	08.02.2010 14:15:04	2	1	84
115	08.02.2010 14:15:04	1	1	84
116	08.02.2010 14:15:05	1	1	84
117	08.02.2010 14:15:05	3	1	84
118	08.02.2010 14:15:06	1	1	84
119	08.02.2010 14:15:06	1	1	84
120	08.02.2010 14:15:07	18	1	84

Comment ajouter des éléments à un WinCC OnlineTrendControl

Introduction

Dans l'exemple suivant, sous insérez les éléments Fenêtre de courbe, Axe des valeurs, Axe des temps et Courbes dans un WinCC OnlineTrendControl vide.

Conditions requises

- Une archive est créée dans l'éditeur Tag Logging avec trois variables d'archive.
- Un "WinCC OnlineTrendControl" est ajouté dans le synoptique au niveau du Graphics Designer avec le nom "Control2".
- Un bouton est ajouté dans le Graphics Designer. Vous avez par ex. configuré l'événement "Clic de souris" avec une action en C et le script suivant.

Exemple

```
//create reference to new window, time and value axis
SetPropChar(lpszPictureName, "Control2", "TrendWindowAdd", "myWindow");
SetPropChar(lpszPictureName, "Control2", "TimeAxisAdd", "myTimeAxis");
SetPropChar(lpszPictureName, "Control2", "ValueAxisAdd", "myValueAxis");
```

```
//assign time and value axis to the window
SetPropChar(lpszPictureName, "Control2", "TimeAxisTrendWindow", "myWindow");
SetPropChar(lpszPictureName, "Control2", "ValueAxisTrendWindow", "myWindow");

//add new trend and assign propertys
SetPropChar(lpszPictureName, "Control2", "TrendAdd", "myTrend1");
SetPropWord(lpszPictureName, "Control2", "TrendProvider", 1);
SetPropChar(lpszPictureName, "Control2", "TrendTagName", "Process value archive\
\PDL_ZT_1");
SetPropWord(lpszPictureName, "Control2", "TrendColor",  RGB(255,0,0));
SetPropChar(lpszPictureName, "Control2", "TrendTrendWindow", "myWindow");
SetPropChar(lpszPictureName, "Control2", "TrendTimeAxis", "myTimeAxis");
SetPropChar(lpszPictureName, "Control2", "TrendValueAxis", "myValueAxis");

//add new trend and assign propertys
SetPropChar(lpszPictureName, "Control2", "TrendAdd", "myTrend2");
SetPropWord(lpszPictureName, "Control2", "TrendProvider", 1);
SetPropChar(lpszPictureName, "Control2", "TrendTagName", "Process value archive\
\PDL_ZT_2");
SetPropWord(lpszPictureName, "Control2", "TrendColor", RGB(0,255,0));
SetPropChar(lpszPictureName, "Control2", "TrendTrendWindow", "myWindow");
SetPropChar(lpszPictureName, "Control2", "TrendTimeAxis", "myTimeAxis");
SetPropChar(lpszPictureName, "Control2", "TrendValueAxis", "myValueAxis");

//add new trend and assign propertys
SetPropChar(lpszPictureName, "Control2", "TrendAdd", "myTrend3");
SetPropWord(lpszPictureName, "Control2", "TrendProvider", 1);
SetPropChar(lpszPictureName, "Control2", "TrendTagName", "Process value archive\
\PDL_ZT_3");
SetPropWord(lpszPictureName, "Control2", "TrendColor", RGB(0,0,255));
SetPropChar(lpszPictureName, "Control2", "TrendTrendWindow", "myWindow");
SetPropChar(lpszPictureName, "Control2", "TrendTimeAxis", "myTimeAxis");
SetPropChar(lpszPictureName, "Control2", "TrendValueAxis", "myValueAxis");
```

2.15.5 Listes

2.15.5.1 Orientation du bargraphe

Orientation du bargraphe	Valeur chiffrée
en haut	0
en bas	1
à gauche	2
vers la droite	3

2.15.5.2 Mise à l'échelle du bargraphe

Valeur chiffrée	Mise à l'échelle du bargraphe
0	Linéaire (même priorité)
1	Logarithmique (valeurs basses soulignées)
2	Logarithmique négatif (valeurs élevées soulignées)
3	Automatique (linéaire)
4	Tangente (valeurs basses et élevées soulignées)
5	Carré (valeurs élevées soulignées)
6	Cubique (valeurs élevées fortement soulignées)

2.15.5.3 Fréquences de clignotement

Valeur chiffrée	Fréquence de clignotement
0	0,5 Hz
1	2 Hz
2	8 Hz

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence de clignotement dépend du système et du matériel (nombre d'objets, processeur, mémoire, temps de rafraîchissement, etc.).

Pour cette raison, les données du tableau ne doivent être considérées que comme valeurs d'orientation.

2.15.5.4 Champ d'E/S, Format de sortie

Une indication de format détermine comment des valeurs chiffrées sont représentées lors de la sortie dans un champ E/S.

Une indication de format se compose d'un ou de plusieurs signes de formatage. Les signes de formatage en vigueur et leur signification sont regroupés dans le tableau suivant :

Caractères	Signification	Remarque
s	Les chiffres positifs sont sortis avec un signe.	Se trouve toujours à la première position de l'indication de format Ne doit exister qu'une seule fois dans l'indication de format
0(NULL)	Des zéros en tête et des zéros finaux sont sortis.	Se trouve toujours après s Se trouve toujours à la première position si s manque Ne doit exister qu'une seule fois dans l'indication de format
9	Indique la position d'un chiffre dans le nombre à sortir	Peut exister dans l'indication de format autant de fois que vous voulez.
, (Virgule)	Position du signe décimal	
e	Sort le nombre en représentation exponentielle	Se trouve toujours à la dernière position de l'indication de format

Exemples :

chiffre	Format	Représentation
123,455	999,999	123,455
123,455	999,99	123,46
123,455	9999,9999	123,4550
123,455	s09999,9999	+0123,4550
123,455	9,99999e	1,23455e+002

Si le signe décimal est omis dans l'indication de format, les chiffres après la virgule ne sont pas affichés et le nombre est arrondi à un nombre entier.

Si, dans l'indication de format, moins de chiffres après la virgule que possède le nombre sont prévus, seulement autant de chiffres après la virgule sont sortis que déterminé dans l'indication de format.

Le nombre est donc arrondi.

Si le nombre possède plus de chiffres avant la virgule que déterminé dans l'indication de format, trois astérisques (***) sont sortis pour montrer que le nombre ne peut pas être affiché dans ce format.

2.15.5.5 Champ E/S, Type de données du contenu du champ

Type de données	Valeur chiffrée
binaire	0
déc.	1
string (chaîne)	2
hexadécimal	3

2.15.5.6 Champ E/S, Type de champ

Type	Valeur chiffrée
Édition	0
Entrée	1
Sortie et entrée	2

2.15.5.7 Alignement d'élément dans des cases à cocher et des cases d'option

Alignement	Valeur chiffrée
à gauche	0
vers la droite	-1

2.15.5.8 Tableau de couleurs

Les 16 couleurs de base sont :

Couleur	Valeur de couleur(Hex)	Constante symbolique
Rouge	0x000000FF	CO_RED
Rouge foncé	0x00000080	CO_DKRED
Signal vidéo vert	0x0000FF00	CO_GREEN
Vert foncé	0x00008000	CO_DKGREEN

Couleur	Valeur de couleur(Hex)	Constante symbolique
Bleu	0x00FF0000	CO_BLUE
Bleu foncé	0x00800000	CO_DKBLUE
Cyan	0x00FFFF00	CO_CYAN
Cyan foncé	0x00808000	CO_DKCYAN
Jaune	0x0000FFFF	CO_YELLOW
Jaune foncé	0x00008080	CO_DKYELLOW
Magenta	0x00FF00FF	CO_MAGENTA
Magenta foncé	0x00800080	CO_DKMAGENTA
Gris clair	0x00C0C0C0	CO_LTGRAY
Gris	0x00808080	CO_DKGRAY
Noir	0x00000000	CO_BLACK
Blanc	0x00FFFFFF	CO_WHITE

Remarque

Les constantes symboliques sont prédéfinies de façon externe par #define et sont mises à disposition par WinCC.

2.15.5.9 Descripteur de format

Le type suivant est attendu des descripteurs de format :

%d = DWORD / Int

%f = double

%s = char*

De plus, il est possible de lire, par ex., une variable de texte avec %d s'il est sûr que la valeur peut être représentée dans un DWORD.

Dans le cas de l'affectation suivante, il est sûr que la valeur peut être représentée :

variable	Format	Variable C
bit	%d	DWORD / long int signed
octet	%d	DWORD / long int signed
SByte	%d	DWORD / long int signed
Mot	%d	DWORD / long int signed
SWord	%d	DWORD / long int signed

variable	Format	Variable C
DWord	%d	DWORD / long int signed
SDWord	%d	DWORD / long int signed
Float	%f	double
Double	%f	double
Char	%s	char*

Remarque

Pour lire un "DWORD" dont le 32è bit est mis, il faut utiliser un descripteur de format pour les nombres entiers sans signe (%u).

2.15.5.10 Motif de remplissage

Motif de remplissage	Valeur
Transparent	65536
Massif	0

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
	1048576		196611		196627
	1048577		196612		196628
	1048578		196613		196629
	1048579		196614		196630
	1048832		196615		196631
	1048833		196616		196632
	1048834		196617		196633
	1048835		196618		196634

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
	131072		196619		196635
	131073		196620		196636
	131074		196621		196637
	131075		196622		196638
	131076		196623		196639
	196608		196624		196640
	196609		196625		196641
	196610		196626		196642

Remarque

Le motif de remplissage "Massif" remplit l'objet avec la couleur d'arrière-plan configurée.

2.15.5.11 Types de trait

Style de trait	Mnémoniques	Valeur
	LS_SOLID	0
	LS_DASH	1
	LS_DOT	2
	LS_DASHDOT	3
	LS_DASHDOT DOT	4
caché	LS_INVISIBLE	5

2.15.5.12 Fins de trait

Fin de trait	Mnémonique	Valeur pour les fins de trait gauches	Valeur pour les fins de trait droites
	LE_NO	0	0
	LE_HOLLOW_ARROW	1	65536
	LE_FULL_ARROW	2	131072
	LE_CFULL_ARROW	3	196608
	LE_LINE	4	262144
	LE_HOLLOW_CIRCLE	5	327680
	LE_FULL_CIRCLE	6	393216

Remarque

A partir d'une largeur de ligne > 5, la fin de trait "Cercle vide" est représentée sous forme d'un cercle plein.

2.15.5.13 Types de liste

Type de liste	Valeur chiffrée
déc.	0
binaire	1
bit	2

2.15.5.14 Identificateurs de langue

WinCC ne supporte que les langues SUBLANG_DEFAULT de Windows.

Mnémonique	Valeur(hexadécimale)	Abréviation
LANG_ARABIC	0x0401	
LANG_AFRIKAANS	0x0436	
LANG_ALBANIAN	0x041C	
LANG_BASQUE	0x042D	
LANG_BULGARIAN	0x0402	

Mnémonique	Valeur(hexadécimale)	Abréviation
LANG_BYELORUSSIAN	0x0423	
LANG_CATALAN	0x0403	
LANG_CHINESE	0x0404	
LANG_CROATIAN	0x041A	
LANG_CZECH	0x0405	CSY
LANG_DANISH	0x0406	DAN
LANG_DUTCH	0x0413	NLD
LANG_ENGLISH	0x0409	ENU
LANG_ESTONIAN	0x0425	
LANG_FAEROESE	0x0438	
LANG_FARSI	0x0429	
LANG_FINNISH	0x040B	FIN
LANG_FRENCH	0x040C	FRA
LANG_GERMAN	0x0407	DEU
LANG_GREEK	0x0408	
LANG_HEBREW	0x040D	
LANG_HUNGARIAN	0x040E	HUN
LANG_ICELANDIC	0x040F	ISL
LANG_INDONESIAN	0x0421	
LANG_ITALIAN	0x0410	ITA
LANG_JAPANESE	0x0411	
LANG_KOREAN	0x0412	
LANG_LATVIAN	0x0426	
LANG_LITHUANIAN	0x0427	
LANG_NORWEGIAN	0x0414	NOR
LANG_POLISH	0x0415	PLK
LANG_PORTUGUESE	0x0416	PTB
LANG_ROMANIAN	0x0418	
LANG_RUSSIAN	0x0419	RUS
LANG_SLOVAK	0x041B	SKY
LANG_SLOVENIAN	0x0424	
LANG_SORBIAN	0x042E	
LANG_SPANISH	0x040A	ESP
LANG_SWEDISH	0x041D	EAS
LANG_THAI	0x041E	
LANG_TURKISH	0x041F	TRK
LANG_UKRAINIAN	0x0422	

2.15.5.15 Sens du texte

Horizontal	Alignement	Valeur chiffrée
	à gauche	0
	centré	1
	vers la droite	2

A la verticale	Alignement	Valeur chiffrée
	en haut	0
	centré	1
	en bas	2

2.15.5.16 Etats de variable

Valeur (décimale)	Valeur (hexadécimale)	Signification
0	0x0000	Pas d'erreur
1	0x0001	Pas d'établissement de liaison au partenaire
2	0x0002	Erreur de protocole
4	0x0004	Module réseau défectueux
8	0x0008	Limite supérieure configurée dépassée
16	0x0010	Limite inférieure configurée ne pas atteinte
32	0x0020	Dépassement de limite supérieure de format
64	0x0040	Dépassement de limite inférieure de format
128	0x0080	Erreur de transtypage
256	0x0100	Valeur d'initialisation de la variable
512	0x0200	Valeur de substitution de la variable
1024	0x0400	Erreur d'adressage dans le canal
2048	0x0800	Variable ne pas trouvée ou ne pas existante
4096	0x1000	Accès à la variable non autorisé !
8192	0x2000	Timeout, pas de réponse du canal
16384	0x4000	Serveur non disponible

2.15.6 Définitions de structure

2.15.6.1 Définition de structure CCAPErrorExecute

```
typedef struct {
DWORD dwCurrentThreadID; Thread ID of the current thread
DWORD dwErrorCode1;      Error code 1
DWORD dwErrorCode2;      Error code 2
BOOL bCycle;             cycle/acycle
char* szApplicationName; Name of the application
char* szFunctionName;    Name of the function
char* szTagName;         Name of the tag
LPVOID lpParam;          Pointer to the action stack
DWORD dwParamSize;       Size of the action stack
DWORD dwCycle;           Cycle of the variable
CMN_ERROR* pError;       Pointer to CMN_ERROR
} CCAPErrorExecute;
```

Members

La signification des différents identificateurs et les éléments de structure transmis en dépendance de ceci sont indiqués dans le tableau suivant :

dwErrorCode1	dwErrorCode2	bCycle	szApplicationName	szFunctionName	szTagName	lpParamSize	dwParamSize	dwCycle	pError	Description
1007001	0	X	x	x		x	x			Exception dans l'action
1007001	1	x	x	x		x	x			Exception lors de l'accès au résultat de renvoi
1007001	4097	x	x	x		x	x			Débordement de pile lors de l'exécution de l'action
1007001	4098	x	x	x		x	x			Division par 0 dans l'action
1007001	4099	x	x	x		x	x			Accès à un symbole non disponible dans l'action
1007001	4100	x	x	x		x	x			Access violation dans l'action
1007004	0	x	x	x						Fonction inconnue
1007005	1	x	x							Action ne contient aucun code P
1007005	2	x	x							Nom de fonction erroné

1007005	4	x	x	x		x	x			Type de la valeur de renvoi invalide
1007005	32768ff	X	X	X		x	x			Erreur dans Ciss Compiler lors du chargement de l'action
1007006	0	x	x	x	x	x	x	x		Variable n'est pas définie
1007006	1	x	x	x	x	x	x	x		Variable timeout
1007006	2	X	X	X	X	X	X	X	X	Variable ne peut pas être fournie dans le format souhaité
1007006	3	X	X	X	X	X	X	X	X	Variable fournit violation d'état, l'état est indiqué dans CMN_ERROR.dwError1
1007007	1	x	x	x		x	x		X	Erreur dans PDLRTGetProp
1007007	2	x	x	x		x	x		X	Erreur dans PDLRTSetProp
1007007	3	x	x	x		x	x		X	Erreur lors d'appel DM

Structure d'erreur

La fonction OnErrorExecute utilise la structure d'erreur pour évaluer ou sortir des messages d'erreur si ceci est marqué avec un "x" dans la colonne pError.

Voir aussi

Définition de structure CMN_ERROR (Page 1594)

2.15.6.2 Définition de structure CCAPTime

```
typedef struct {
DWORD dwCurrentThreadID; ThreadID of the current Thread
DWORD dwCode;            Code
BOOL bCycle;             cycle/acycle
char* szApplicationName; Name of the Application
char* szFunctionName;    Name of the Function
LPVOID lpParam;          Pointer to the Action-Stack
DWORD dwParamSize;       size of the Action-Stack
double dblTime;
DWORD dwFlags;           flags
} CCAPTime;
```

Members

dwCode

L'élément de structure dwCode donne des informations sur l'appel de OnTime :

dwCode = 113	Appel avec détermination de temps pour chaque action
dwCode = 114	Appel avec détermination de temps pour une action

dwFlags

L'élément de structure dwFlags donne des informations sur le type de la sortie :

dwFlags = TRUE	Les résultats sont sortis dans un tableau
dwFlags = FALSE	Les résultats sont sortis dans la fenêtre de diagnostic

2.15.6.3 Définition de structure CMN_ERROR

```
struct CMNERRORSTRUCT {
DWORD       dwError1,
DWORD       dwError2,
DWORD       dwError3,
DWORD       dwError4,
DWORD       dwError5;
TCHAR       szErrorText[MAX_ERROR_LEN];
}
CMN_ERROR
```

Description

Le code d'erreur et un texte d'erreur pour les erreurs survenues se trouvent dans la structure d'erreur étendu. Chaque application peut utiliser la structure d'erreur pour évaluer ou sortir des messages d'erreur.

Members

dwError1 .. dwError5

Ces entrées peuvent être utilisées par les fonctions API comme vous le souhaitez.

Dans les descriptions API est décrit quelle entrée contient quelle valeur dans le cas d'une erreur. Si rien d'autre n'est indiqué, les codes d'erreur se trouvent dans dwError1.

szErrorText

Tampon pour la description en texte de la cause d'erreur

Le contenu est déterminé à partir des ressources et dépend donc de la langue.

2.15.6.4 Définition de structure DM_TYPEREF

```
typedef struct {
DWORD dwType;
DWORD dwSize;
char szTypeName[MAX_DM_TYPE_NAME + 1];
}
DM_TYPEREF;
```

Members

dwType

Spécifie le type de la variable

DM_VARTYPE_BIT	Variable binaire
DM_VARTYPE_SBYTE	Valeur 8 bits signée
DM_VARTYPE_BYTE	Valeur 8 bits non signée
DM_VARTYPE_SWORD	Valeur 16 bits signée
DM_VARTYPE_WORD	Valeur 16 bits non signée
DM_VARTYPE_SDWORD	Valeur 32 bits signée
DM_VARTYPE_DWORD	Valeur 32 bits non signée
DM_VARTYPE_FLOAT	Nombre à virgule flottante 32 bits IEEE 754
DM_VARTYPE_DOUBLE	Nombre à virgule flottante 64 bits IEEE 754
DM_VARTYPE_TEXT_8	Variable texte 8 bits jeu de caractères
DM_VARTYPE_TEXT_16	Variable texte 16 bits jeu de caractères
DM_VARTYPE_RAW	Type de données brutes
DM_VARTYPE_STRUCT	Variable de type structure
DM_VARTYPE_TEXTREF	Variable de référence de texte

dwSize

Indique la longueur du type de données en octets.

szTypeName

Contient, dans le cas de variables de structure, le nom du type de structure

2.15.6.5 Définition de structure DM_VAR_UPDATE_STRUCT

```
typedef struct {
DM_TYPEREF dmTypeRef;
DM_VARKEY dmVarKey;
VARIANT dmValue;
DWORD dwState;
}
DM_VAR_UPDATE_STRUCT;
```

Members

dmTypeRef

Contient des indications concernant le type de variable. En cas de demandes cycliques, rien n'est entré dans cette structure pour des raisons de performance.

dmVarKey

Spécifie les variables à traiter.

dmValue

Valeur de la variable.

En cas d'accès à la valeur de VARIANT, il faut entrer ".u." entre le nom du VARIANT et le nom du Member.

```
Exemple
// Variant versorgen
myVariant.vt = VT_I4;
myVariant.u.lVal = 233;
```

Pour la description du type de données VARIANT, veuillez vous référer aux ouvrages spécialisés. Le VARIANT dmValue doit être initialisé avant sa première utilisation par VariantInit() et être validé après utilisation par VariantClear(&dmValue). La structure DM_VAR_UPDATE_STRUCT ne doit pour cette raison pas être effacée avec ZeroMemory() ou memset().

dwState

Identifie l'état de la variable.

Voir aussi

Etats de variable (Page 1591)

Définition de structure DM_VARKEY (Page 1598)

Définition de structure DM_TYPEREF (Page 1595)

2.15.6.6 Définition de structure DM_VAR_UPDATE_STRUCTEX

```
typedef struct {
DM_TYPEREF dmTypeRef;
DM_VARKEY dmVarKey;
VARIANT dmValue;
DWORD dwState;
DWORD dwQualityCode;
}
DM_VAR_UPDATE_STRUCTEX;
```

Members

dmTypeRef

Contient des indications concernant le type de variable. En cas de demandes cycliques, rien n'est entré dans cette structure pour des raisons de performance.

dmVarKey

Spécifie les variables à traiter.

dmValue

Valeur de la variable.

En cas d'accès à la valeur de VARIANT, il faut entrer ".u." entre le nom du VARIANT et le nom du Member.

```
Exemple
// Variant versorgen
myVariant.vt = VT_I4;
myVariant.u.lVal = 233;
```

Pour la description du type de données VARIANT, veuillez vous référer aux ouvrages spécialisés. Le VARIANT dmValue doit être initialisé avant sa première utilisation par VariantInit() et être validé après utilisation par VariantClear(&dmValue). La structure DM_VAR_UPDATE_STRUCTEX ne doit pour cette raison pas être effacée avec ZeroMemory() ou memset().

dwState

Identifie l'état de la variable.

dwQualityCode

Identifie le QualityCode de la variable.

Voir aussi

Etats de variable (Page 1591)
Définition de structure DM_VARKEY (Page 1598)
Définition de structure DM_TYPEREF (Page 1595)

2.15.6.7 Définition de structure DM_VARKEY

```
typedef struct {
DWORD dwKeyType;
DWORD dwID;
char szName[ MAX_DM_VAR_NAME + 1 ];
LPVOID lpvUserData;
}
DM_VARKEY;
```

Members

dwKeyType

Indique si l'accès à la variable doit s'effectuer via un ID clé ou via son nom :

DM_VARKEY_ID Spezifizierung über Schlüssel ID

DM_VARKEY_NAME Spezifizierung über Variablennamen

dwID

Contient l'ID clé de la variable si dwKeyType est mis comme nécessaire

szName

Contient le nom de la variable si dwKeyType est mis comme nécessaire.

lpvUserData

Pointeur sur données spécifiques à l'application

2.15.6.8 Définition de structure LINKINFO

```
typedef struct {
LINKTYPE LinkType;
DWORD dwCycle;
TCHAR szLinkName[256];
}
LINKINFO;
```

Members

LinkType

LinkType sont des constantes d'énumération, définies dans le fichier "Trigger.h". Vous pouvez les intégrer dans votre script avec l'instruction #include "Trigger.h" et la constante d'énumération correspondante.

BUBRT_LT_NOLINK	0	Pas de lien
BUBRT_LT_VARIABLE_DIRECT	1	Variable directe
BUBRT_LT_VARIABLE_INDIRECT	2	Variable indirecte
BUBRT_LT_ACTION	3	Action en C
BUBRT_LT_ACTION_WIZARD	4	Dynamic Dialog
BUB_LT_DIRECT_CONNECTION	5	Liaison directe
BUBRT_LT_ACTION_WIZARD_INPROC	6	Dynamic Dialog

Pour la fonction SetLink vous ne pouvez utiliser que les constantes d'énumération BUBRT_LT_VARIABLE_DIREKT et BUBRT_LT_VARIABLE_INDIRECT. Pour la fonction GetLink toutes les constantes d'énumération mentionnées peuvent être renvoyées.

dwCycle

Temps de cycle de rafraîchissement

dwCycle	Cycle d'actualisation
255	Cycle vue
235	Cycle fenêtre
0	En cas de modification
1	250ms
2	500ms
3	1s
4	2s
5	5s
6	10s
7	1min
8	5min
9	10min
10	1h
11-15	Cycle utilisateur 1-5

szLinkName

Nom de variable

2.15.6.9 Définition de structure MSG_FILTER_STRUCT

```
typedef struct {
CHAR        szFilterName[MSG_MAX_TEXTLEN+1];
WORD        dwFilter;
SYSTEMTIME  st[2];
DWORD       dwMsgNr[2];
DWORD       dwMsgClass;
DWORD       dwMsgType[MSG_MAX_CLASS];
DWORD       dwMsgState;
WORD        wAGNr[2];
WORD        wAGSubNr[2];
DWORD       dwArchivMode;
char        szTB[MSG_MAX_TB][
MSG_MAX_TB_CONTENT+1]
DWORD       dwTB;
Double      dPValue[MSG_MAX_PVALUE][2];
DWORD       dwPValue[2];
DWORD       dwMsgCounter[2];
DWORD       dwQuickSelect;
}
MSG_FILTER_STRUCT;
```

Description

Dans cette structure sont indiquées les critères de filtrage.

Members

dwFilter

Les conditions de filtrage sont définies par les constantes suivantes du fichier "m_global.h" :

MSG_FILTER_DATE_FROM	Date de
MSG_FILTER_DATE_TO	Date à
MSG_FILTER_TIME_FROM	Heure de
MSG_FILTER_TIME_TO	Heure à
MSG_FILTER_NR_FROM	Numéro d'alarme de
MSG_FILTER_NR_TO	Numéro d'alarme à
MSG_FILTER_CLASS	Classes d'alarme
MSG_FILTER_STATE	Etat d'alarme
MSG_FILTER_AG_FROM	Numéro d'automate de
MSG_FILTER_AG_TO	Numéro d'automate à
MSG_FILTER_AGSUB_FROM	Sous-numéro API de
MSG_FILTER_AGSUB_TO	Sous-numéro API à
MSG_FILTER_TEXT	Textes d'alarme
MSG_FILTER_PVALUE	valeurs de process

MSG_FILTER_COUNTER_FROM	Compteur d'alarmes interne de
MSG_FILTER_COUNTER_TO	Compteur d'alarmes interne à
MSG_FILTER_PROCESSMSG	Alarmes de process
MSG_FILTER_SYSMSG	Alarmes système
MSG_FILTER_BEDMSG	Messages de conduite
MSG_FILTER_DATE	Date de à
MSG_FILTER_TIME	Heure de à
MSG_FILTER_NR	Numéro d'alarme de à
MSG_FILTER_VISIBLEONLY	Afficher les alarmes visibles
MSG_FILTER_HIDDENONLY	Afficher les alarmes masquées

st

Date / Heure de - à

st[0] étant moment initial (de), st[1] moment final (à)

Assignez ces champs par les critères de filtrage : MSG_FILTER_DATE, MSG_FILTER_DATE_FROM, MSG_FILTER_DATE_TO, MSG_FILTER_TIME, MSG_FILTER_TIME_FROM, ou MSG_FILTER_TIME_TO

Si un temps actuel est nécessaire pour la transmission d'un paramètre SYSTEMTIME, utilisez la fonction GetLocalTime et ne pas GetSystemTime. En général, il existe une différence de temps importante entre ces deux fonctions.

dwMsgNr

Numéro d'alarme de - à

dwMsgNr[0] étant numéro initial (de), dwMsgNr[1] numéro final (à)

Assignez ces champs par les critères de filtrage : MSG_FILTER_NR, MSG_FILTER_NR_FROM bzw. MSG_FILTER_NR_TO

dwMsgClass

Classes d'alarme codées par bit

Assignez ce champ par le critère de filtrage : MSG_FILTER_CLASS

dwMsgType

Type d'alarme par classe d'alarme, codé par bit

Assignez ce champ par le critère de filtrage : MSG_FILTER_CLASS

dwMsgState

Etat d'alarme codé par bit

Assignez ce champ par le critère de filtrage : MSG_FILTER_STATE

wAGNr

Numéro API de - à

Assignez ces champs par les critères de filtrage : MSG_FILTER_AG_FROM bzw. MSG_FILTER_AG_TO

wAGSubNr

Sous-numéro API de - à

Assignez ce champ par les critères de filtrage : MSG_FILTER_AGSUB_FROM bzw. MSG_FILTER_AGSUB_TO

dwArchivMode

Archivage / Journalisation

Doit être assigné par 0.

szTB

Textes des blocs d'alarme

Assignez ces champs par le critère de filtrage : MSG_FILTER_TEXT

dwTB

Blocs de texte actifs (de - á, codés par bit)

Assignez ce champ par le critère de filtrage : MSG_FILTER_TEXT

dPValue

Valeurs de process de - à

Assignez ces champs par le critère de filtrage : MSG_FILTER_PVALUE

dwPValue

Valeurs de process actives (de - à, codées par bit)

Assignez ce champ par le critère de filtrage : MSG_FILTER_PVALUE

dwMsgCounter

Compteur d'alarmes interne de - à

Assignez ces champs par les critères de filtrage : MSG_FILTER_COUNTER_FROM, MSG_FILTER_COUNTER_TO

dwQuickSelect

Sélection rapide pour Heure, Jour, Mois

Le paramètre est réservé pour des extensions ultérieures et doit être "0" par défaut.

Assignez ce champ par le critère de filtrage : MSG_FILTER_QUICKSELECT

Type LOWORD :

MSG_FILTER_QUICK_MONTH	Sélection rapide n derniers mois
MSG_FILTER_QUICK_DAYS	Sélection rapide n derniers jours
MSG_FILTER_QUICK_HOUR	Sélection rapide n dernières heures

HIWORD Nombre : 1...n

Le moment final de la sélection rapide se réfère à l'haure système actelle de l'ordinateur local. Moment de départ devient n * (Mois, Jours, Heures ramenés).

2.15.6.10 Définition de structure MSG_RTDATA_STRUCT

```
typedef struct {
DWORD       dwMsgState;
DWORD                dwMsgNr;
SYSTEMTIME           stMsgTime;
DWORD                dwTimeDiff;
DWORD                dwCounter;
DWORD                dwFlags;
WORD                 wPValueUsed;
WORD                 wTextValueUsed;
double               dPValue[MSG_MAX_PVALUE];
MSG_TEXTVAL_STRUCT   mtTextValue[MSG_MAX_PVALUE];
}
MSG_RTDATA_STRUCT;
```

Members

dwMsgState

Etat d'une alarme

MSG_FILTER_STATE	0x00000001	Alarme arrivée
MSG_STATE_GO	0x00000002	Alarme sortie
MSG_STATE_QUIT	0x00000003	Alarme acquittée
MSG_STATE_LOCK	0x00000004	Alarme verrouillée
MSG_STATE_UNLOCK	0x00000005	Déverrouillage d'alarme
MSG_STATE_QUIT_SYSTEM	0x00000010	Alarme acquittée par système
MSG_STATE_QUIT_EMERGENCY	0x00000011	Acquittement de secours
MSG_STATE_QUIT_HORN	0x00000012	Acquittement d´avertisseur sonore
MSG_STATE_COMEGO	0x00000013	Alarme arrivée et partie, affichage uniquement dans la liste d'alarmes
MSG_STATE_UPDATE	0x00010000	Bit pour rafraîchissement d'alarme
MSG_STATE_RESET	0x00020000	Bit pour réinitialisation d'alarme
MSG_STATE_SUMTIME	0x00040000	Bit pour heure d'été actif
MSG_STATE_INSTANCE	0x00080000	Bit pour alarme d'instance (n alarmes d'un n°)

dwMsgNr

Numéro d´alarme

stMsgTime

Date / Heure : Heure télégramme selon la fonction déclenchant l'appel

dwTimeDiff

Durée arrivant / Heure télégramme en secondes

dwCounter

Compteur d'alarmes internes

dwFlags

Pointeurs d'alarmes dans la base de données

MSG_FLAG_SUMTIME	0x00000001	Heure d'été activée
MSG_FLAG_COMMENT	0x00000002	Alarme possède un commentaire
MSG_FLAG_ARCHIV	0x00000004	Archiver
MSG_FLAG_PROTOCOL	0x00000008	Journal
MSG_FLAG_TEXTVALUES	0x00000010	Alarme possède des valeurs d'accompagnement de texte
MSG_FLAG_TIMEINVALID	0x00000020	Bit pour horodatage non valide
MSG_FLAG_INSTANCE	0x00000040	Identification d'alarmes d'instance (185269)

wPValueUsed

Valeurs de process utilisées, codées en bit Chaque bit ne doit être mis que dans un des deux éléments de structure "wPValueUsed" ou "wTextValueUsed". Une valeur d'accompagnement peut être nombre ou texte.

wTextValueUsed

Valeurs de texte utilisées, codées en bit Chaque bit ne doit être mis que dans un des deux éléments de structure "wPValueUsed" ou "wTextValueUsed". Une valeur d'accompagnement peut être nombre ou texte.

VBA pour la configuration automatisée 3

3.1 Automatisation de la configuration

Contenu

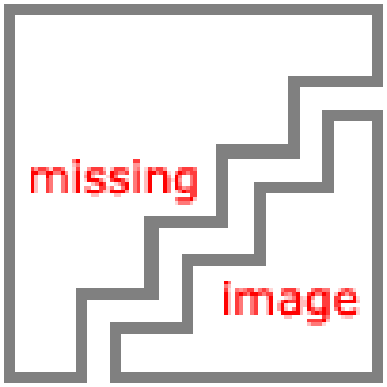
VBA vous permet d'automatiser la configuration dans le Graphics Designer, à savoir :

- Adaptation du Graphics Designer
- Edition d'images
- Edition d'objets
- Dynamisation avec VBA
- Accès aux applications externes

Un Editeur VBA est à votre disposition à cet effet dans l'"Editeur Graphics Designer".

Ce chapitre comprend :

- une courte introduction à l'utilisation de VBA dans WinCC,
- les principes d'utilisation de VBA dans le Graphics Designer et
- la référence au modèle d'objet VBA du Graphics Designer.



3.2 Introduction : Utilisation de VBA dans WinCC

3.2.1 Introduction : Utilisation de VBA dans WinCC

Introduction

Le Graphics Designer met à votre disposition un éditeur VBA à l'aide duquel vous pouvez automatiser la configuration des vues. L'Editeur VBA est identique à celui de la famille de produits Office de Microsoft. Vous pouvez donc mettre directement à profit votre expérience de programmation VBA.

Principe

VBA vous permet d'étendre la fonctionnalité du Graphics Designer et d'automatiser la configuration. Dans le Graphics Designer, vous pouvez utiliser VBA notamment pour :

- créer des menus et des barres d'outils personnalisés,
- créer et éditer des objets standards, Smart et Windows,
- Dynamisation des propriétés d'une vue ou d'un objet
- configurer des actions liées aux vues et aux objets,
- accéder à des produits prenant VBA en charge (p. ex. les produits de la Suite MS Office).

Vous trouverez la description du modèle d'objet VBA pour le Graphics Designer dans la présente documentation, à la rubrique "Référence VBA".

Voir aussi

Organisation du code VBA dans un projet WinCC (Page 1607)

Référence VBA (Page 1724)

VBA dans le Graphics Designer (Page 1614)

Exécution de macros VBA dans le Graphics Designer (Page 1612)

Différenciation : Utilisation de VBA (Page 1606)

3.2.2 Différenciation : Utilisation de VBA

Introduction

VBA est utilisé uniquement pour la configuration et l'extension des fonctions sous Graphics Designer. Les cas dans lesquels, p. ex., d'autres possibilités seraient plus adéquates pour une configuration efficace ou VBA ne peut pas être utilisé sont exposés dans ce qui suit.

Scripts en VB et C

Les scripts en VB et C sont actif seulement en Runtime et sont utilisés pour la dynamisation de propriétés de vues et d'objets.

Assistants Dynamic

VBA ne remplace pas les assistants Dynamic. VBA vous offre cependant la possibilité d'étendre la fonctionnalité des assistants Dynamic de manière confortable.

ODK

ODK sont des appels de fonction permettant l'accès à toutes les fonctionnalités de WinCC aussi bien dans le système de configuration qu'en Runtime. Contrairement à ODK, VBA se caractérise par l'accès simplifié aux des objets de Graphics Designer.

Voir aussi

VBA dans le Graphics Designer (Page 1614)

Exécution de macros VBA dans le Graphics Designer (Page 1612)

Organisation du code VBA dans un projet WinCC (Page 1607)

Introduction : Utilisation de VBA dans WinCC (Page 1606)

3.2.3 Organisation du code VBA dans un projet WinCC

Introduction

L'organisation du code VBA de votre projet WinCC s'effectue dans l'Editeur VBA. C'est là que vous définissez si le code VBA doit être placé dans une seule vue, dans l'ensemble du projet ou dans tous les projets. Selon l'endroit où vous placez le code VBA, celui-ci est dit :

- code VBA général,
- code VBA spécifique à un projet, ou
- code VBA spécifique à une vue.

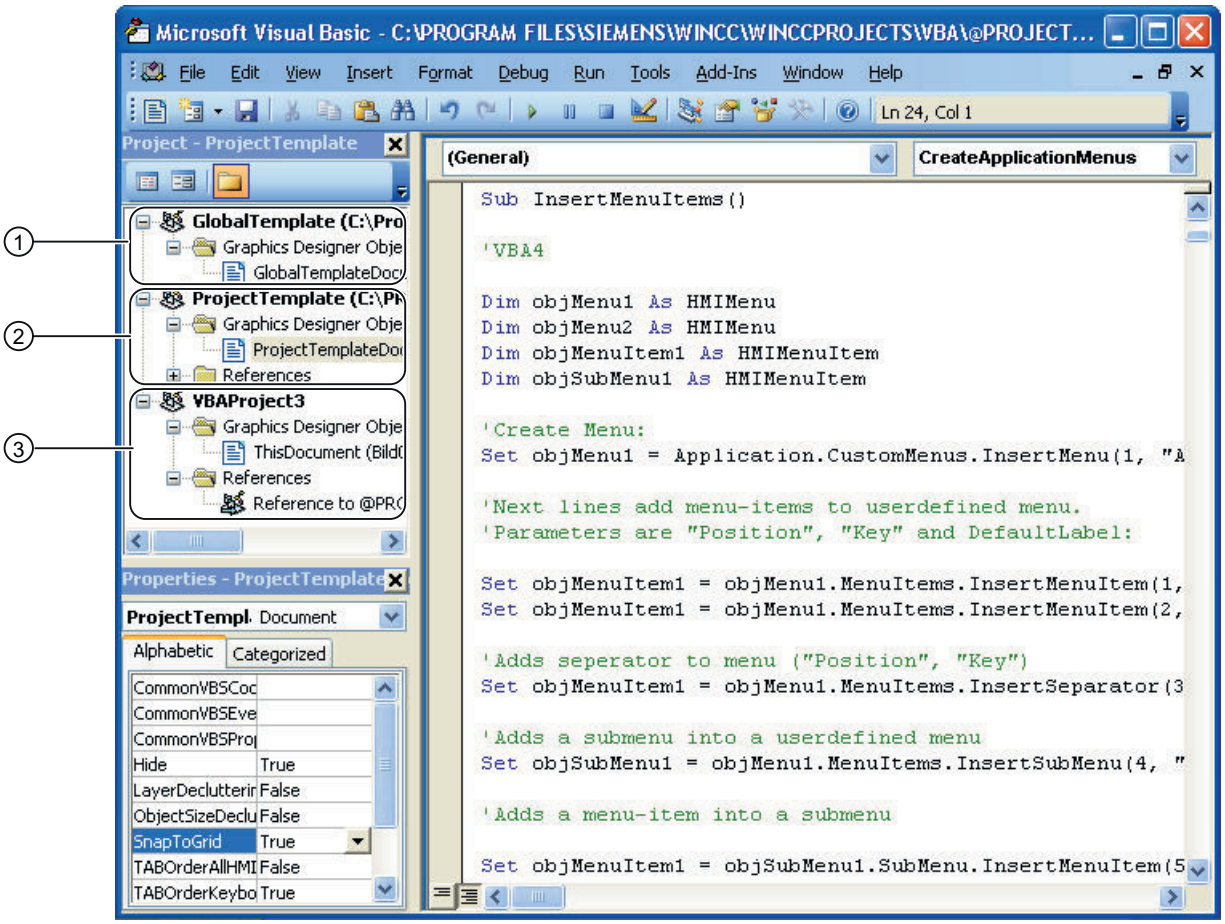
Remarque

Une vue du Graphics Designer est appelée "Document" dans le modèle d'objets VBA.

L'Editeur VBA

Dans Graphics Designer, démarrez l'Editeur VBA avec <Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor". Si vous n'avez pas encore ouvert de vue dans le Graphics Designer, vous ne pouvez éditer que le code VBA général ou le code VBA spécifique à un projet.

Les données générales et spécifiques à un projet, ainsi que toutes les vues ouvertes, sont affichées dans l'Explorateur de projets :



Code VBA général(1)

Désigne le code VBA que vous éditez par l'Editeur VBA dans le document "GlobalTemplateDocument". Ce code VBA est enregistré dans le fichier "@GLOBAL.PDT" qui se trouve dans le répertoire d'installation de WinCC.

Dans "GlobalTemplateDocument", vous positionnez le code VBA qui doit être disponible pour tous les projets WinCC qui se trouvent sur votre ordinateur. Si vous avez besoin de ce code VBA sur un autre ordinateur, utilisez les fonction d'exportation/importation de l'Editeur VBA.

Un ordinateur WinCC utilise exclusivement le fichier @GLOBAL.PDT résidant localement dans le répertoire d'installation (…\Siemens\WinCC\Templates) de WinCC.

Remarque

Si vous installez une mise à jour, votre modèle global "@Global.pdt" sera enregistré dans le fichier de sauvegarde "@Global.sav". Le fichier de sauvegarde est enregistré dans le répertoire …\Siemens\WinCC\Templates. Votre code VBA de l'ancien modèle global n'est repris pas automatiquement dans le nouveau modèle global.

Reprise du code VBA de l'ancien modèle global :

Pour adopter le code VBA de l'ancien modèle après avoir installé une mise à jour, procédez comme suit :

1. Si vous avez déjà spécifié le code VBA dans le nouveau modèle global, ouvrez l'éditeur VBA dans Graphics Designer et copiez le code VBA.
2. Fermez WinCC.
3. Dans l'Explorateur Windows, ouvrez le répertoire …\Siemens\WinCC\Templates.
4. Supprimez le nouveau modèle global "@Global.pdt".
5. Renommez le fichier de sauvegarde "@Global.sav" en "@Global.pdt".
6. Si vous avez déjà copié le code VBA du nouveau modèle global, ouvrez l'éditeur VBA dans Graphics Designer et collez le code VBA copié.

Le code VBA de votre ancien modèle global est maintenant à nouveau disponible.

Code VBA spécifique à un projet (2)

Désigne le code VBA que vous éditez par l'Editeur VBA dans le document "ProjectTemplateDocument". Ce code VBA est enregistré dans le fichier "@PROJECT.PDT" qui se trouve dans le répertoire principal de tout projet WinCC.

Le fichier "@PROJECT.PDT" se réfère au fichier "@GLOBAL.PDT". Les fonctions et les procédures que vous avez enregistrées dans le fichier "@GLOBAL.PDT" peuvent être appelées directement à partir de "ProjectTemplateDocument".

Dans "ProjectTemplateDocument", vous positionnez le code VBA que vous voulez utiliser dans toutes les vues du projet ouvert. Si vous avez besoin de ce code VBA sur un autre ordinateur, utilisez les fonction d'exportation/importation de l'Editeur VBA.

Vous pouvez ouvrir et éditer un fichier "@PROJECT.PDT" comme un fichier PDL. Vous pouvez ainsi utiliser le fichier "@PROJECT.PDT" comme modèle : Vous pouvez p. ex. y créer la vue de base de votre installation qui sera alors automatiquement transférée dans chaque nouveau fichier PDL du projet. Les propriétés des vues, comme les niveaux et le Zoom, ainsi que le code VBA, ne sont pas reprises dans le fichier PDL..

Code VBA spécifique à une vue (3)

Désigne le code VBA que vous éditez par l'Editeur VBA dans le document "ThisDocument" de la vue correspondante. Ce code VBA est enregistré avec la vue sous forme d'un fichier PDL.

Le fichier PDL se réfère au fichier "@PROJECT.PDT". Les fonctions et les procédures que vous avez enregistrées dans le fichier "@PROJECT.PDT" peuvent être appelées directement

à partir de ce fichier PDL. Toutefois, vous n'avez pas accès aux fonctions et aux procédures qui sont enregistrées dans le fichier "@GLOBAL.PDT".

Remarque

Dans chaque document, vous pouvez créer des Modules, des Class Modules et des User Forms.

Vous pouvez protéger le code VBA d'un module par un mot de passe pour interdire tout accès illicite. A cet effet, sélectionnez l'option de menu "Outils" > "VBAObject Properties" dans l'Editeur VBA.

Particularités d'exécution des macros VBA

La règle suivante doit être respectée pour exécuter les macros VBA : Le code VBA spécifique de la vue sera exécuté avant celui du projet. Si vous voulez par exemple appeler une macro VBA qui est contenue à la fois dans la vue et dans le code VBA spécifique du projet, seule la macro de vue sera exécutée. Ceci permet d'interdire la double exécution de macros et de fonctions VBA et d'éviter ainsi des erreurs éventuelles.

Dans la gestion d'événements, la transmission des événements est activée par défaut. Vous pouvez désactiver cette transmission si vous souhaitez que la réaction à un événement ne concerne que le code VBA spécifique de la vue.

Vous trouverez d'autres informations à cet égard au chapitre "Gestion des événements".

Tester avec Debugger

Vous pouvez tester vos scripts en VB en Runtime avec le débogueur de l'éditeur VBA. Vous trouverez de plus amples information dans l'aide en ligne de l'éditeur VBA.

Voir aussi

Gestion des événements (Page 1704)

VBA dans le Graphics Designer (Page 1614)

Exécution de macros VBA dans le Graphics Designer (Page 1612)

Pour exporter ou importer un code VBA (Page 1611)

3.2.4 Pour exporter ou importer un code VBA

Principe

L'Editeur VBA vous permet d'exporter et d'importer un code VBA pour le transférer à un autre ordinateur. Ceci permet de conserver les références aux procédures et aux fonctions à appeler dans le cours du projet.

Remarque

Lorsque vous importez un code VBA, vous devez rajouter manuellement les références aux bibliothèques externes sur l'ordinateur cible.

Marche à suivre

Exportation d'un code VBA

1. Dans le Project Explorer de l'Editeur VBA, sélectionnez le module dont vous voulez exporter le code.
2. Sélectionnez la rubrique de menu "File" > "Export File".
3. Sélectionnez le chemin et donnez un nom au fichier.
4. Cliquez sur "Enregistrer".

Le code VBA est exporté dans un fichier. Le type de fichier dépend du module dont le code VBA a été exporté.

Exportation d'un code VBA

1. Dans le Project Explorer de l'Editeur VBA, sélectionnez le document dont vous voulez importer le code VBA.
2. Sélectionnez la rubrique de menu "File > "Import File".
3. Sélectionnez le fichier et cliquez sur "Ouvrir" pour importer le code VBA sous "ThisDocument" dans le dossier "Class Modules".
4. Dans le dossier "Class Modules", ouvrez le document "ThisDocument" et copiez le code VBA de ce document dans le document du projet cible.

Voir aussi

Organisation du code VBA dans un projet WinCC (Page 1607)

3.2.5 Exécution de macros VBA dans le Graphics Designer

Introduction

Il existe trois possibilités d'exécution de macros VBA dans le Graphics Designer :

- Gestion des événements
- Menu ou barre d'outils personnalisés
- Editeur VBA

Gestion des événements

Dans le Graphics Designer, la vue active ou la bibliothèque des modules, des événements prédéfinis peuvent intervenir (p. ex. l'ouverture d'une vue) auxquels vous pouvez réagir à l'aide de gestionnaires d'événement VBA. Ces événements n'interviennent que pendant le travail dans le Graphics Designer et n'ont aucun rapport avec les événements de la configuration des actions.

Dans l'exemple qui suit, un bref message doit apparaître à l'ouverture d'une vue. A cet effet, on utilise l'événement "Opened" :

```
Private Sub Document_Opened(CancelForwarding As Boolean)
MsgBox ("La vue a été ouverte !")
End Sub
```

Vous trouverez des informations supplémentaires sur la "Gestion des événements" aux rubriques "Gestion des événements" et "Evénements".

Menu ou barre d'outils personnalisés

Le VBA vous permet de créer des menus et des barres d'outils personnalisés dans le Graphics Designer. A chaque rubrique de menu ou icône d'outil personnalisé, vous pouvez attribuer une macro VBA qui sera exécutée lorsque vous cliquerez sur la rubrique ou l'icône. Ceci vous permet d'étendre les fonctionnalités du Graphics Designer en fonction de vos besoins.

Vous trouverez des informations supplémentaires sur les menus et les barres d'outils personnalisés à la rubrique "Création de menus et de barres d'outils personnalisés".

Editeur VBA

Dans l'Editeur VBA, vous pouvez démarrer une macro VBA avec <F5>. <F8> vous permet d'exécuter un macro VBA étape par étape.

Voir aussi

Référence VBA (Page 1724)
Gestion des événements (Page 1704)
Création de menus et de barres d'outils personnalisés (Page 1620)
VBA dans le Graphics Designer (Page 1614)
Organisation du code VBA dans un projet WinCC (Page 1607)
Introduction : Utilisation de VBA dans WinCC (Page 1606)

3.3 VBA dans le Graphics Designer

3.3.1 VBA dans le Graphics Designer

Introduction

Vous utilisez VBA dans le Graphic Designer pour automatiser des tâches récurrentes dans la conception d'un projet. Vous pouvez créer des menus et des barres d'outils personnalisés pour exécuter leurs macros de manière confortable.

En principe, toutes les tâches de conception dans le Graphics Designer qui sont normalement exécutées avec la souris peuvent être remplacées par des macros VBA. Ceci concerne en particulier les surfaces (niveaux et zoom) et l'édition d'objets dans des vues, y compris la dynamisation.

Adaptation du Graphics Designer avec VBA.

Le Graphics Designer est représenté dans VBA par l'objet Application. Avec VBA, vous pouvez configurer en plusieurs langues dans le Graphics Designer, créer des menus et des barres d'outils personnalisés et accéder à la bibliothèque des modules.

Traitement de vues avec VBA

Dans le Graphics Designer, une vue est représentée par l'objet Document.

Avec VBA, vous pouvez accéder aux propriétés de la vue et configurer les réglages de niveaux et de zoom. En outre, vous pouvez créer des menus et des barres d'outils spécifiques de la vue, lesquels ne seront visibles que tant que la vue sera active.

Edition d'objets avec VBA

Un objet d'une vue est représenté par l'objet HMIObject. Avec VBA, vous pouvez créer des objets, les supprimer et accéder aux propriétés de l'objet. Vous pouvez, par exemple, créer rapidement un grand nombre d'objets aux propriétés identiques pour votre vue d'installation.

Création de dynamisations avec VBA

VBA vous permet de dynamiser les propriétés et les événements de vues et d'objets.

Gestion des événements

VBA vous permet de réagir à des événements qui interviennent dans le Graphics Designer ou dans la vue lorsque, par exemple, vous insérez un nouvel objet dans une vue. Vous utilisez la gestion des événements pour exécuter des macros VBA dans des situations de programme déterminées.

Accès aux applications externes

VBA vous permet d'accéder à des programmes compatibles VBA, p. ex. les produits de la suite Microsoft-Office. Vous avez ici par ex. la possibilité de lire des valeurs d'une table Excel et de lui attribuer des propriétés d'objet.

Remarque

Accès à des applications compilées avec .net

Les applications compilées avec .net doivent être compilées encore une fois pour que l'accès à VBA dans Graphics Designer soit possible.

Voir aussi

Edition de vues avec VBA (Page 1647)

Objet SymbolLibrary (Page 2023)

Objet HMIObject (Page 1943)

Objet Document (Page 1908)

Objet Application (Page 1876)

Accès aux applications externes avec VBA (Page 1707)

Gestion des événements (Page 1704)

Création de dynamisations avec VBA (Page 1680)

Edition d'objets avec VBA (Page 1653)

Adaptation du Graphics Designer avec VBA. (Page 1616)

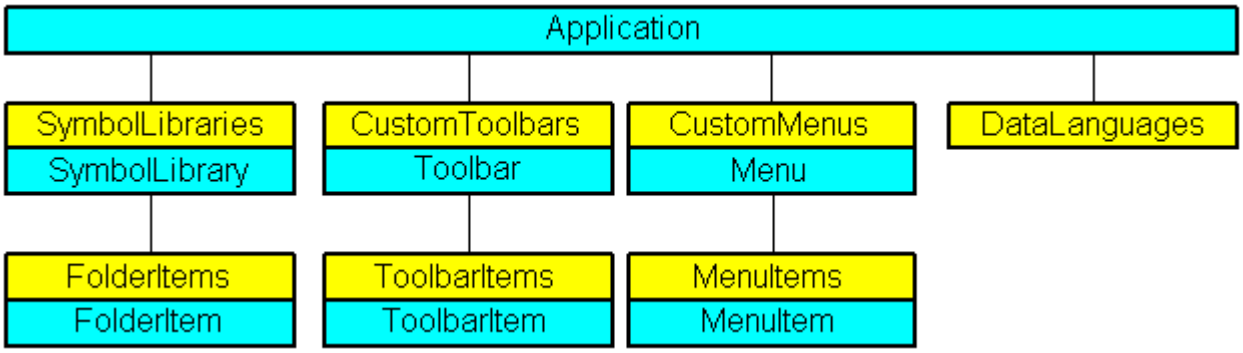
Introduction : Utilisation de VBA dans WinCC (Page 1606)

3.3.2 Adaptation du Graphics Designer avec VBA.

3.3.2.1 Adaptation du Graphics Designer avec VBA.

Introduction

Dans VBA, l'objet Application représente le Graphics Designer :



Accès à la bibliothèque des modules

Avec VBA, vous avez l'accès intégral à la bibliothèque des modules. Vous pouvez modifier la bibliothèque des modules, par exemple par création ou suppression de dossier, ainsi que copier des objets et les insérer dans une image.

Création de menus et de barres d'outils personnalisés

Pour exécuter des macros VBA dans le Graphics Designer, vous pouvez créer des menus et des barres d'outils personnalisés. Ceci vous permet d'adapter la fonctionnalité du Graphics Designer à vos besoins.

Conception multilingue

Avec VBA, vous pouvez travailler en plusieurs langues dans le Graphics Designer. A cet effet, vous avez accès aux propriétés linguistiques d'objet et vous pouvez créer des menus et des barres d'outils personnalisés en plusieurs langues.

Voir aussi

Edition de vues avec VBA (Page 1647)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Création de menus et de barres d'outils personnalisés (Page 1620)

Configuration multilingue avec VBA (Page 1617)

VBA dans le Graphics Designer (Page 1614)

3.3.2.2 Configuration multilingue avec VBA

Introduction

VBA vous permet de travailler en plusieurs langues dans le Graphics Designer. D'une part, vous pouvez accéder aux propriétés dépendant de la langue des objets et, d'autre part, vous pouvez présenter les menus et les barres d'outils personnalisés en plusieurs langues. Dans VBA, les diverses versions linguistiques de textes sont enregistrées dans un listage du type "LanguageTexts". Les configurations de polices spécifiques aux langues sont enregistrées dans un listage du type "LanguageFonts".

Vous trouverez des informations supplémentaires sur la configuration multilingue dans la documentation WinCC "Structure des projets multilingues".

Langue d'interface

Vous ne pouvez changer la langue d'interface que dans WinCC, pas avec VBA. Un changement de la langue d'interface dans WinCC déclenche l'événement "DesktopLanguageChanged". Vous pouvez adapter les menus et les barres d'outils personnalisés en fonction de l'utilisateur en remplaçant p. ex. les icônes d'outils en fonction de la langue à utiliser.

Les objets suivants et leurs propriétés linguistiques sont concernés par le changement de langue de l'interface :

- Objet FolderItem
- Objets Menu et MenuItem
- Objet ToolbarItem
- Vous trouverez des informations supplémentaires sur la langue de l'interface dans la documentation WinCC "Structure des projets multilingues" à la rubrique "Terminologie de WinCC".

Langue de configuration

Avec VBA, vous pouvez définir la langue de travail par la propriété "CurrentDataLanguage".

L'exemple qui suit est celui de la définition de l'anglais comme langue de travail :

```
Sub ChangeCurrentDataLanguage()
'VBA1
Application.CurrentDataLanguage = 1033
MsgBox "The Data language has been changed to english"
Application.CurrentDataLanguage = 1031
MsgBox "The Data language has been changed to german"
End Sub
```

Toutes les propriétés qui dépendent de la langue, comme p. ex. ToolTipText, sont concernées par le changement.

Conception d'un projet multilingue dans VBA

Avec VBA, vous avez deux possibilités de conception multilingue

- Changement de langue : propriétés textuelles des objets.
- Liste des Language Texts : propriétés de texte de menus, de barres d'outils ainsi que d'objets personnalisés.

Changement de langue

VBA vous permet de modifier les propriétés linguistiques (p. ex. "Text") des objets. A cet effet, vous attribuez d'abord le texte à la propriété considérée et, ensuite,vous changez la langue de travail pour attribuer le texte dans la langue considérée.

Listage LanguageTexts

Vous pouvez enregistrer les textes multilingues de l'objet considéré directement dans le listage afférent du type "LanguageTexts". A cet effet, vous saisissez le code de la langue et le texte afférent.

Vous trouverez dans la documentation WinCC la liste des codes de langue (Index > Language Code).

L'exemple qui suit est celui de l'attribution d'un texte en anglais et en allemand au bouton "myButton" :

```
Sub AddLanguagesToButton()
'VBA2
Dim objLabelText As HMILanguageText
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
'
'Set defaultlabel:
objButton.Text = "Default-Text"
'
'Add english label:
Set objLabelText = objButton.LDTexts.Add(1033, "English Text")
'Add german label:
Set objLabelText = objButton.LDTexts.Add(1031, "German Text")
End Sub
```

Voir aussi

Objet LanguageTexts (liste) (Page 1954)

Objet LanguageFonts (liste) (Page 1951)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Pour créer des menus multilingues (Page 1626)

Référence VBA (Page 1724)

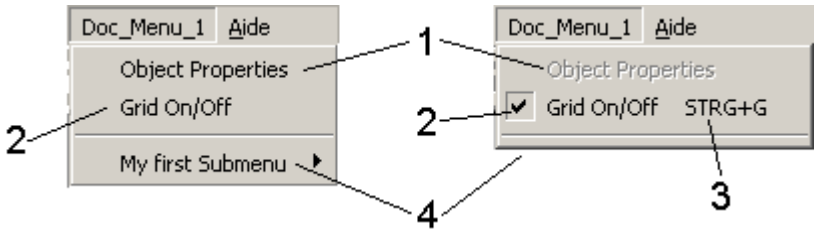
3.3.2.3 Création de menus et de barres d'outils personnalisés

Configuration des menus et des barres d'outils

Introduction

Vous pouvez configurer les menus et les barres d'outils personnalisés pour qu'ils réagissent à certaines situations de programme dans le Graphics Designer. Par exemple, s'il faut qu'une icône d'outil ne soit pas disponible lorsqu'aucun objet n'est sélectionné, vous pouvez la masquer en grisé. Une coche devant une option de menu peut, p. ex., indiquer qu'une sélection est activée.

L'illustration suivante indique les possibilités de configuration à l'aide d'un exemple :



Actif (oui/non) (1)

Active l'option ou la masque en grisé. Vous pouvez utiliser la propriété "Enabled" pour les menus, les options de menu et les icônes d'outils que vous avez personnalisés :

```
'VBA13
Application.ActiveDocument.CustomMenus(1).MenuItems(1).Enabled = False
```

Coché (oui/non) (2)

Coche l'option de menu. Vous ne pouvez utiliser la propriété Checked que pour les options d'un menu personnalisé :

```
'VBA14
Application.ActiveDocument.CustomMenus(1).MenuItems(2).Checked = True
```

Shortcut(3)

Définit un raccourci clavier affecté à une option de menu ou à une icône d'outil. Vous ne pouvez utiliser la propriété "Shortcut" que pour les options de menu et les icônes d'outil que vous avez personnalisés :

```
'VBA15
Application.ActiveDocument.CustomMenus(1).MenuItems(3).Shortcut = "Ctrl+G"
```

Visible (oui/non) (4)

Affiche ou masque l'option en grisé. Vous ne pouvez utiliser la propriété "Visible" que pour les menus, options de menu et barres d'outils que vous avez personnalisés, ainsi que pour leurs icônes :

```
'VBA16
Application.ActiveDocument.CustomMenus(1).MenuItems(4).Visible = False
```

Création de menus et de barres d'outils personnalisés

Menus et barres d'outils personnalisés dans le Graphics Designer

Dans le Graphics Designer, vous pouvez utiliser des menus et des barres d'outils personnalisés pour exécuter des macros VBA. A cet égard, on distingue les menus et barres d'outils spécifiques à une application, et ceux spécifiques à une vue. Leurs propriétés sont les suivantes :

- Menu/Barre d'outils spécifique à une application : toujours visible lorsque le Graphics Designer est ouvert. On utilise les menus et les barres d'outils spécifiques à une application lorsque les macros VBA correspondantes à exécuter doivent être accessibles en permanence.
- Menu/Barre d'outils spécifique à une vue : est lié à une vue et n'est visible que tant que cette vue est active. On utilise les menus et les barres d'outils spécifiques à une vue lorsque les macros VBA appliquées ne concernent que la vue en question.

Positionnement des menus et barres d'outils personnalisés

Dans les menus personnalisés, le paramètre "Position" définit la positon définitive de la barre de menu. Toutefois,

- dans le Graphics Designer, les menus personnalisés sont toujours positionnés à droite du menu "Fenêtre", tandis que

les menus spécifiques à une vue sont toujours positionnés à gauche du menu "?".

Les barres d'outils spécifiques à une application ne sont pas traitées de façon "préférentielle" : C'est l'ordre dans lequel vous insérez les barres d'outils qui est décisif. Les barres d'outils sont positionnés sous la barre d'outils du Graphics Designer :

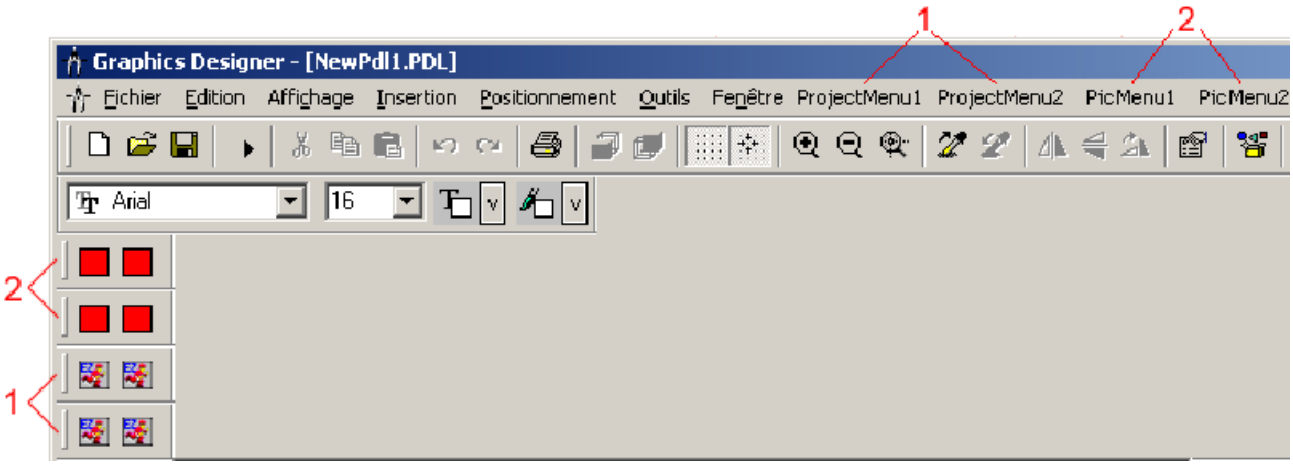


Figure 3-1 Représentation : menus et de barres d'outils personnalisés. (1) spécifiques à une application, (2) spécifiques à une vue

Dans l'illustration ci-dessus, on a inséré d'abord les menus et barres d'outils personnalisés spécifiques à une vue, puis ceux spécifiques à une application.

Propriétés des menus et barres d'outils personnalisés

Dans les menus et barres d'outils personnalisés, vous pouvez utiliser des tirets pour décomposer les options, p. ex. par catégories. Vous pouvez aussi créer des sous-menus dans un menu personnalisé.

Pour les menus et les barres d'outils personnalisés (et leurs entrées), vous disposez des choix de configuration suivants :

- Visible (oui/non) : Affiche ou masque l'option (propriété Visible).
- Activé (oui/non) : Active l'option ou la masque en grisé (propriété Enabled).
- Coché (oui/non) : uniquement pour les options de menu (propriété Checked).
- Shortcut : combinaison de touches pour appeler l'option de menu (propriété ShortCut).
- Texte d'état : texte affiché dans la barre d'état (propriété StatusText).
- Texte info-bulle - disponible uniquement pour les icônes d'outil (propriété ToolTipText).

Par exemple, vous pouvez masquer une option de menu lorsque la macro n'est pas exécutable à un certain moment. Ceci permet d'éviter des erreurs de commande.

Vous pouvez créer en plusieurs langues tous les textes et toutes les légendes des menus et des barres d'outils personnalisés afin que ces derniers puissent aussi réagir à une instruction de changement de langue.

Voir aussi

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)

Pour créer une nouvelle barre d'outils spécifique à une application (Page 1629)

Pour créer des menus multilingues (Page 1626)

Pour insérer une nouvelle option de menu (Page 1623)

Pour créer des menus et des barres d'outils spécifiques à une vue (Page 1648)

Pour créer un nouveau menu spécifique à une application (Page 1622)

Configuration des menus et des barres d'outils (Page 1619)

Exécution de macros VBA dans le Graphics Designer (Page 1612)

Pour créer un nouveau menu spécifique à une application

Introduction

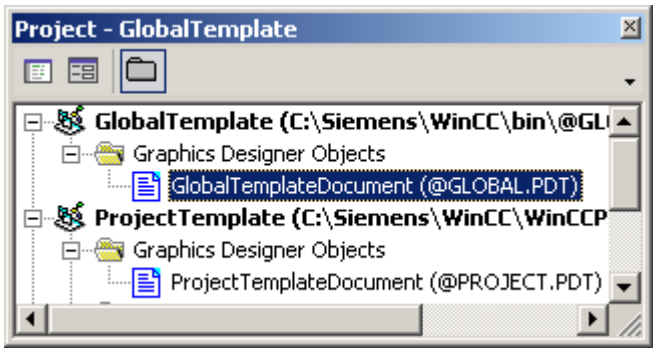
Les menus spécifiques à une application demeurent visibles même après fermeture de toutes les vues dans le Graphics Designer. Pour insérer un menu spécifique à une application au démarrage, utilisez p. ex. l'événement Started.

Placez le code VBA :

- dans "GlobalTemplateDocument" si le menu doit être disponible dans tous les projets, ou
- dans "ProjectTemplateDocument" si le menu doit être disponible dans le projet en cours.

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



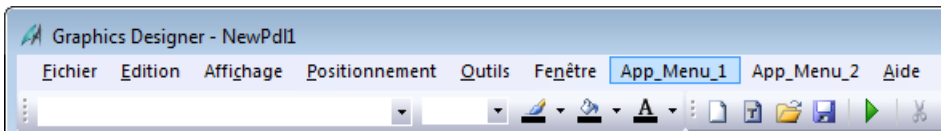
3. Pour créer un menu personnalisé dans le Graphics Designer, insérez p. ex. la procédure "CreateApplicationMenus()" dans le document. L'exemple qui suit est celui de la création de deux menus personnalisés :

```
Sub CreateApplicationMenus()
'VBA3
'Declaration of menus...:
Dim objMenu1 As HMIMenu
Dim objMenu2 As HMIMenu
'
'Add menus. Parameters are "Position", "Key" and "DefaultLabel":
Set objMenu1 = Application.CustomMenus.InsertMenu(1, "AppMenu1",
"App_Menu_1")
Set objMenu2 = Application.CustomMenus.InsertMenu(2, "AppMenu2",
"App_Menu_2")
End Sub
```

4. Démarrez la procédure avec <F5>.

Résultat

Les deux menus "App_Menu_1" et "App_Menu_2" sont insérés à droite du menu "Fenêtre" :



Voir aussi

Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode InsertMenu (Page 1825)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Pour créer des menus multilingues (Page 1626)

Pour insérer une nouvelle option de menu (Page 1623)

Configuration des menus et des barres d'outils (Page 1619)

Organisation du code VBA dans un projet WinCC (Page 1607)

Pour insérer une nouvelle option de menu

Condition préalable

Vous devez avoir créé un menu personnalisé.

Introduction

Dans un menu personnalisé, vous disposez de trois types d'insertion :

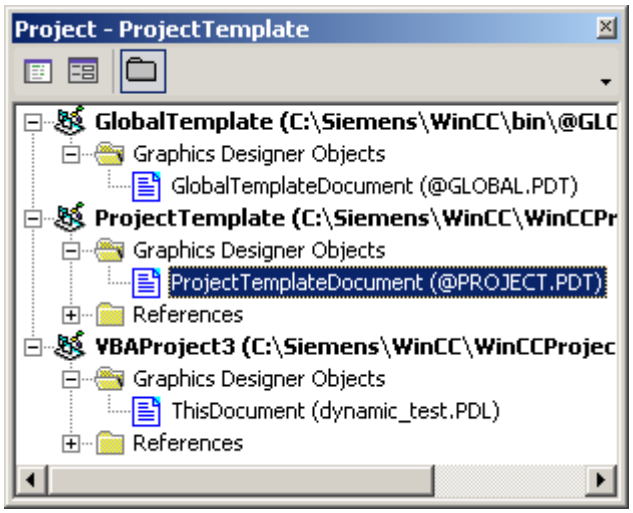
- Inscription au menu : pour la sélection des macros VBA.
- Ligne de séparation : pour améliorer la lisibilité de la structure du menu personnalisé.
- Sous-menu : similaire à un menu personnalisé (p. ex. pour l'articulation des commandes).

Dans le menu personnalisé, le paramètre "Position" détermine l'ordre des options de menu.

Le paramètre "Key" identifie l'option de menu de manière univoque. Ce paramètre est utilisé lorsque les macros VBA sont appelées par l'événement "MenuItemClicked".

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



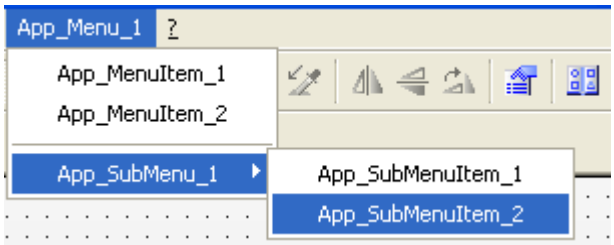
3. Pour créer des options de menu dans un menu personnalisé existant, insérez p. ex. la procédure "InsertMenuItems()" dans le document. L'exemple qui suit est celui de la création de quelques options de menu dans le menu personnalisé "App_Menu_1" :

```
Sub InsertMenuItems()
'VBA4
Dim objMenu1 As HMIMenu
Dim objMenu2 As HMIMenu
Dim objMenuItem1 As HMIMenuItem
Dim objSubMenu1 As HMIMenuItem
'Create Menu:
Set objMenu1 = Application.CustomMenus.InsertMenu(1, "AppMenu1",
"App_Menu_1")
'Next lines add menu-items to userdefined menu.
'Parameters are "Position", "Key" and DefaultLabel:
Set objMenuItem1 = objMenu1.MenuItems.InsertMenuItem(1,
"mItem1_1", "App_MenuItem_1")
Set objMenuItem1 = objMenu1.MenuItems.InsertMenuItem(2,
"mItem1_2", "App_MenuItem_2")
'
'Adds seperator to menu ("Position", "Key")
Set objMenuItem1 = objMenu1.MenuItems.InsertSeparator(3,
"mItem1_3")
'
'Adds a submenu into a userdefined menu
Set objSubMenu1 = objMenu1.MenuItems.InsertSubMenu(4, "mItem1_4",
"App_SubMenu_1")
'
'Adds a menu-item into a submenu
Set objMenuItem1 = objSubMenu1.SubMenu.InsertMenuItem(5,
"mItem1_5", "App_SubMenuItem_1")
Set objMenuItem1 = objSubMenu1.SubMenu.InsertMenuItem(6,
"mItem1_6", "App_SubMenuItem_2")
End Sub
```

4. Démarrez la procédure avec <F5>.

Résultat

La procédure "InsertMenuItems()" insère le menu "App_Menu_1" avec ces options de menu :



Voir aussi

Méthode InsertSeparator (Page 1828)

Méthode InsertSubmenu (Page 1829)

Méthode InsertMenu (Page 1825)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Pour créer des menus multilingues (Page 1626)

Pour créer un nouveau menu spécifique à une application (Page 1622)

Configuration des menus et des barres d'outils (Page 1619)

Création de menus et de barres d'outils personnalisés (Page 1620)

Pour créer des menus multilingues

Introduction

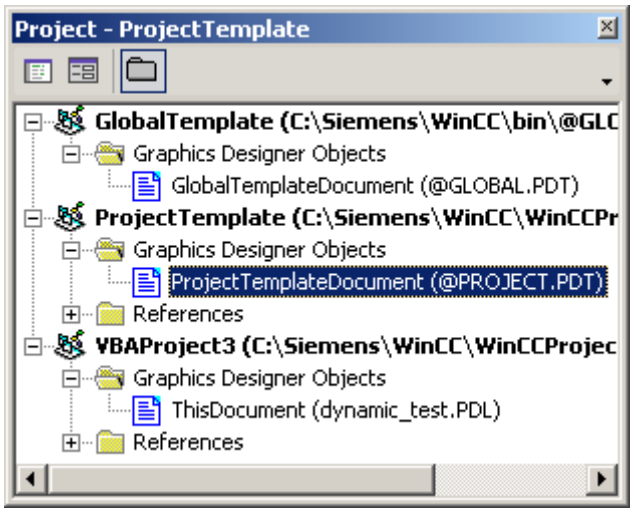
Vous pouvez créer un menu personnalisé qui réagisse à un changement de langue. A cet effet, vous devez définir, pour le menu et chacune de ses commandes, le nombre souhaité de légendes linguistiques.

Une légende linguistique est composée de le code de la langue (LCID) et d'un texte dans la langue considérée (DisplayName).

Vous trouverez dans la documentation WinCC la liste des codes de langue (Index > Language Code).

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



3. Pour définir des légendes linguistiques pour un menu personnalisé, insérez p. ex. la procédure MultipleLanguagesForAppMenu1() dans le document. L'exemple qui suit est celui de la définition de légendes anglaises pour le menu App_Menu_1 :

```
Sub InsertMenuItems()
'VBA5
'Execute this procedure first
Dim objMenu1 As HMIMenu
Dim objMenu2 As HMIMenu
Dim objMenuItem1 As HMIMenuItem
Dim objSubMenu1 As HMIMenuItem
'Insert Menu:
Set objMenu1 = Application.CustomMenus.InsertMenu(1, "AppMenu1",
"App_Menu_1")
'Next lines inserts menu-items to userdefined menu.
'parameters are "Position", "Key" and DefaultLabel:
Set objMenuItem1 = objMenu1.MenuItems.InsertMenuItem(1,
"mItem1_1", "App_MenuItem_1")
Set objMenuItem1 = objMenu1.MenuItems.InsertMenuItem(2,
"mItem1_2", "App_MenuItem_2")
'
'Inserts seperator into menu ("Position", "Key")
Set objMenuItem1 = objMenu1.MenuItems.InsertSeparator(3,
"mItem1_3")
'
'Inserts a submenu into a userdefined menu
Set objSubMenu1 = objMenu1.MenuItems.InsertSubMenu(4, "mItem1_4",
"App_SubMenu_1")
'
'Inserts a menu-item into a submenu
Set objMenuItem1 = objSubMenu1.SubMenu.InsertMenuItem(5,
"mItem1_5", "App_SubMenuItem_1")
Set objMenuItem1 = objSubMenu1.SubMenu.InsertMenuItem(6,
"mItem1_6", "App_SubMenuItem_2")
End Sub
Sub MultipleLanguagesForAppMenu1()
'   execute this procedure after "InsertMenuItems()" was run
'Object "objLanguageTextMenu1" contains the
'foreign-language labels for the menu
Dim objLanguageTextMenu1 As HMILanguageText
'
'Object "objLanguageTextMenu1Item" contains the
'foreign-language labels for the menu-items
Dim objLanguageTextMenuItem1 As HMILanguageText
Dim objMenu1 As HMIMenu
Dim objSubMenu1 As HMIMenuItem
Set objMenu1 = Application.CustomMenus("AppMenu1")
Set objSubMenu1 =
Application.CustomMenus("AppMenu1").MenuItems("mItem1_4")
'
'Inserts foreign-language label into a menu:
'("Add(LCID, DisplayName)"-Methode:
```

```
Set objLanguageTextMenu1 = objMenu1.LDLabelTexts.Add(1033,
"English_App_Menu_1")
'
'Inserts foreign-language label into a menuitem:
Set objLanguageTextMenuItem1 =
objMenu1.MenuItems("mItem1_1").LDLabelTexts.Add(1033, "My first
menu item")
'
'Adds a foreign-language label into a submenu:
Set objLanguageTextMenuItem1 =
objSubMenu1.SubMenu.Item("mItem1_5").LDLabelTexts.Add(1033, "My
first submenu item")
End Sub
```

4. Démarrez la procédure avec <F5>.

Résultat

Si, maintenant, vous choisissez l'anglais comme langue de travail, certaines options du menu personnalisé seront affichées en anglais.

Voir aussi

Objet LanguageTexts (liste) (Page 1954)

Propriété LDLabelTexts (Page 2248)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Pour insérer une nouvelle option de menu (Page 1623)

Pour créer un nouveau menu spécifique à une application (Page 1622)

Configuration des menus et des barres d'outils (Page 1619)

Création de menus et de barres d'outils personnalisés (Page 1620)

Configuration multilingue avec VBA (Page 1617)

Pour créer une nouvelle barre d'outils spécifique à une application

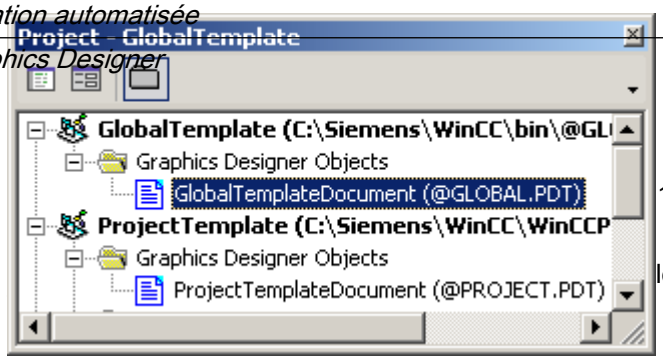
Introduction

Les barres d'outils spécifiques à une application demeurent visibles même après fermeture de toutes les vues dans le Graphics Designer.

Placez le code VBA :

- dans GlobalTemplateDocument si la barre d'outils doit être disponible dans tous les projets, ou
- dans ProjectTemplateDocument si la barre d'outils doit être disponible dans le projet en cours.

Marche à suivre



1. 1> ou "Outils" > "Macros" > "Visual
2. lequel vous voulez créer le code
3. Pour créer une barre d'outils personnalisés dans le Graphics Designer, insérez p. ex. la procédure CreateApplicationToolbars() dans le document. L'exemple suivant est celui de la création de deux barres d'outils personnalisées :

```
Sub CreateApplicationToolbars()
'VBA6
'Declare toolbar-objects...:
Dim objToolbar1 As HMIToolbar
Dim objToolbar2 As HMIToolbar
'
'Add the toolbars with parameter "Key"
Set objToolbar1 = Application.CustomToolbars.Add("AppToolbar1")
Set objToolbar2 = Application.CustomToolbars.Add("AppToolbar2")
End Sub
```

4. Démarrez la procédure avec <F5>.

Résultat

Les deux barres d'outils sont insérées sous les barres d'outils du Graphics Designers.

Voir aussi

Méthode Add (listage CustomToolbars) (Page 1768)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)

Configuration des menus et des barres d'outils (Page 1619)

Création de menus et de barres d'outils personnalisés (Page 1620)

Insertion d'une nouvelle icône d'outil dans une barre d'outils

Condition préalable

Vous devez avoir créé la barre d'outils personnalisée.

Introduction

Dans une barre d'outils personnalisée, vous pouvez insérer deux types d'objet :

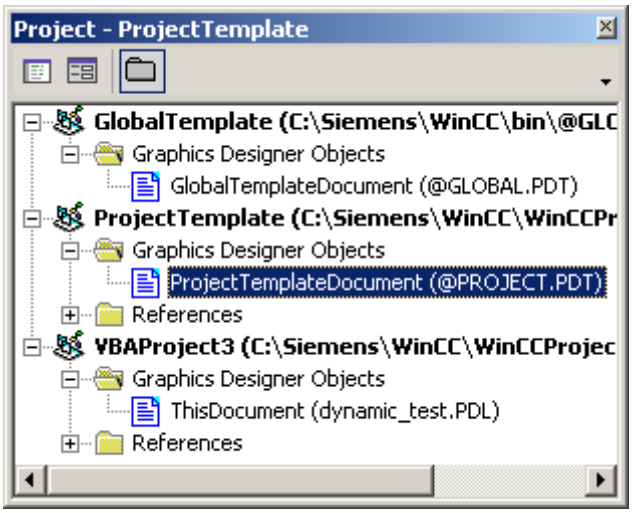
- Icône : pour la sélection des macros VBA.
- Ligne de séparation : pour améliorer la lisibilité de la structure de la barre d'outils personnalisée.

Dans la barre d'outils personnalisée, le paramètre "Position" définit l'ordre des icônes d'outil.

Le paramètre "Key" identifie l'icône d'outil de manière univoque. Ce paramètre est utilisé lorsque les macros ToolbarItemClicked sont appelées par l'événement VBA.

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



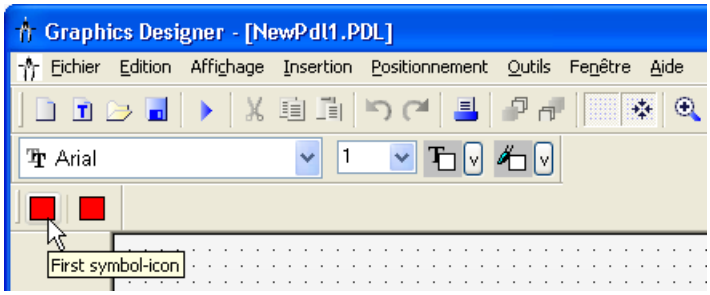
3. Pour créer des icônes dans une barre d'outils personnalisée existante, insérez, p. ex., la procédure "InsertToolbarItems()" dans le document. L'exemple qui suit est celui de la création de deux icônes d'outil dans la barre d'outils personnalisée "AppToolbar1", séparées par une ligne de séparation :

```
Sub InsertToolbarItems()
'VBA7
Dim objToolbar1 As HMIToolbar
Dim objToolbarItem1 As HMIToolbarItem
'
'Add a new toolbar:
Set objToolbar1 = Application.CustomToolbars.Add("AppToolbar1")
'Adds two toolbar-items to the toolbar
'("InsertToolbarItem(Position, Key, DefaultToolTipText)"-Methode):
Set objToolbarItem1 =
objToolbar1.ToolbarItems.InsertToolbarItem(1, "tItem1_1", "First
Symbol-Icon")
Set objToolbarItem1 =
objToolbar1.ToolbarItems.InsertToolbarItem(3, "tItem1_2", "Second
Symbol-Icon")
'
'Adds a seperator between the two toolbar-items
'("InsertSeparator(Position, Key)"-Methode):
Set objToolbarItem1 = objToolbar1.ToolbarItems.InsertSeparator(2,
"tSeparator1_3")
End Sub
```

4. Démarrez la procédure avec <F5>.

Résultat

La procédure "InsertToolbarItems()" ajoute aux barres d'outils du Graphics Designer une nouvelle barre d'outils comportant deux icônes séparées par une ligne de séparation :



Remarque

Pour déterminer un graphisme (*.ICO-Format) pour un icône d'outil, utilisez la propriété d'icône.

Voir aussi

Création de menus et de barres d'outils personnalisés (Page 1620)

Propriété Icon (Page 2206)

Méthode InsertSeparator (Page 1828)

Méthode InsertToolbarItem (Page 1831)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Pour créer une nouvelle barre d'outils spécifique à une application (Page 1629)

Configuration des menus et des barres d'outils (Page 1619)

Pour affecter des textes d'aide à des menus et à des barres d'outils

Condition préalable

Vous devez avoir créé le menu ou la barre d'outils personnalisés.

Introduction

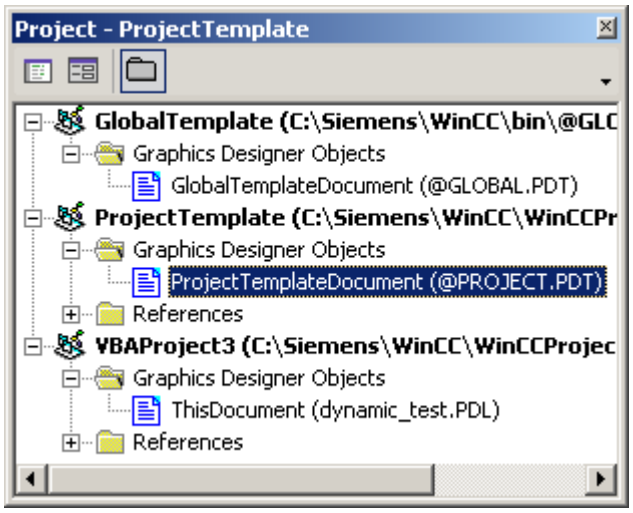
Vous pouvez préparer des textes d'aide qui expliquent la marche à suivre au projecteur au moment où celui-ci passe le pointeur de la souris sur une option d'un menu personnalisé ou sur une icône personnalisée :

- Pour les options de menu et les icônes d'outil personnalisées, vous pouvez définir des textes d'aide qui s'afficheront dans la barre d'état.
- Pour les icônes d'outil personnalisées, créez le texte d'aide par défaut comme info-bulle.

Les textes d'état et les textes d'étiquettes peuvent être rédigés dans des langues différentes.

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



3. Pour affecter un texte d'état à une option d'un menu personnalisé, insérez p. ex. la procédure AddStatusTextsToAppMenu1() dans le document. L'exemple qui suit est celui de l'affectation d'un texte d'état en allemand et en anglais à la première option du menu AppMenu1 précédemment créé :

```
Sub AddStatusTextsToAppMenu1()
'VBA8
Dim objMenu1 As HMIMenu
'
'Object "objStatusTextMenuItem1" contains foreign-language texts
Dim objStatusTextMenuItem1 As HMILanguageText
Set objMenu1 = Application.CustomMenus("AppMenu1")
'
'Assign a statustext to a menuitem:
objMenu1.MenuItems("mItem1_1").StatusText = "Statustext the first
menuitem"
'
'Assign a foreign statustext to a menuitem:
Set objStatusTextMenuItem1 =
objMenu1.MenuItems("mItem1_1").LDStatusTexts.Add(1033, "This is
my first status text in English")
End Sub
```

4. Pour affecter un texte d'état et d'info-bulle en langue étrangère à une icône personnalisée, insérez p. ex. la procédure "AddStatusAndTooltipTextsToAppToolbar1()" dans le document. Dans cet exemple, un texte d'état (français/anglais) et texte d'info-bulle en anglais a été affecté à la première icône de la barre d'outils précédemment créée :
```
Sub AddStatusAndTooltipTextsToAppToolbar1()
'VBA9
Dim objToolbar1 As HMIToolbar
'
'Variable "StatusTextToolbarItem1" for foreign statustexts
Dim objStatusTextToolbarItem1 As HMILanguageText
'
'Variable "TooltipTextToolbarItem1 for foreign tooltiptexts
Dim objTooltipTextToolbarItem1 As HMILanguageText
Set objToolbar1 = Application.CustomToolbars("AppToolbar1")
'
'Assign a statustext to a toolbaritem:
objToolbar1.ToolbarItems("tItem1_1").StatusText = "Texte d'état
pour le premier icône"
'
'Assign a foreign statustext to a toolbaritem:
Set objStatusTextToolbarItem1 =
objToolbar1.ToolbarItems("tItem1_1").LDStatusTexts.Add(1033,
"This is my first status text in English")
'
'Assign a foreign tooltiptext to a toolbaritem:
Set objTooltipTextToolbarItem1 =
objToolbar1.ToolbarItems("tItem1_1").LDTooltipTexts.Add(1033,
"This is my first tooltip text in English")
End Sub
```
5. Démarrez chaque procédure avec <F5>.

Résultat

Le texte d'état s'affiche lorsque vous passez le pointeur de la souris sur l'option de menu personnalisée ou sur l'icône d'outil.

Voir aussi

Propriété LDTooltipTexts (Page 2252)

Propriété LDStatusTexts (Page 2250)

Objet LanguageTexts (liste) (Page 1954)

Méthode Add (Page 1765)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)

Pour insérer une nouvelle option de menu (Page 1623)

Configuration des menus et des barres d'outils (Page 1619)

Création de menus et de barres d'outils personnalisés (Page 1620)

Pour affecter des macros VBA à des menus et à des barres d'outils

Introduction

Pour affecter des macros VBA à des menus ou à des barres d'outils personnalisés, vous avez le choix entre :

- utiliser les gestionnaires d'événement VBA MenuItemClicked et ToolbarItemClicked, ou bien
- la propriété Macro.

Remarque

Vous trouverez le code VBA pour créer les menus et les barre d'outils personnalisés dans cette documentation aux rubriques "Création de nouvelles options de menu" et "Insertion d'une nouvel icône d'outil dans une barre d'outils".

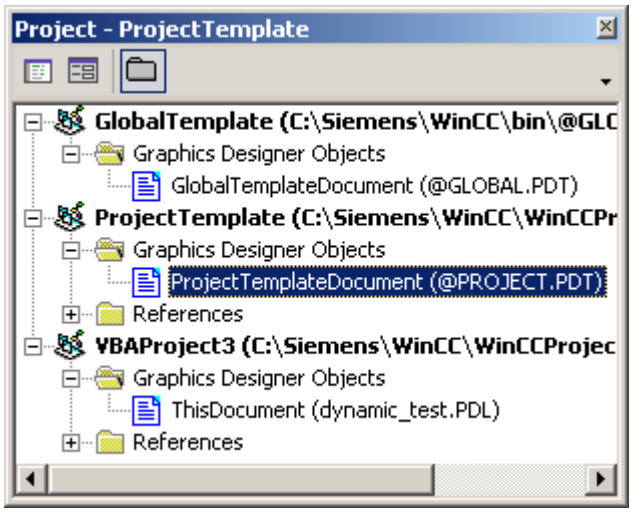
Marche à suivre

Affectation de la macro VBA par gestionnaire d'événement VBA

Remarque

Vous trouverez des informations supplémentaires sur les gestionnaires d'événements à rubrique Gestion des événements.

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :

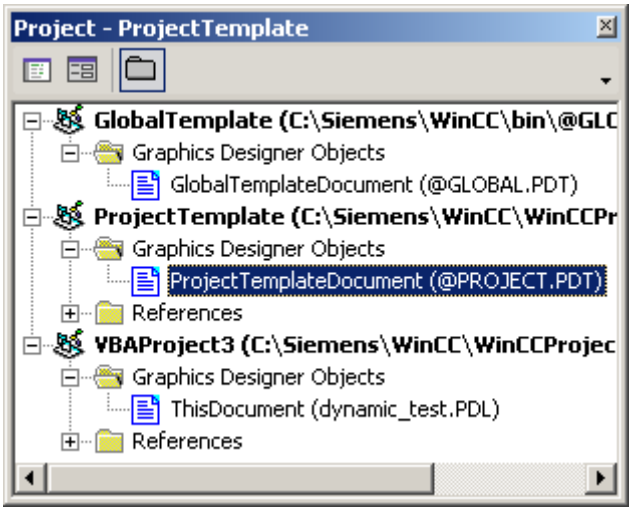


3. Pour démarrer une macro VBA par un déclencheur de macro VBA, utilisez. les événements MenuItemClicked ou ToolbarItemClicked :

4. Ajoutez le code VBA du tableau "VBA10".
5. Démarrez chaque procédure avec <F5>.

Affectation de la macro VBA par la propriété Macro

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



3. Pour démarrer une macro VBA par la propriété Macro, affectez la macro VBA à chaque option de menu ou icône d'outil. Dans l'exemple qui suit, un menu définie par l'utilisateur est créé avec deux options qui appellent deux macros VBA différentes :
   Le code VBA de l'exemple suivant VBA11 dépend du type de fichier. Le code VBA est ajouté comme exemple pour un fichier PDL et PDT. Les deux cas diffèrent comme suit :
   - Fichier PDL :
     Le code VBA dans un fichier PDL sera exécuté seulement si le fichier PDL est affiché à l'instant.
   - Fichier PDT :
     Le code VBA dans un fichier PDT sera toujours exécuté quand le Graphics Designer est ouvert.
4. Ajoutez le code VBA du tableau "VBA11" : Code-exemple pour fichier PDL" ou "VBA821 : Code-exemple pour fichier PDT".
   Vous pouvez appeler ces deux procédures à l'aide des options de menu du menu défini par l'utilisateur DocMenu1 :
5. Ajoutez le code VBA du tableau "VBA12".
6. Démarrez chaque procédure avec <F5>.

Les tableaux suivants affichent les codes VBA pour l'exemple :

Démarrer VBA avec le gestionnaire d'évènements (VBA10)

```
Option Explicit
'VBA10
'The next declaration has to be placed in the modul section
Dim WithEvents theApp As grafexe.Application

Private Sub SetApplication()
'This procedure has to be executed (with "F5") first
Set theApp = grafexe.Application
End Sub

Private Sub theApp_MenuItemClicked(ByVal MenuItem As IHMIMenuItem)
Dim objClicked As HMIMenuItem
Dim varMenuItemKey As Variant
Set objClicked = MenuItem
'
'"varMenuItemKey" contains the value of parameter "Key"
'from clicked menu-item
varMenuItemKey = objClicked.Key
Select Case varMenuItemKey
Case "mItem1_1"
MsgBox "The first menuitem was clicked!"
End Select
End Sub

Private Sub theApp_ToolbarItemClicked(ByVal ToolbarItem As IHMIToolbarItem)
Dim objClicked As HMIToolbarItem
Dim varToolbarItemKey As Variant
Set objClicked = ToolbarItem
'
'"varToolbarItemKey" contains the value of parameter "Key"
'from clicked toolbar-item
varToolbarItemKey = objClicked.Key
Select Case varToolbarItemKey
Case "tItem1_1"
MsgBox "The first symbol-icon was clicked!"
End Select
End Sub
```

Créer menu (VBA11 : Code-exemple pour fichier PDL)

```
Sub CreateDocumentMenusUsingMacroProperty()
'VBA11
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1",
"Doc_Menu_1")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1",
"First Menuitem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2",
"Second Menuitem")
```

```
'
'Assign a VBA-macro to every menu item
With ActiveDocument.CustomMenus("DocMenu1")
.MenuItems("dmItem1_1").Macro = "TestMacro1"
.MenuItems("dmItem1_2").Macro = "TestMacro2"
End With
End Sub
```

Créer menu (VBA821 : Code-exemple pour fichier PDT)

```
Sub CreateDocumentMenusUsingMacroProperty()
'VBA821
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Set objDocMenu = Application.CustomMenus.InsertMenu(1, "DocMenu1",
"Doc_Menu_1")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1",
"First Menuitem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2",
"Second Menuitem")
'
'Assign a VBA-macro to every menu item
With Application.CustomMenus("DocMenu1")
.MenuItems("dmItem1_1").Macro = "TestMacro1"
.MenuItems("dmItem1_2").Macro = "TestMacro2"
End With
End Sub
```

Macros pour options de menu personnalisées (VBA12)

```
Sub TestMacro1()
'VBA12
MsgBox "TestMacro1 was executed"
End Sub

Sub TestMacro2()
MsgBox "TestMacro2 was executed"
End Sub
```

Voir aussi

Propriété Macro (Page 2267)

Evénement ToolbarItemClicked (Page 1761)

Evénement MenuItemClicked (Page 1755)

Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)

Pour insérer une nouvelle option de menu (Page 1623)

Gestion des événements (Page 1704)

Exécution de macros VBA dans le Graphics Designer (Page 1612)

3.3.2.4 Accès à la bibliothèque des modules avec VBA

Accès à la bibliothèque des modules avec VBA

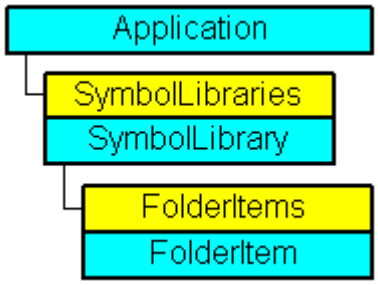
Introduction

La bibliothèque des modules renferme un vaste choix d'objets prédéfinis qui vous permettent de composer vos vues avec efficacité. La bibliothèque des modules se compose d'une bibliothèque générale et d'une bibliothèque du projet :

- La bibliothèque globale renferme des objets prédéfinis, fournis avec WinCC. Ces objets sont classés dans des dossiers thématiques, comme Vannes, Moteurs, Conduites, et bien d'autres.
- La bibliothèque du projet ne comprend aucun objet ni dossier au moment de la création d'un nouveau projet. Vous pouvez ranger les objets dont vous n'avez besoin que pour le projet considéré dans la bibliothèque du projet.

VBA vous permet d'accéder sans restriction à la bibliothèque des modules : Vous pouvez créer/supprimer un dossier et enregistrer des objets dans la bibliothèque des modules ou encore insérer une vue.

Accès à la bibliothèque des modules avec VBA



Dans VBA, la bibliothèque des modules est représentée par le listage "SymbolLibraries". Ce listage comprend deux éléments qui représentent la bibliothèque globale et la bibliothèque du

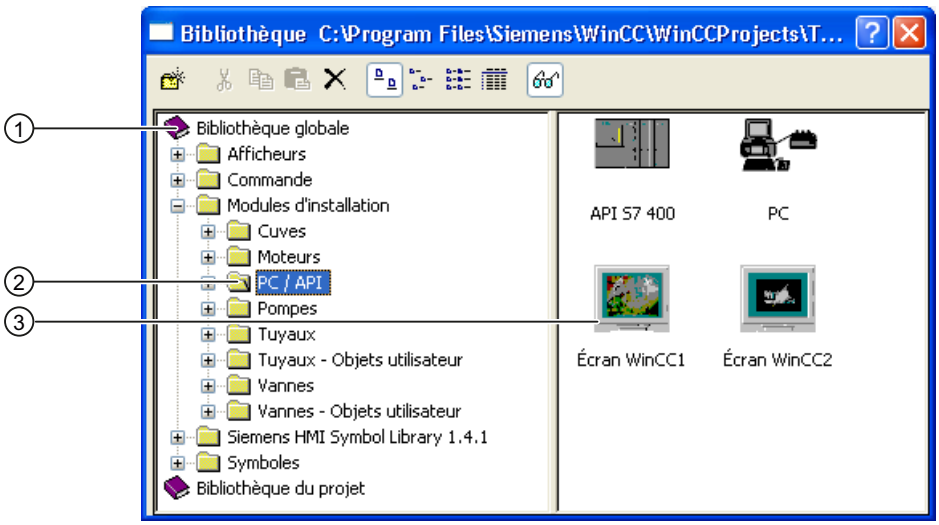
projet. Le listage "FolderItems" comprend des éléments qui peuvent représenter des dossiers ou des objets.

Remarque

Pour accéder à un objet du listage "SymbolLibraries", vous pouvez utiliser son numéro d'index ou son nom interne.

Pour connaître le nom interne d'un objet, dans la bibliothèque des modules, faites un clic droit sur l'objet et sélectionnez "Copier le chemin" dans le menu contextuel.

Le chemin d'accès à l'objet dans la bibliothèque des modules est ainsi copié dans le presse-papier.



Bibliothèque globale (1)

La bibliothèque globale est le premier élément du listage "SymbolLibraries", accessible par le numéro d'index 1. La bibliothèque du projet est accessible par le numéro d'index 2.

Accès à la bibliothèque globale avec VBA :

```
'VBA17
Application.SymbolLibraries(1)
```

Dossier (2)

Un dossier de la bibliothèque globale peut renfermer des sous-dossiers ou des objets thématiques. Dans VBA, un dossier correspond à l'objet FolderItem et est du type Folder. Les dossiers sont dans le listage "FolderItems". Avec VBA, vous pouvez créer un nouveau dossier, supprimer un dossier, ou ajouter un objet dans un dossier à partir du presse-papier.

Accès au dossier Modules de l'installation avec VBA :

```
'VBA18
```

```
Application.SymbolLibraries(1).FolderItems("Folder2")
```

Objet (3)

Dans VBA, un dossier correspond à l'objet FolderItem et est du type Folder. Les objets sont dans le listage Folder. Avec VBA, vous pouvez supprimer un objet ou le copier dans le presse-papier.

Accès à l'objet PC avec VBA :

```
'VBA19
Application.SymbolLibraries(1).FolderItems("Folder2").Folder("Folder2").Folder.Item("Objec
t1").DisplayName
```

Création ou suppression d'un dossier de la bibliothèque des modules

Les méthodes suivantes permettent de créer ou de supprimer un dossier :

- Methode "AddFolder(DefaultName)": Crée un nouveau dossier dans la bibliothèque des modules. Un nouveau dossier se voit attribué le nom interne FolderX, dans lequel X est le numéro d'ordre du dossier.
- Methode "Delete()": supprime un dossier existant (et tous les sous-dossiers et objets qu'il contient) de la bibliothèque des modules.

Insertion ou suppression d'objets dans la bibliothèque des modules

Vous pouvez copier des objets dans la bibliothèque des modules en interne (p. ex. de la bibliothèque globale à la bibliothèque du projet), y insérer un objet provenant d'une vue ou en effacer un objet :

- Méthodes "CopyToClipboard()" et "AddFromClipboard()": Copie un objet au sein de la bibliothèque de modules par le biais du presse-papiers.
- Methode "AddItem(DefaultName, pHMIObject)": Copie dans un dossier de la bibliothèque de modules un objet disponible dans la vue.
- Methode "Delete()": Supprime un objet.

Recherche d'objets ou de dossiers dans la bibliothèque des modules

Pour rechercher un objet ou un dossier, on utilise la méthode "FindByDisplayName("DisplayName")". L'intitulé de recherche dépend de la langue de travail configurée. La recherche s'arrête à la première occurrence de l'objet ou du dossier recherché.

Insertion d'un objet de la bibliothèque des modules dans une vue

Pour insérer un objet de la bibliothèque des modules dans la vue en cours, on utilise les méthodes CopyToClipboard() et PasteClipboard().

Voir aussi

Méthode CopyToClipboard (Page 1801)

Méthode PasteClipboard (Page 1843)

Méthode GetItemByPath (Page 1822)

Méthode FindByDisplayName (Page 1818)

Méthode Delete (Page 1807)

Méthode AddItem (Page 1782)

Méthode AddFromClipboard (Page 1779)

Méthode AddFolder (Page 1778)

Objet SymbolLibrary (Page 2023)

Objet SymbolLibraries (liste) (Page 2024)

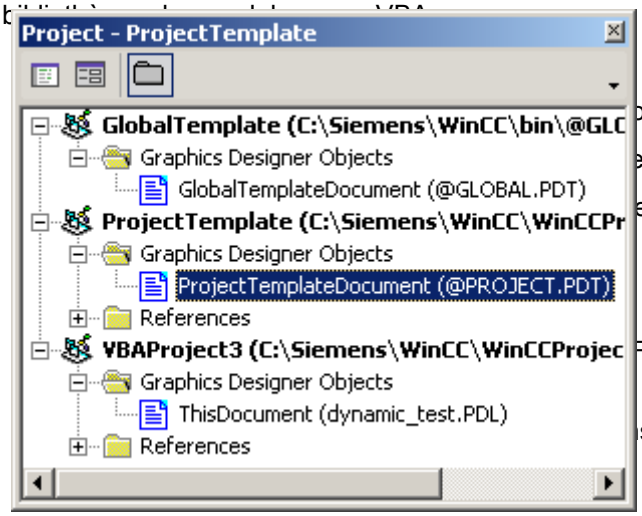
Pour insérer avec VBA dans une vue un objet provenant de la bibliothèque des modules (Page 1645)

Pour utiliser la bibliothèque des modules avec VBA (Page 1643)

Pour utiliser la bibliothèque des modules avec VBA

Introduction

Vous trouverez dans cette rubrique les instructions pour effectuer les tâches suivantes dans la b



- odules
- e la vue active
- es

Marche à suivre

1. F11> ou "Outils" > "Macros" > "Visual
2. s lequel vous voulez créer le code

3. Pour créer un dossier dans la bibliothèque des modules, insérez p. ex. la procédure AddNewFolderToProjectLibrary() dans le document. L'exemple qui suit est celui de la création du dossier Mon dossier :
```
Sub AddNewFolderToProjectLibrary()
'VBA20
Dim objProjectLib As HMISymbolLibrary
Set objProjectLib = Application.SymbolLibraries(2)
'
'("AddFolder(DefaultName)"-Methode):
objProjectLib.FolderItems.AddFolder ("Custom Folder")
End Sub
```
4. Pour copier un objet dans la bibliothèque de projet depuis la bibliothèque générale, insérez p. ex. la procédure CopyObjectFromGlobalLibraryToProjectLibrary() dans le document. Dans cet exemple, l'objet Object1 est copié :
```
Sub CopyObjectFromGlobalLibraryToProjectLibrary()
'VBA21
Dim objGlobalLib As HMISymbolLibrary
Dim objProjectLib As HMISymbolLibrary
Set objGlobalLib = Application.SymbolLibraries(1)
Set objProjectLib = Application.SymbolLibraries(2)
'
'Copies object "PC" from the "Global Library" into the clipboard
objGlobalLib.FolderItems("Folder2").Folder("Folder2").Folder.Item(
"Object1").CopyToClipboard
'
'The folder "Custom Folder" has to be available
objProjectLib.FolderItems("Folder1").Folder.AddFromClipBoard
("Copy of PC/PLC")
End Sub
```
5. Pour copier un objet dans la bibliothèque de projet depuis la vue active, insérez p. ex. la procédure AddObjectFromPictureToProjectLibrary() dans le document. Dans cet exemple, l'objet Circle1 est créé dans la vue active et copié ensuite dans le répertoire Folder1 :
```
Sub AddObjectFromPictureToProjectLibrary()
'VBA22
Dim objProjectLib As HMISymbolLibrary
Dim objCircle As HMICircle
Set objProjectLib = Application.SymbolLibraries(2)
'
'Insert new object "Circle1"
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1",
"HMICircle")
'
'The folder "Custom Folder" has to be available
'("AddItem(DefaultName, pHMIObject)"-Methode):
objProjectLib.FolderItems("Folder1").Folder.AddItem "ProjectLib
Circle", ActiveDocument.HMIObjects("Circle1")
End Sub
```

6. Pour supprimer un objet dans la bibliothèque des modules, insérez p. ex. la procédure DeleteObjectFromProjectLibrary() dans le document. L'exemple qui suit, le répertoire Folder1 créé préalablement est supprimé :

```
Sub DeleteObjectFromProjectLibrary()
'VBA23
Dim objProjectLib As HMISymbolLibrary
Set objProjectLib = Application.SymbolLibraries(2)
'
'The folder "Custom Folder" has to be available
'("Delete"-Methode):
objProjectLib.FolderItems("Folder1").Delete
End Sub
```

7. Démarrez chaque procédure avec <F5>.

Voir aussi

Objet SymbolLibrary (Page 2023)

Objet SymbolLibraries (liste) (Page 2024)

Méthode PasteClipboard (Page 1843)

Méthode Delete (Page 1807)

Méthode CopyToClipboard (Page 1801)

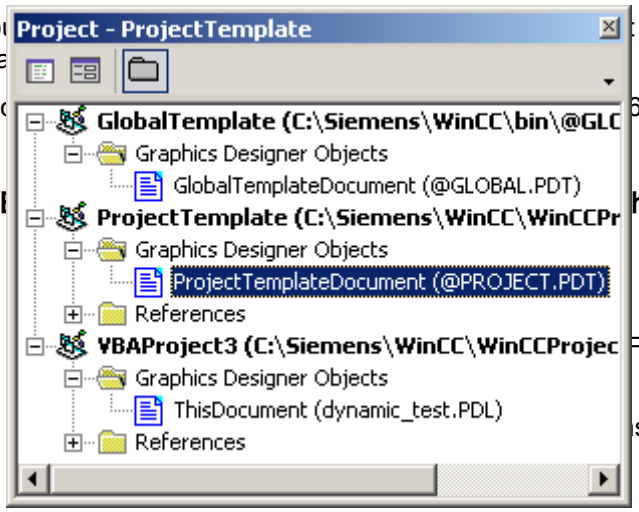
Méthode AddItem (Page 1782)

Méthode AddFromClipboard (Page 1779)

Méthode AddFolder (Page 1778)

Pou de la bibliothèque des modules (Pa

Acc 640)



Pour insérer avec VI hèque des modules

Marche à suivre

1. F11> ou "Outils" > "Macros" > "Visual

2. s lequel vous voulez créer le code

3. Pour insérer dans la vue active un objet provenant de la bibliothèque générale, insérez p. ex. la procédure CopyObjectFromGlobalLibraryToActiveDocument() dans le document. L'exemple qui suit est celui de l'insertion de l'objet Object1 dans la vue active :

```
Sub CopyObjectFromGlobalLibraryToActiveDocument()
'VBA24
Dim objGlobalLib As HMISymbolLibrary
Dim objHMIObject As HMIObject
Dim iLastObject As Integer
Set objGlobalLib = Application.SymbolLibraries(1)
'
'Copy object "PC" from "Global Library" to clipboard
objGlobalLib.FolderItems("Folder2").Folder("Folder2").Folder.Item(
"Object1").CopyToClipboard
'
'Get object from clipboard and add it to active document
ActiveDocument.PasteClipboard
'
'Get last inserted object
iLastObject = ActiveDocument.HMIObjects.Count
Set objHMIObject = ActiveDocument.HMIObjects(iLastObject)
'
'Set position of the object:
With objHMIObject
.Left = 40
.Top = 40
End With
End Sub
```

4. Démarrez la procédure avec <F5>.

Voir aussi

Méthode PasteClipboard (Page 1843)

Méthode CopyToClipboard (Page 1801)

Pour utiliser la bibliothèque des modules avec VBA (Page 1643)

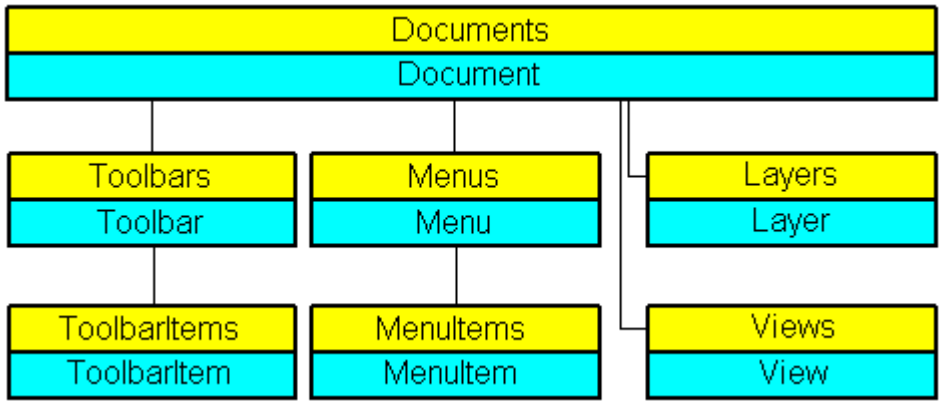
Accès à la bibliothèque des modules avec VBA (Page 1640)

3.3.3 Edition de vues avec VBA

3.3.3.1 Edition de vues avec VBA

Introduction

Les vues permettent de visualiser le process à commander et à observer. Elles montrent les principales étapes ou zones du process et représentent schématiquement le déroulement de la production. Dans VBA, une vue est représentée par l'objet Document.



Menus et barres d'outils spécifiques à une vue

Contrairement aux menus et aux barres d'outils spécifiques à une application, ceux spécifiques à une vue sont liés à la vue en question. Les menus et barres d'outils spécifiques à une vue demeurent visibles tant que la vue correspondante est active.

Vous utilisez les menus et les barres d'outils spécifiques à une vue lorsque vous ne souhaitez utiliser les macros appelées que dans la vue considérée.

Plans

Avec VBA, vous pouvez accéder aux niveaux dans le Graphics Designer. Un niveau est représenté par l'objet Layer. Les propriétés de l'objet Layer vous permettent notamment de donner un nom aux niveaux et de régler le zoom.

Le réglage de visibilité des niveaux RT s'effectue à l'aide de l'objet Document. Le réglage de visibilité des niveaux CS s'effectue à l'aide de l'objet Vue.

Copie d'une vue

Avec VBA, vous pouvez générer des copies d'une vue pour qu'elle figure dans plusieurs affichages différente. Dans VBA, la copie d'une vue est représentée par l'objet Vue.

Les propriétés de l'objet Vue vous permettent notamment de régler le facteur de zoom et de sélectionner le détail de vue à afficher.

Remarque

Si vous souhaitez exécuter avec WinCC V7.0 un code VBA dans une vue enregistrée dans WinCC V7.0 SP1, désactivez la "CCHMIDotNetObj 1.0 Type Library" dans l'éditeur VBA sous "Tools/References".

Le programme VBA est ensuite exécuté avec les fonctionnalités usuelles dans WinCC V7.0. Vous ne pouvez pas utiliser les nouvelles fonctions de WinCC V7.0 SP1.

Voir aussi

Edition d'une copie de vue avec VBA (Page 1651)

Pour créer des menus et des barres d'outils spécifiques à une vue (Page 1648)

Edition de niveaux avec VBA (Page 1650)

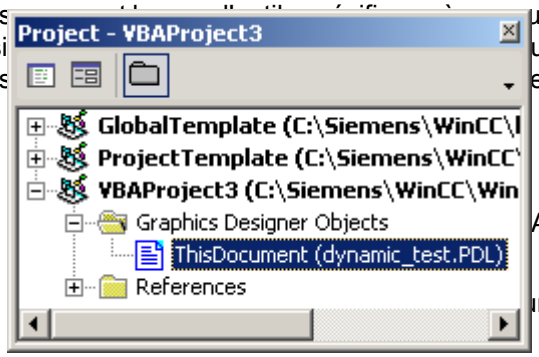
Edition d'objets avec VBA (Page 1653)

Adaptation du Graphics Designer avec VBA. (Page 1616)

3.3.3.2 Pour créer des menus et des barres d'outils spécifiques à une vue

Introduction

Les ue sont liés à la vue en question et demeurent visi us et les barres d'outils spécifiques à une vue lors ent que la vue en question.



Marche à suivre

1. A (<Alt+F11> ou "Outils" > "Macros" > "Visual
2. ument "ThisDocument" :

3. Pour créer un menu spécifique à une vue, insérez p. ex. la procédure CreateDocumentMenus() dans le document ThisDocument :
```
Sub CreateDocumentMenus()
'VBA25
'Declare menuobjects:
Dim objMenu1 As HMIMenu
Dim objMenu2 As HMIMenu
'Insert Menus ("InsertMenu"-Methode) with
'Parameters - "Position", "Key", "DefaultLabel":
Set objMenu1 = ActiveDocument.CustomMenus.InsertMenu(1,
"DocMenu1", "Doc_Menu_1")
Set objMenu2 = ActiveDocument.CustomMenus.InsertMenu(2,
"DocMenu2", "Doc_Menu_2")
End Sub
```
4. Pour créer une barre d'outils spécifique à une vue, insérez p. ex. la procédure CreateDocumentToolbars() dans le document ThisDocument :
```
Sub CreateDocumentToolbars()
'VBA26
'Declare required number of toolbarobjects:
Dim objToolbar1 As HMIToolbar
Dim objToolbar2 As HMIToolbar
'
'Insert toolbars ("Add"-Methode) with
'Parameter - "Key":
Set objToolbar1 = ActiveDocument.CustomToolbars.Add("DocToolbar1")
Set objToolbar2 = ActiveDocument.CustomToolbars.Add("DocToolbar2")
End Sub
```
5. Démarrez chaque procédure avec <F5>.

Voir aussi

Méthode Add (listage CustomToolbars) (Page 1768)

Méthode InsertMenu (Page 1825)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)

Pour créer des menus multilingues (Page 1626)

Pour insérer une nouvelle option de menu (Page 1623)

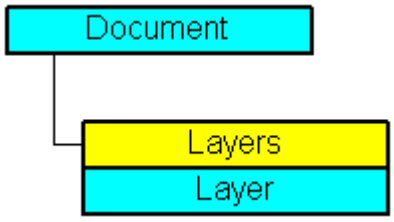
Création de menus et de barres d'outils personnalisés (Page 1620)

Configuration des menus et des barres d'outils (Page 1619)

3.3.3.3 Edition de niveaux avec VBA

Introduction

Dans le Graphics Designer, vous pouvez agencer des objets sur 32 niveaux. Pour pouvoir régler séparément la visibilité des niveaux de la vue (CS) et dans Runtime (RT), on fait une distinction entre les niveaux CS et les niveaux RT. Dans VBA, un niveau est représenté par l'objet Layer :



Dans le Graphics Designer, le niveau le plus bas est désigné par Niveau 0. Dans VBA, pour restituer le niveau le plus bas, utilisez l'indice 1 :

```
ActiveDocument.Layers(1)
```

Utilisation de l'objet Layer

L'objet Layer permet de définir le niveau minimum et le niveau maximum de zoom d'un objet et de lui donner un nom. Dans l'exemple qui suit, on définit les paramètres du niveau le plus bas de la vue active :

```
Sub ConfigureSettingsOfLayer
'VBA27
Dim objLayer As HMILayer
Set objLayer = ActiveDocument.Layers(1)
With objLayer
'Configure "Layer 0"
.MinZoom = 10
.MaxZoom = 100
.Name = "Configured with VBA"
End With
End Sub
```

Réglage de visibilité des niveaux CS et RT

Le réglage de visibilité des niveaux CS s'effectue à l'aide de l'objet Vue. Pour déterminer les niveaux à afficher ou à masquer dans Runtime, on utilise l'objet Document. Les méthode suivantes permettent de régler la visibilité des niveaux CS et RT :

- Méthode "IsCSLayerVisible(Index)" : vérifie si le niveau CS désigné est visible.
- Méthode "SetCSLayerVisible(Index, Val)" : Affiche ou masque le niveau CS indiqué.

Pour les niveaux IsRTLayerVisible, les méthodes SetRTLayerVisible et RT s'appliquent analogiquement.

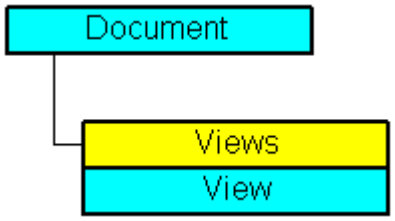
Voir aussi

Méthode IsRTLayerVisible (Page 1833)
Méthode SetRTLayerVisible (Page 1861)
Méthode SetCSLayerVisible (Page 1859)
Méthode IsCSLayerVisible (Page 1832)
Objet Layers (Liste) (Page 1956)
Edition de vues avec VBA (Page 1647)
Configuration multilingue avec VBA (Page 1617)

3.3.3.4 Edition d'une copie de vue avec VBA

Introduction

Avec VBA, vous pouvez générer des copies d'une vue pour qu'elle figure dans plusieurs affichages différente. Chaque affichage apparaît dans une fenêtre distincte. Dans VBA, la copie d'une vue est représentée par l'objet Vue.



Les propriétés de l'objet Vue vous permettent notamment de régler le facteur de zoom et de sélectionner le détail de vue à afficher.

Création d'une copie de vue

Pour créer une copie d'une vue, on utilise la méthode Add. L'exemple qui suit est celui de la création et de l'activation de la copie d'une vue active :

```
Sub CreateAndActivateView()
'VBA28
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
End Sub
```

Edition de la copie d'une vue :

Toute copie d'une vue peut être éditée comme suit :

- Paramétrer le facteur de zoom : Utilisez la propriété Zoom.
- Définition d'une détail de vue : Utilisez les propriétés ScrollPosX et ScrollPosY pour sélectionner un détail de vue à l'aide des barres de défilement de vue.
- Affichage et masquage des niveaux CS : Utilisez p. ex. la méthode SetCSLayerVisible(Index) pour afficher ou masquer le niveau désigné. La propriété ActiveLayer vous permet de sélectionner le niveau dont vous souhaitez éditer les objets.

L'exemple qui suit est celui de la création et de l'activation de la copie d'une vue active : Le facteur de zoom est réglé à 150% et la position de la vue est modifiée :

```
Sub SetZoomAndScrollPositionInActiveView()
'VBA29
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
With objView
.Activate
.ScrollPosX = 40
.ScrollPosY = 10
.Zoom = 150
End With
End Sub
```

Voir aussi

Méthode Add (listage Views) (Page 1773)

Propriété ScrollPosY (Page 2346)

Propriété ScrollPosX (Page 2345)

Propriété ActiveLayer (Page 2056)

Objet View (Page 2050)

Méthode SetCSLayerVisible (Page 1859)

Méthode IsCSLayerVisible (Page 1832)

Méthode Activate (Page 1765)

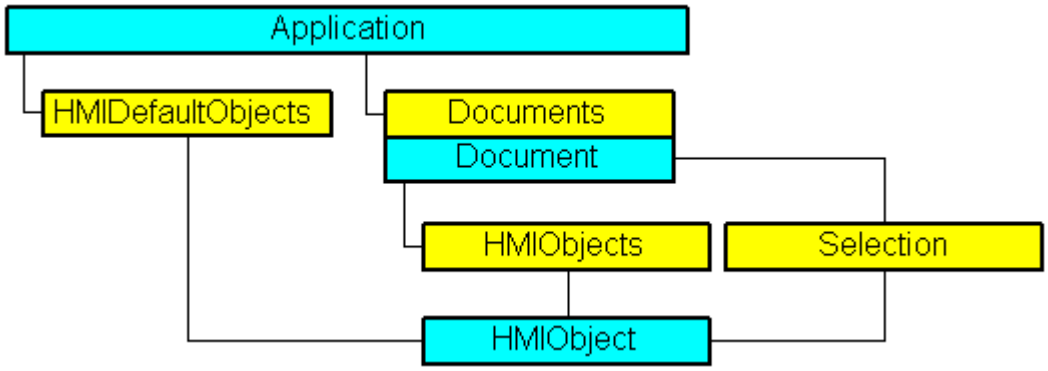
Edition de niveaux avec VBA (Page 1650)

Edition de vues avec VBA (Page 1647)

3.3.4 Edition d'objets avec VBA

3.3.4.1 Edition d'objets avec VBA

Accès aux objets dans le Graphics Designer



Dans VBA, tous les types d'objet de la vue en cours sont contenus dans le listage "HMIObjects". Ils ne sont pas classés selon le type d'objet (simple, complexe, Windows et contrôles) comme dans le Graphics Designer. C'est pourquoi vous pouvez, dans VBA, parcourir tous les objets d'une ou de plusieurs vues avec un seul curseur de défilement.

Une fois sélectionnés, les objets de la vue sont rassemblés dans le listage Selection. Si vous souhaitez modifier les précédents réglages de propriété d'un objet, utilisez le listage "HMIDefaultObjects".

Pour accéder à un objet d'une vue avec VBA, vous pouvez utiliser au choix le nom de l'objet, p. ex. "ActiveDocument.HMIObjects("Cercle1")", ou son numéro d'index. "ActiveDocument.HMIObjects(1)", par exemple, se rapport au premier objet de la vue active.

Edition d'objets avec VBA

Vous avez plusieurs possibilités d'édition des objets avec VBA :

- Création d'un objet dans une vue
- Suppression d'un objet existant
- Copie d'objets existants
- Groupage et dégroupage d'objets existants
- Recherche d'objets
- Affichage ou modification des propriétés d'un objet

Lorsque vous avez inséré un nouvel objet dans une vue avec VBA, cet objet se comporte comme si vous aviez double-cliqué dessus dans la palette des objets du Graphics Designer.

Cet objet reprend les valeurs de réglage de propriété existantes et apparaît dans le coin supérieur gauche de la vue.

L'accès aux propriétés de l'objet dépend de la manière dont vous l'avez créé. A cet égard, deux exemples :

Exemple 1 :

Dans cet exemple, on insère dans l'image en cours un cercle du type "HMIObject". Tout objet VBA du type "HMIObject" peut être utilisé dans le Graphics Designer. Toutefois, vous devez accéder explicitement à chaque caractéristique de l'objet considéré par l'intermédiaire de la propriété "Properties(Index)" :

```
Sub AddObject()
'VBA30
Dim objObject As HMIObject
Set objObject = ActiveDocument.HMIObjects.AddHMIObject("CircleAsHMIObject", "HMICircle")
'
'standard-properties (e.g. the position) are available every time:
objObject.Top = 40
objObject.Left = 40
'
'non-standard properties can be accessed using the Properties-collection:
objObject.Properties("FlashBackColor") = True
End Sub
```

Exemple 2 :

Dans cet exemple, on insère dans l'image en cours un cercle du type "HMICircle". Contrairement à l'exemple 1, vous ne pouvez utiliser l'objet "objCircle" que pour les objets du type "HMICircle" :

```
Sub AddCircle()
'VBA31
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("CircleAsHMICircle", "HMICircle")
'
'The same as in example 1, but here you can set/get direct the
'specific properties of the circle:
objCircle.Top = 80
objCircle.Left = 80
objCircle.FlashBackColor = True
End Sub
```

Voir aussi

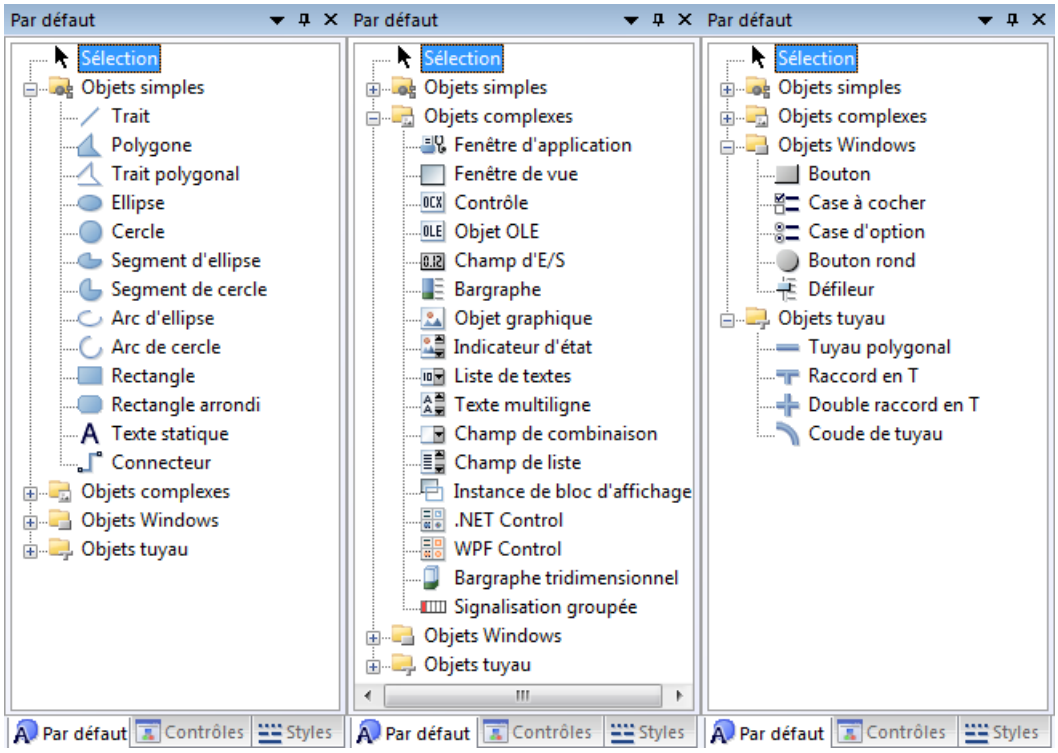
Objet LanguageFonts (liste) (Page 1951)
Référence VBA (Page 1724)
Propriété Underlined (Page 2388)
Propriété Size (Page 2353)
Propriété Parent (Page 2303)
Propriété LanguageID (Page 2219)
Propriété Italic (Page 2212)
Propriété FontFamily (Page 2189)
Propriété Bold (Page 2094)
Propriété Application (Page 2067)

3.3.4.2 Objets standard, complexes, Windows et tuyaux

Objets standard, complexes, Windows et tuyaux

Introduction

Les objets simples, complexes et Windows vous permettent de composer vos vues. Dans le Graphics Designer, ces objets se trouvent dans la palette d'objets à l'onglet "Standard" :



VBA vous permet d'accéder à ces objets dans toutes les vues de votre projet. Si, dans un projet comportant plusieurs vues, vous souhaitez modifier p. ex. la couleur de fond de tous les cercles, vous pouvez utiliser une macro VBA.

Insertion d'objets dans une vue

Utilisez la méthode "AddHMIObject(ObjectName, ProgID)" pour insérer un nouvel objet dans une vue. "ObjectName" étant le nom de l'objet (p. ex. "mon cercle") et "ProgID", la désignation d'objet VBA (p. ex. "HMICircle") :

```
Sub AddCircle()
'VBA32
'Creates object of type "HMICircle"
Dim objCircle As HMICircle
'
'Add object in active document
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("My Circle", "HMICircle")
End Sub
```

Edition d'objets

VBA vous permet d'accéder à toutes les propriétés d'objet que vous avez éditées par l'intermédiaire de la boîte de dialogue Propriétés de l'objet. Vous pouvez modifier et enregistrer les propriétés d'objet, ainsi que sélectionner des objets dans la vue. Si vous n'avez pas sélectionné d'objet dans la vue, vous pouvez utiliser les méthodes suivantes :

- Méthode "Find()" : Recherche un objet dans le listage "HMIObjects".
- Méthode "Delete()" : Supprime un objet HMIObject

Si vous avez sélectionné un objet, vous pouvez l'éditer avec, entre autres, les méthodes suivantes par l'intermédiaire du listage "Selection" :

- "AlignLeft()", "AlignRight()", "AlignTop()", "AlignBottom()" : Ces méthodes alignent les objets.
- "CreateGroup()", "CreateCustomizedObject()" : Ces méthodes créent un objet groupe ou un objet utilisateur.
- Méthode "DeselectAll()" : Supprime la sélection de tous les objets

Supprimer des références à des objets avec "Nothing"

Supprimez toujours les références utilisées pour les contrôles et objets standard ainsi que pour le document après avoir fermé le document. Mettez à cet effet les objets sur "Nothing". Voici un exemple de code pour un contrôle :

```
Public Sub DrawNewControl
   Const strFct = "CreatePdls"
   Dim objControl As HMIObject
   Dim objDoc As Document
   On Local Error GoTo errorhandler
   'open the document
```

```
   Set objDoc =
grafexe.Application.Documents.Open(grafexe.Application.ApplicationDa
taPath & "PDL1.pdl", hmiOpenDocumentTypeInvisible)
    'create new object
   Set objControl = objDoc.HMIObjects.AddActiveXControl("Control1",
"CCAxUserArchiveControl.AxUserArchiveControl.1")
   If objControl Is Nothing Then
      GoTo errorhandler
   End If
   'doing something with the control
   '......
      'delete reference to new control
   Set objControl = Nothing
   'saving PDL und deleting reference to it
   objDoc.Save
   objDoc.Close
   Set objDoc = Nothing
   Exit Sub
' errorhandler
errorhandler:
   If MsgBox("Error ocurred" & vbNewLine & "Yes - resume next" &
vbNewLine & "No  - stop script", vbOKCancel + vbCritical, strFct) =
vbOK Then
      Resume Next
   End If
End Sub
```

Voir aussi

Propriété Parent (Page 2303)

Propriété Item (Page 2213)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

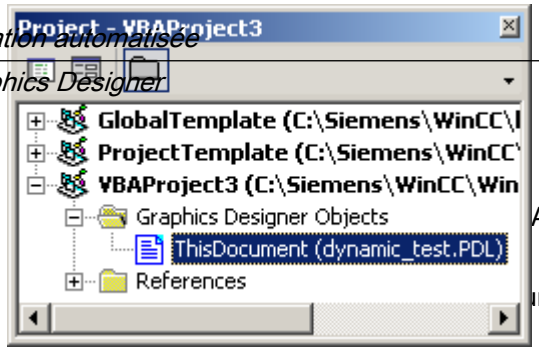
Vous éditez de la sorte les objets standard, complexes, Windows et tuyaux

Introduction

Vous trouverez ici les instructions suivantes relatives à l'édition d'objets simples, complexes et Windows :

- Propriétés d'un objet déterminé
- Propriétés d'un objet indéterminé
- Sélection d'un objet dans la vue active
- Recherche d'objets dans la vue active
- Suppression d'objets

Marche à suivre



1. A (<Alt+F11> ou "Outils" > "Macros" > "Visual
2. ument "ThisDocument" :
3. Pour définir les propriétés d'un type d'objet déterminé (p. ex. HMICircle), insérez p. ex. la procédure EditDefinedObjectType() dans le document "ThisDocument". L'exemple qui suit est celui de l'insertion d'un cercle dans l'image active, avec modification de son épaisseur et de sa couleur de trait :
```
Sub EditDefinedObjectType()
'VBA33
Dim objCircle As HMICircle
Set objCircle =
ActiveDocument.HMIObjects.AddHMIObject("myCircleAsCircle",
"HMICircle")
With objCircle
'direct calling of objectproperties available
.BorderWidth = 4
.BorderColor = RGB(255, 0, 255)
End With
End Sub
```
4. Pour modifier les propriétés d'un type d'objet indéterminé (HMIObject), insérez p. ex. la procédure EditHMIObject() dans le document "ThisDocument". L'exemple qui suit est celui de l'insertion d'un cercle dans l'image active, avec modification de son épaisseur et de sa couleur de trait :
```
Sub EditHMIObject()
'VBA34
Dim objObject As HMIObject
Set objObject =
ActiveDocument.HMIObjects.AddHMIObject("myCircleAsObject",
"HMICircle")
With objObject
'Access to objectproperties only with property "Properties":
.Properties("BorderWidth") = 4
.Properties("BorderColor") = RGB(255, 0, 0)
End With
End Sub
```
5. Pour sélectionner un objet dans la vue active, insérez p. ex. la procédure SelectObject() dans le document "ThisDocument". Dans l'exemple qui suit, on sélectionne un cercle dans la vue active :
```
Sub SelectObject()
'VBA35
Dim objObject As HMIObject
Set objObject =
ActiveDocument.HMIObjects.AddHMIObject("mySelectedCircle",
"HMICircle")
ActiveDocument.HMIObjects("mySelectedCircle").Selected = True
End Sub
```

6. Pour rechercher un objet dans la vue active, insérez p. ex. une procédure FindObjectsByName, ou FindObjectsByType() ou FindObjectsByProperty() dans le document "ThisDocument". Dans l'exemple qui suit, on recherche des objets dont le nom comprend la chaîne de caractères Circle :

```
Sub FindObjectsByName()
'VBA36
Dim colSearchResults As HMICollection
Dim objMember As HMIObject
Dim iResult As Integer
Dim strName As String
'
'Wildcards (?, *) are allowed
Set colSearchResults =
ActiveDocument.HMIObjects.Find(ObjectName:="*Circle*")
For Each objMember In colSearchResults
iResult = colSearchResults.Count
strName = objMember.ObjectName
MsgBox "Found: " & CStr(iResult) & vbCrLf & "Objectname: " &
strName
Next objMember
End Sub
```

Exemple de recherche d'objets du type HMICircle dans l'image active :

```
Sub FindObjectsByType()
'VBA37
Dim colSearchResults As HMICollection
Dim objMember As HMIObject
Dim iResult As Integer
Dim strName As String
Set colSearchResults =
ActiveDocument.HMIObjects.Find(ObjectType:="HMICircle")
For Each objMember In colSearchResults
iResult = colSearchResults.Count
strName = objMember.ObjectName
MsgBox "Found: " & CStr(iResult) & vbCrLf & "Objectname: " &
strName
Next objMember
End Sub
```

Exemple de recherche d'objets ayant la propriété "BackColor" dans l'image active :

```
Sub FindObjectsByProperty()
'VBA38
Dim colSearchResults As HMICollection
Dim objMember As HMIObject
Dim iResult As Integer
Dim strName As String
Set colSearchResults =
ActiveDocument.HMIObjects.Find(PropertyName:="BackColor")
For Each objMember In colSearchResults
iResult = colSearchResults.Count
strName = objMember.ObjectName
MsgBox "Found: " & CStr(iResult) & vbCrLf & "Objectname: " &
strName
Next objMember
```

```
End Sub
```

7. Pour supprimer un objet, insérez p. ex. la procédure DeleteObject() dans le document "ThisDocument". Dans l'exemple qui suit, on supprime le premier objet de la vue active.

```
Sub DeleteObject()
'VBA39
'Delete first object in active document:
ActiveDocument.HMIObjects(1).Delete
End Sub
```

8. Démarrez chaque procédure avec <F5>.

Voir aussi

Méthode Find (Page 1816)

Méthode Delete (Page 1807)

Méthode AddHMIObject (Page 1780)

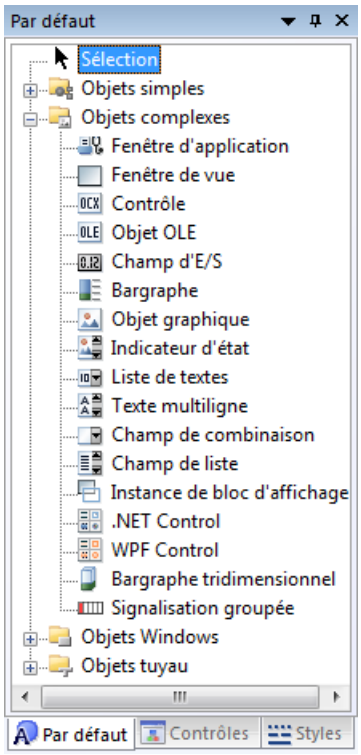
Vous éditez de la sorte les objets standard, complexes, Windows et tuyaux (Page 1657)

Edition d'objets avec VBA (Page 1653)

Objets OLE

Introduction

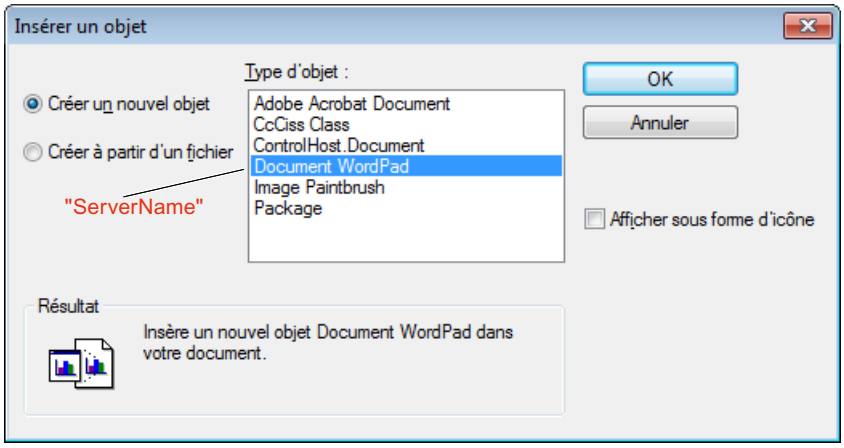
VBA vous permet d'insérer des objets OLE dans une vue. Les objets OLE appartiennent à la catégorie des objets Smart. Dans le Graphics Designer, ces objets se trouvent dans la palette d'objets à l'onglet Standard:



Insertion d'objets OLE dans une vue

Utilisez une méthode "AddOLEControl(ObjectName, ServerName, [CreationType], [UseSymbol])" pour insérer un objet OLE, "ObjectName" étant le nom de l'objet et "ServerName" celui de l'application devant être liée à l'objet OLE. Le paramètre ServerName correspond au Type d'objet dans la boîte de dialogue Insérer objet Les deux dernier

paramètres sont optionnel et représentent les possibilités de réglages dans la boîte de dialogue montrée :



Vous trouverez des informations supplémentaires sur les paramètres sous Méthode "AddOLEObject".

Dans l'exemple qui suit, un objet OLE contenant un document Wordpad est inséré dans la vue active :

```
Sub AddOLEObjectToActiveDocument()
'VBA40
Dim objOLEObject As HMIOLEObject
Set objOLEObject = ActiveDocument.HMIObjects.AddOLEObject("MS Wordpad Document1",
"Wordpad.Document.1")
End Sub
```

L'objet OLEObject est ajouté au listage "HMIObjects" en tant que dernier élément et hérite des propriétés de l'objet HMIObject.

Voir aussi

Objet OLEObject (Page 1975)

Méthode AddOLEObject (Page 1783)

Vous éditez de la sorte les objets standard, complexes, Windows et tuyaux (Page 1657)

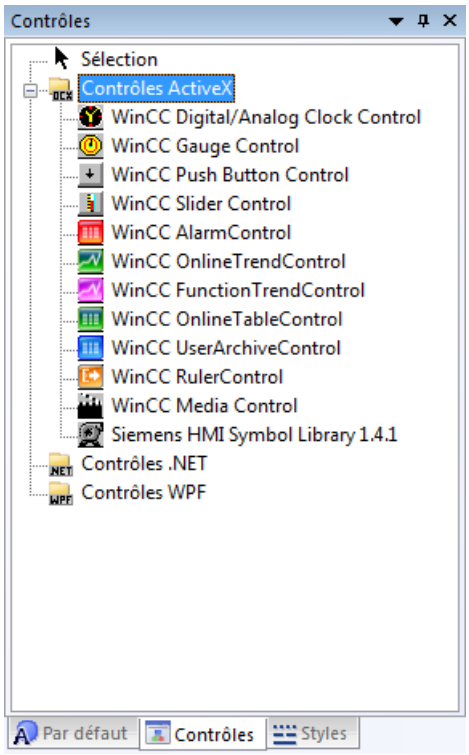
Objets standard, complexes, Windows et tuyaux (Page 1655)

Edition d'objets avec VBA (Page 1653)

ActiveX Controls

Introduction

VBA vous permet d'insérer des contrôles ActiveX dans une vue. Dans le Graphics Designer, vous trouverez les contrôles ActiveX fournis par WinCC dans la palette des objets à l'onglet Contrôles :



Vous trouverez des renseignements supplémentaires dans cette documentation à la rubrique Méthode "AddActiveXControl", ainsi que dans la documentation WinCC à la rubrique Palette des objets.

Insertion de contrôles ActiveX standard

Outre les contrôles ActiveX fournis par WinCC, vous pouvez insérer dans une vue tous les contrôles ActiveX standard enregistrés dans le système d'exploitation. Ceci vous permet d'utiliser dans vos vues des contrôles ActiveX que vous avez programmés. Vous trouverez une liste des contrôles ActiveX standard testés avec WinCC dans la documentation WinCC.

Insertion d'un ActiveX Control dans une vue

Pour insérer un nouvel ActiveX Control dans une vue, on utilise la méthode "AddActiveXControl(ObjectName, ProgID)", avec "ObjectName" étant le nom de l'ActiveX

Control (p. ex. "WinCC Gauge") et "ProgID", la désignation d'objet VBA (p. ex. "XGauge.XGauge.1") :

```
Sub AddActiveXControl()
'VBA41
Dim objActiveXControl As HMIActiveXControl
Set objActiveXControl = ActiveDocument.HMIObjects.AddActiveXControl("WinCC_Gauge",
"XGAUGE.XGaugeCtrl.1")
End Sub
```

L'objet ActiveXControl est inséré à la fin de la liste HMIObjects et hérite des propriétés de l'objet HMIObject.

Accès aux propriétés d'un ActiveX Control

Vous devez accéder aux propriétés d'objet de l'ActiveX Control par l'intermédiaire de la propriété Properties(Index). Le dialogue Propriétés de l'objet dans le Graphics Designer, ou le listage Properties, vous permet de connaître les propriétés de l'ActiveX Control :

```
Sub AddActiveXControl()
'VBA42
Dim objActiveXControl As HMIActiveXControl
Set objActiveXControl = ActiveDocument.HMIObjects.AddActiveXControl("WinCC_Gauge2",
"XGAUGE.XGaugeCtrl.1")
'
'move ActiveX-control:
objActiveXControl.Top = 40
objActiveXControl.Left = 60
'
'Change individual property:
objActiveXControl.Properties("BackColor").value = RGB(255, 0, 0)
End Sub
```

Accès limité aux graphiques d'arrière-plan de contrôles

Dans les contrôles suivants, le graphique du fond ne peut pas être configuré avec VBA :

Contrôle	Attribut
WinCC Digital/Analog Clock Control	Image d'arrière-plan
WinCC Gauge Control	Vue d'arrière-plan Graphique cadre
WinCC Push Button Control	PictureSelected PictureUnselected
WinCC Slider Control	Image d'arrière-plan Image curseur

Voir aussi

Objet ActiveXControl (Page 1873)

Méthode AddActiveXControl (Page 1774)

Vous éditez de la sorte les objets standard, complexes, Windows et tuyaux (Page 1657)

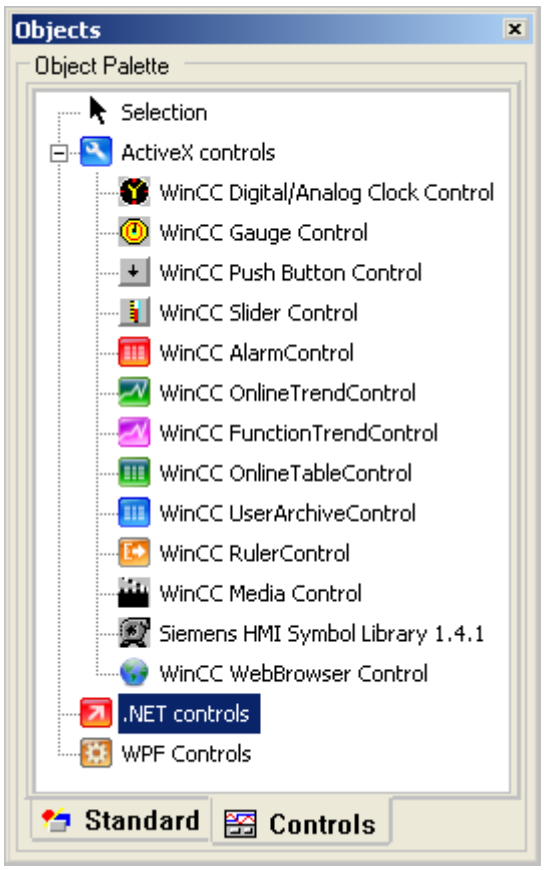
Objets standard, complexes, Windows et tuyaux (Page 1655)

Edition d'objets avec VBA (Page 1653)

Contols .Net

Introduction

VBA permet d'insérer des Controls .Net dans une vue. Dans le Graphics Designer, les Controls .Net se trouvent dans la palette d'objets à l'onglet "Standard" :



Vous trouverez des renseignements supplémentaires dans cette documentation à la rubrique "Méthode AddDotNetControl", ainsi que dans la documentation WinCC à la rubrique "Créer des vues de process > Utiliser des Controls > Controls .Net".

Insertion d'un Control .Net dans une vue

Utilisez la méthode "AddDotNetControl(ObjectName, ControlType, InGAC, AssemblyInfo)" pour insérer un nouveau Control .Net dans une vue, "ObjectName" remplace le nom du Control .Net. "ControlType" indique la plage de nom de l'objet. Si "InGAC" porte la valeur "TRUE", l'objet est enregistré dans Global Assembly Cache et les informations correspondantes sont mentionnées via "AssemblyInfo".

```
Sub AddDotNetControl()
'VBA851
Dim DotNetControl As HMIDotNetControl
Set DotNetControl = ActiveDocument.HMIObjects.AddDotNetControl("MyVBAControl",
"System.Windows.Forms.Label", True,"Assembly=System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089")
End Sub
```

L'objet Control .Net est inséré comme élément de la liste "HMIObjects" et hérite des propriétés de l'objet HMIObject.

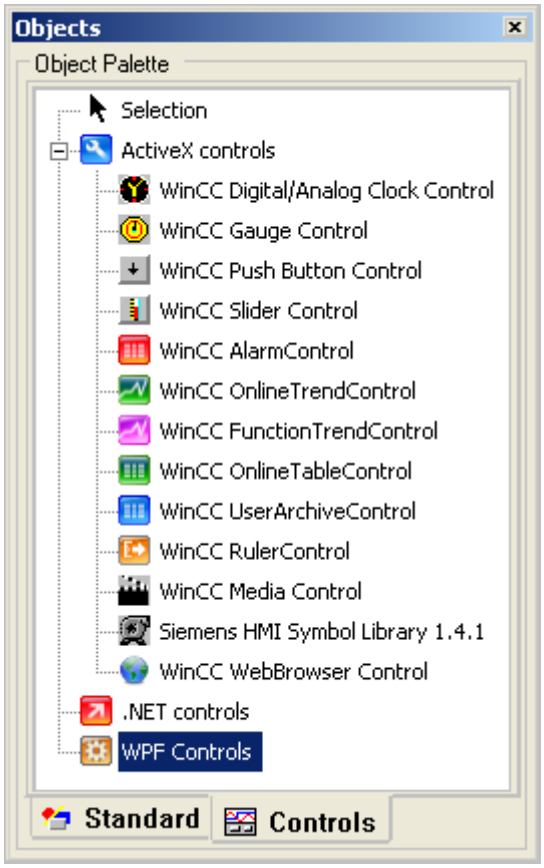
Accès aux propriétés d'un Control .Net

Le dialogue "Propriétés de l'objet > Propriétés Control" dans le Graphics Designer permet de connaître les propriétés du Control .Net.

Contrôles WPF

Introduction

VBA permet d'insérer des Controls .Net dans une vue. Dans le Graphics Designer, les Controls .Net se trouvent dans la palette d'objets à l'onglet "Standard" :



Vous trouverez des renseignements supplémentaires dans cette documentation à la rubrique "Méthode AddWPFControl", ainsi que dans la documentation WinCC à la rubrique "Créer des vues de process > Utiliser des Controls > Controls WPF".

Insertion d'un Control WPF dans une vue

Utilisez la méthode "AddWPFControl(ObjectName, ControlType, InGAC, AssemblyInfo)" pour insérer un nouveau Control WPF dans une vue, "ObjectName" remplace le nom du Control .Net. "ControlType" indique la plage de nom de l'objet. Si "InGAC" porte la valeur "TRUE", l'objet est enregistré dans Global Assembly Cache et les informations correspondantes sont mentionnées via "AssemblyInfo".

```
Sub AddWPFControl()
'VBA852
Dim WPFControl As HMIWPFControl
```

```
Set WPFControl = ActiveDocument.HMIObjects.AddWPFControl("MyWPFVBAControl",
"WinCCWPFControl.TestControl", False, "Assembly=Z:\TestControl\WinCCWPFControl.dll")
End Sub
```

L'objet ActiveXControl est inséré comme élément de la liste "HMIObjects" et hérite des propriétés de l'objet HMIObject.

Accès aux propriétés du Control WPF

Le dialogue "Propriétés de l'objet > Propriétés Control" dans le Graphics Designer permet de connaître les propriétés du Control WPF.

3.3.4.3 Objets Groupe

Objets Groupe

Introduction

Avec VBA, vous pouvez créer un objet Groupe à partir d'objets sélectionnés dans le Graphics Designer. Vous pouvez ajouter ou supprimer des objets d'un objet Groupe sans avoir à dissocier l'objet Groupe lui-même. Vous pouvez accéder sans restriction aux propriétés d'objet de chaque objet de l'objet Groupe. Vous pouvez dissocier à nouveau un objet Groupe ou le supprimer complètement.

Les objets suivants ne peuvent pas faire partie d'un objet Groupe :

- CustomizedObject (objet utilisateur)
- ActiveXControl
- OLEObject

Vous trouverez des renseignements supplémentaires sur les Objets groupe dans la documentation WinCC à la rubrique "Objet groupe".

Création d'un objet Groupe

Pour créer un objet Groupe, sélectionnez dans le Graphics Designer les objets devant en faire partie. Les objets sélectionnés sont alors contenus dans le listage Selection. Pour générer le groupe, on utilise la méthode "CreateGroup()" :

```
Sub CreateGroup()
'VBA43
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
```

```
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
objGroup.ObjectName = "myGroup"
End Sub
```

L'objet Groupe est ajouté au listage "HMIObjects" en tant que dernier élément. Les objets contenus dans le groupe conservent leur numéro d'index et sont en outre disponibles dans le listage "HMIObjects".

En outre, les objets de l'objet Groupe sont ajoutés au listage "GroupedHMIObjects", avec actualisation des numéros d'index.

Donnez un nom à l'objet Groupe (objGroup.Name = "My Group") pour vous permettre de l'identifier de manière univoque. Si vous ne lui donnez pas de nom, l'objet Groupe se voit attribué un nom spécifique par défaut (p. ex. Group1).

Un objet Groupe a les mêmes propriétés qu'un objet du type Object.

Edition d'un objet Groupe

Les possibilités d'édition d'un objet Groupe sont les suivantes :

- Méthode "Add(Index)" : Ajoute un nouvel objet à l'objet Groupe.
- Méthode "Remove(Index)" : Retire un objet de l'objet Groupe.
- Méthode "UnGroup()" : Dissocie l'objet groupe (annulation du groupage).
- Méthode "Delete()" : Supprime l'objet groupe et les objets qui le composent.

Edition d'objets d'un objet-groupe

Pour sélectionner un objet d'un objet Groupe, on utilise le listage "GroupedHMIObjects". Pour accéder à ses propriétés d'objet, vous devez accéder au nom de la propriété d'objet par l'intermédiaire de la propriété Properties, p. ex.:

```
Sub ModifyPropertyOfObjectInGroup()
'VBA44
Dim objGroup As HMIGroup
Set objGroup = ActiveDocument.HMIObjects("myGroup")
objGroup.GroupedHMIObjects(1).Properties("BorderColor") = RGB(255, 0, 0)
End Sub
```

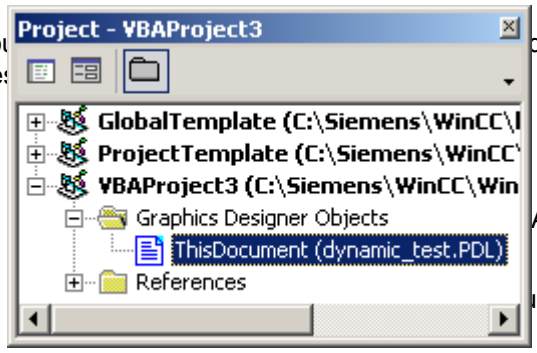
Voir aussi

Objet Selection (liste) (Page 2010)
Objet GroupedObjects (liste) (Page 1938)
Méthode Ungroup (Page 1866)
Méthode Remove (Page 1846)
Méthode Delete (Page 1807)
Méthode Add (listage GroupedObjects) (Page 1770)
Pour éditer les objets d'un objet Groupe avec VBA (Page 1673)
Pour éditer les objets Groupe avec VBA (Page 1670)
Edition d'objets avec VBA (Page 1653)
VBA dans le Graphics Designer (Page 1614)

Pour éditer les objets Groupe avec VBA

Condition

Vou deux objets graphiques dans le Graphics De



Marche à suivre

1. A (<Alt+F11> ou "Outils" > "Macros" > "Visual
2. ument "ThisDocument" :

3. Pour créer un objet Groupe à partir d'objets sélectionnés, insérez p. ex. la procédure CreateGroup() dans le document "ThisDocument" L'exemple qui suit est celui de la création de l'objet Groupe "My Group" :
```
Sub CreateGroup()
'VBA45
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle",
"HMICircle")
Set objRectangle =
ActiveDocument.HMIObjects.AddHMIObject("sRectangle",
"HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
'The name identifies the group-object
objGroup.ObjectName = "My Group"
End Sub
```
4. Pour ajouter un objet à l'objet Groupe "My Group", insérez p. ex. la procédure AddObjectToGroup() dans le document "ThisDocument". L'exemple qui suit est celui de l'insertion d'une ellipse dans l'objet Groupe "My Group" :
```
Sub AddObjectToGroup()
'VBA46
Dim objGroup As HMIGroup
Dim objEllipseSegment As HMIEllipseSegment
'Adds new object to active document
Set objEllipseSegment =
ActiveDocument.HMIObjects.AddHMIObject("EllipseSegment",
"HMIEllipseSegment")
Set objGroup = ActiveDocument.HMIObjects("My Group")
'Adds the object to the group
objGroup.GroupedHMIObjects.Add ("EllipseSegment")
End Sub
```

5. Pour supprimer un objet de l'objet Groupe, insérez p. ex. la procédure RemoveObjectFromGroup() dans le document "ThisDocument". Dans l'exemple qui suit, on supprime le premier objet de l'objet-groupe "My Group" :
```
Sub RemoveObjectFromGroup()
'VBA47
Dim objGroup As HMIGroup
Set objGroup = ActiveDocument.HMIObjects("My Group")
'delete first object of the group-object
objGroup.GroupedHMIObjects.Remove (1)
End Sub
```
6. Pour dissocier l'objet Groupe "My Group", insérez p. ex. la procédure UnGroup() dans le document "ThisDocument". L'exemple qui suit est celui de la dissociation de l'objet Groupe "My Group" :
```
Sub UnGroup()
'VBA48
Dim objGroup As HMIGroup
Set objGroup = ActiveDocument.HMIObjects("My Group")
objGroup.UnGroup
End Sub
```
7. Pour supprimer l'objet Groupe "My Group", insérez p. ex. la procédure DeleteGroup() dans le document "ThisDocument". L'exemple qui suit est l'objet Groupe "My Group" est supprimé avec l'ensemble des objets qui y sont contenus :
```
Sub DeleteGroup()
'VBA49
Dim objGroup As HMIGroup
Set objGroup = ActiveDocument.HMIObjects("My Group")
objGroup.Delete
End Sub
```
8. Démarrez chaque procédure avec <F5>.

Voir aussi

Méthode Ungroup (Page 1866)

Méthode Remove (Page 1846)

Méthode Delete (Page 1807)

Méthode CreateGroup (Page 1805)

Méthode Add (listage GroupedObjects) (Page 1770)

Objet Selection (liste) (Page 2010)

Objet GroupedObjects (liste) (Page 1938)

Objet Group (Page 1934)

Pour éditer les objets d'un objet Groupe avec VBA (Page 1673)

Objets Groupe (Page 1668)

Edition d'objets avec VBA (Page 1653)

VBA dans le Graphics Designer (Page 1614)

Pour éditer les objets d'un objet Groupe avec VBA

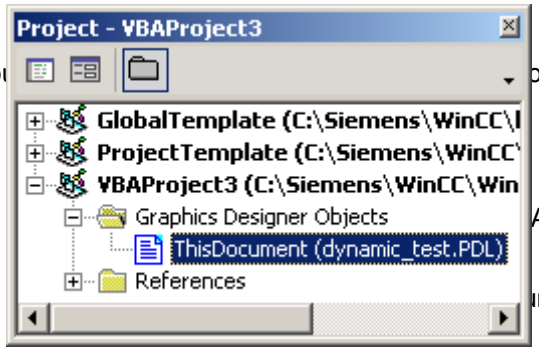
Introduction

Vous trouverez ici les instructions suivantes relatives à l'édition d'objets d'un objet Groupe avec VBA :

- Edition d'une propriété d'un objet faisant partie d'un objet Groupe
- Edition d'une propriété de tous les objets d'un objet groupe

Condition

Vou objets graphiques dans le Graphics Designer.



Marche à suivre

1. A (<Alt+F11> ou "Outils" > "Macros" > "Visual
2. ument "ThisDocument" :

3. Pour éditer une propriété d'un objet faisant partie d'un objet Groupe, insérez p. ex. la procédure ChangePropertiesOfGroupMembers() dans le document "ThisDocument". L'exemple qui suit est celui de la modification de trois objets différents faisant partie de l'objet Groupe "My Group" :

```
Sub ChangePropertiesOfGroupMembers()
'VBA50
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle",
"HMICircle")
Set objRectangle =
ActiveDocument.HMIObjects.AddHMIObject("sRectangle",
"HMIRectangle")
Set objEllipse =
ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
With objEllipse
.Top = 120
.Left = 120
.Selected = True
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
objGroup.ObjectName = "My Group"
'Set bordercolor of 1. object = "red":
objGroup.GroupedHMIObjects(1).Properties("BorderColor") =
RGB(255, 0, 0)
'set x-coordinate of 2. object = "120" :
objGroup.GroupedHMIObjects(2).Properties("Left") = 120
'set y-coordinate of 3. object = "90":
objGroup.GroupedHMIObjects(3).Properties("Top") = 90
End Sub
```

4. Pour modifier les propriétés de tous les objets Groupe, insérez p. ex. la procédure ChangePropertiesOfAllGroupMembers() dans le document "ThisDocument". L'exemple qui suit est celui de la modification de la propriété "BorderColor" de chacun des objets de l'objet Groupe "My Group" : Pour que cet exemple fonctionne, vous devez avoir créé l'objet Groupe "My Group" :
```
Sub ChangePropertiesOfAllGroupMembers()
'VBA51
Dim objGroup As HMIGroup
Dim iMaxMembers As Integer
Dim iIndex As Integer
Set objGroup = ActiveDocument.HMIObjects("My Group")
iIndex = 1
'
'Get number of objects in group-object:
iMaxMembers = objGroup.GroupedHMIObjects.Count
'
'set linecolor of all objects = "yellow":
For iIndex = 1 To iMaxMembers
objGroup.GroupedHMIObjects(iIndex).Properties("BorderColor") =
RGB(255, 255, 0)
Next iIndex
End Sub
```
5. Démarrez chaque procédure avec <F5>.

Voir aussi

Objet Properties (liste) (Page 1992)

Objet GroupedObjects (liste) (Page 1938)

Méthode Ungroup (Page 1866)

Méthode Remove (Page 1846)

Méthode Delete (Page 1807)

Méthode Add (listage GroupedObjects) (Page 1770)

Pour éditer les objets Groupe avec VBA (Page 1670)

Objets Groupe (Page 1668)

Edition d'objets avec VBA (Page 1653)

VBA dans le Graphics Designer (Page 1614)

3.3.4.4 Objets utilisateur

Objets utilisateur

Introduction

Avec VBA, vous pouvez créer un objet utilisateur à partir d'objets sélectionnés dans le Graphics Designer. Contrairement aux objets Groupe, les seules propriétés d'objet d'un objet utilisateur qui sont disponibles sont celles que vous avez sélectionnées dans son dialogue de configuration. Un objet utilisateur ne peut pas être configuré avec VBA.

Vous trouverez des renseignements supplémentaires sur les objets utilisateur dans la documentation WinCC à la rubrique "Objet utilisateur".

Création d'un objet utilisateur avec VBA

Pour créer un objet utilisateur à partir d'objets sélectionnés, on utilise la méthode "CreateCustomizedObject()" :

```
Sub CreateCustomizedObject()
'VBA52
Dim objCustomizedObject As HMICustomizedObject
Set objCustomizedObject = ActiveDocument.Selection.CreateCustomizedObject
objCustomizedObject.ObjectName = "My Customized Object"
End Sub
```

Lorsque vous utilisez la méthode "CreateCustomizedObject()", le Dialogue de configuration s'affiche pour vous permettre de sélectionner les propriétés de l'objet. L'objet utilisateur ainsi créé est ajouté au listage "HMIObjects". Donnez à l'objet utilisateur un nom caractéristique (objCustomizedObject.Name My Customized Object) pour vous permettre de l'identifier sans ambiguïté.

Remarque

Si vous avez ouvert un document en mode non visible, n'y créez aucun objet utilisateur par script VBA. L'exécution du programme est interrompue par un dialogue de configuration.

Edition d'un objet utilisateur

Les possibilités d'édition d'un objet utilisateur sont les suivantes :

- Méthode "Destroy" : Dissocie un objet utilisateur.
- Méthode "Delete" : Supprime l'objet utilisateur et les objets qui le composent.

Edition d'objets composant un objet utilisateur

Pour accéder aux propriétés d'objet sélectionnées des objets qui composent un objet utilisateur, on utilise le listage Properties :

```
Sub EditCustomizedObjectProperty()
'VBA53
Dim objCustomizedObject As HMICustomizedObject
Set objCustomizedObject = ActiveDocument.HMIObjects(1)
objCustomizedObject.Properties("BackColor") = RGB(255, 0, 0)
End Sub
```

Si vous avez sélectionné plusieurs propriétés identiques (p. ex. la couleur de fond d'un cercle et d'un carré), ces propriétés sont numérotées (BackColor et BackColor1).

Voir aussi

Objet HMIObject (Page 1943)

Objet CustomizedObject (Page 1900)

Méthode Destroy (Page 1811)

Méthode Delete (Page 1807)

Méthode CreateCustomizedObject (Page 1802)

Pour éditer un objet utilisateur avec VBA (Page 1677)

Pour éditer les objets Groupe avec VBA (Page 1670)

Objets Groupe (Page 1668)

Edition d'objets avec VBA (Page 1653)

Pour éditer un objet utilisateur avec VBA

Introduction

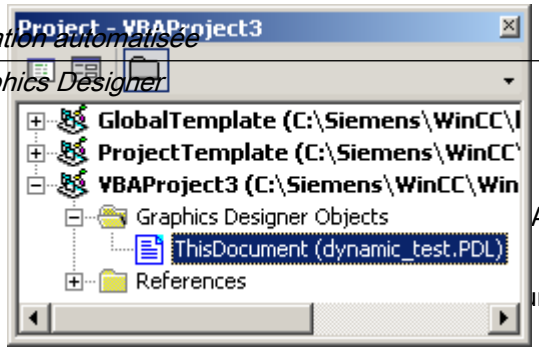
Vous trouverez dans cette rubrique les instructions suivantes relatives à l'édition d'un objet utilisateur avec VBA :

- Création d'un objet utilisateur à partir d'objets sélectionnés
- Dissociation d'un objet utilisateur
- Suppression d'un objet utilisateur

Remarque

Un objet utilisateur ne peut pas être configuré avec VBA.

Marche à suivre



1. A (<Alt+F11> ou "Outils" > "Macros" > "Visual
2. ument "ThisDocument" :
3. Pour créer un objet utilisateur à partir d'objets sélectionnés, insérez p. ex. la procédure "CreateCustomizedObject()" dans le document "ThisDocument". L'exemple qui suit est celui de la création de l'objet utilisateur "My Customized Object" :
```
Sub CreateCustomizedObject()
'VBA54
Dim objCustomizedObject As HMICustomizedObject
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle",
"HMICircle")
Set objRectangle =
ActiveDocument.HMIObjects.AddHMIObject("sRectangle",
"HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
Set objCustomizedObject =
ActiveDocument.Selection.CreateCustomizedObject
'
'*** The "Configurationdialog" started. ***
'*** Configure the costumize-object with the "configurationdialog"
***
'
objCustomizedObject.ObjectName = "My Customized Object"
End Sub
```
4. Pour dissocier un objet utilisateur, insérez p. ex. la procédure "DestroyCustomizedObject()" dans le document "ThisDocument". L'exemple qui suit est celui de la dissociation de l'objet utilisateur "My Customized Object" précédemment créé :
```
Sub DestroyCustomizedObject()
'VBA55
Dim objCustomizedObject As HMICustomizedObject
Set objCustomizedObject = ActiveDocument.HMIObjects("My
Customized Object")
objCustomizedObject.Destroy
End Sub
```

5. Pour supprimer un objet utilisateur, insérez p. ex. la procédure "DeleteCustomizedObject()" dans le document "ThisDocument". L'exemple qui suit est celui de la suppression de l'objet utilisateur "My Customized Object" précédemment créé :
```
Sub DeleteCustomizedObject()
'VBA56
Dim objCustomizedObject As HMICustomizedObject
Set objCustomizedObject = ActiveDocument.HMIObjects("My
Customized Object")
objCustomizedObject.Delete
End Sub
```
6. Démarrez chaque procédure avec <F5>.

Voir aussi

Méthode Destroy (Page 1811)

Méthode Delete (Page 1807)

Méthode CreateCustomizedObject (Page 1802)

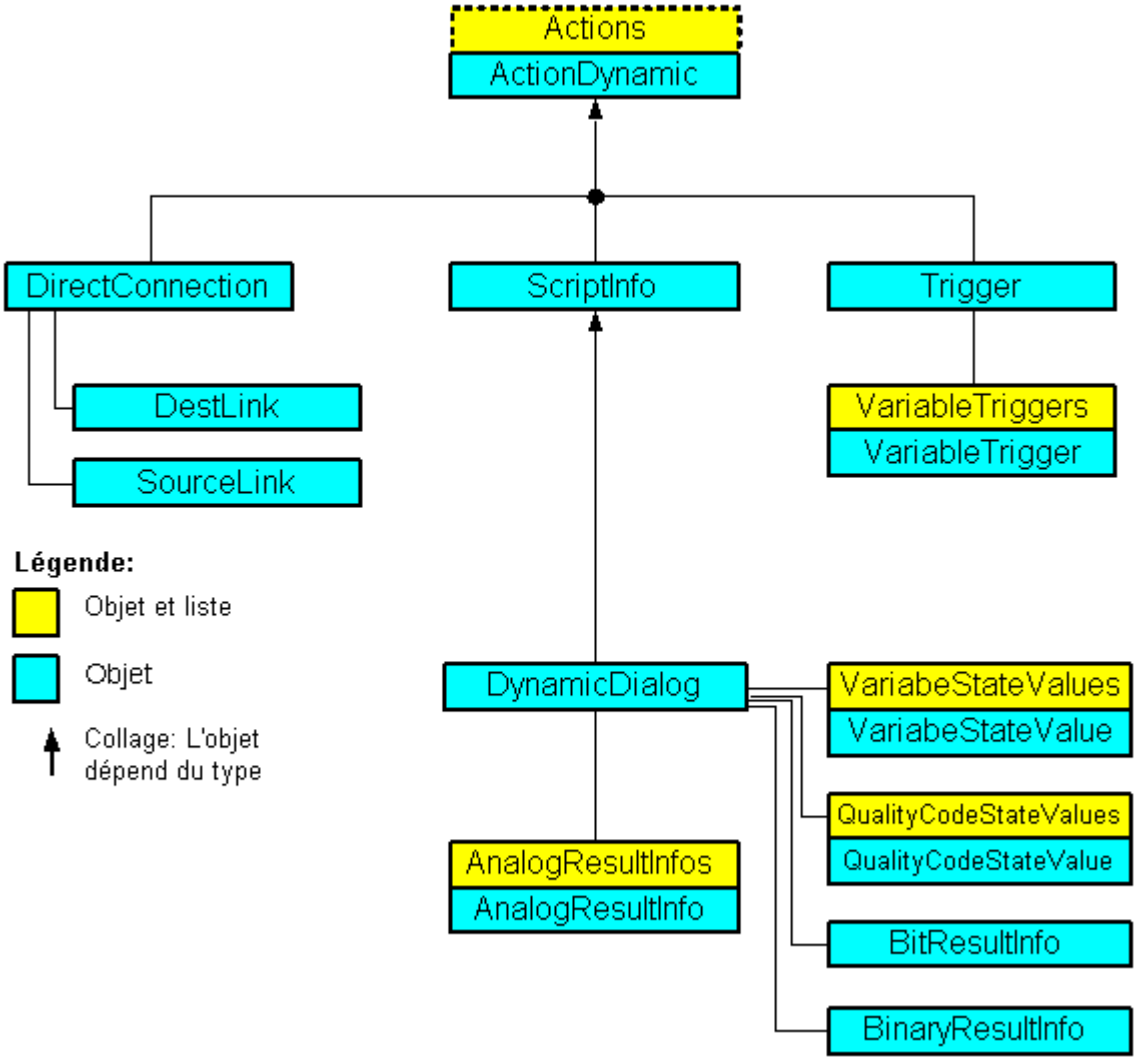
Objets utilisateur (Page 1676)

3.3.5 Création de dynamisations avec VBA

3.3.5.1 Création de dynamisations avec VBA

Introduction

VBA vous permet de dynamiser les propriétés des vues et des objets et de définir des actions commandées par un événement. A cet effet, vous disposez dans VBA de l'objet ActionDynamic :



L'objet ActionDynamic représente une interface qui dépend du type d'objet :

- Lorsque vous associez une dynamique à une propriété (objet Property), l'objet ActionDynamic-Objekt hérite des caractéristiques des objets ScriptInfo, Trigger et DynamicDialog.
- Lorsque vous configurez une action commandée par un événement (objet Event), l'objet ActionDynamic-Objekt hérite des caractéristiques des objets ScriptInfo et DirectConnection.

Dynamisation des propriétés d'une vue ou d'un objet

VBA vous permet de dynamiser les propriétés de vues et d'objets. Pour dynamiser, vous pouvez utiliser des variables, des scripts ou le dialogue Dynamique. La dynamisation vous permet, p. ex. de configurer le changement de couleur d'un objet au changement de valeur d'une variable.

Configuration d'actions pilotées par un événement

VBA vous permet de configurer des actions commandées par un événement. L'action (script ou liaison directe) est déclenchée à l'occurrence dans Runtime de l'événement défini. Un événement peut être, p. ex., la modification d'une propriété d'objet ou un clic de souris sur un bouton.

Edition de déclencheurs

Vous pouvez éditer des déclencheurs avec VBA. Les déclencheurs sont nécessaires à la dynamisation. Ils déterminent l'actualisation dans Runtime d'une valeur dynamisée, cette actualisation pouvant intervenir, p. ex., à intervalles réguliers ou au moment d'un changement de vue.

Dans la configuration d'actions commandées par événement, l'événement représente le déclencheur.

Voir aussi

Edition de déclencheurs (Page 1701)

Configuration d'actions commandées par événement avec VBA (Page 1693)

Dynamisation des propriétés d'une vue ou d'un objet (Page 1682)

3.3.5.2 Dynamisation des propriétés d'une vue ou d'un objet

Dynamisation des propriétés d'une vue ou d'un objet

Introduction

VBA vous permet de dynamiser les propriétés de vues et d'objets. Dans Runtime, les propriétés d'objet dynamisées peuvent être modifiées, p. ex. en fonction de la valeur d'une variable. Les méthodes de dynamisation disponibles sont les suivantes :

- Introduction de variable
- Dialogue Dynamique
- Scripts

Principe

L'exemple qui suit illustre le principe de marche à suivre pour dynamiser une propriété d'objet :

```
Sub CreateDynamicOnProperty()
'VBA57
Dim objVariableTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1", "HMICircle")
'
'Create dynamic with type "direct Variableconnection" at the
'property "Radius":
Set objVariableTrigger =
objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVariableDirect, "'NewDynamic1'")
'
'To complete dynamic, e.g. define cycle:
With objVariableTrigger
.CycleType = hmiVariableCycleType_2s
End With
End Sub
```

Remarque

Notez que la définition du nom de la variable ne suffit pas à créer la variable. A cet effet, il faut utiliser le dialogue de sélection de variable.

Introduction de variable

Pour dynamiser une propriété par liaison directe ou indirecte à une variable, on utilise l'objet VariableTrigger. La propriété ainsi dynamisée réagit dans Runtime à la modification des variables définies. A cet effet, on définit dans VBA le nom de la variable (propriété VarName) et le temps de cycle (propriété CycleTime).

Dialogue Dynamique

Pour dynamiser une propriété à l'aide du dialogue Dynamique, on utilise l'objet DynamicDialog. La propriété ainsi dynamisée réagit dans Runtime en fonction de la plage de valeurs de la valeur d'une variable. Pour définir cette plage de valeurs, on dispose des objets suivants :

- Objet AnalogResultInfos : On utilise cet objet pour définir une valeur constante sur les plages de valeurs d'une variable ou d'un script. Cette valeur constante sera affectée à la propriété dynamisée lorsque la valeur retournée de la variable ou du script se trouvera dans la plage définie.
- Objet BinaryResultInfo : On utilise cet objet pour définir une valeur constante sur les plages de valeurs binaires (zéro et une valeur différente de zéro) d'une variable ou d'un script. Cette valeur constante sera affectée à la propriété dynamisée lorsque la valeur retournée de la variable ou du script sera égale à l'une des deux valeurs de la plage.
- Objet VariableStateValue : On utilise cet objet pour affecter une valeur constante à un état (p. ex. "Limite supérieure dépassée") d'une variable définie. Cette valeur constante sera attribuée à la propriété dynamisée à l'occurrence de l'état en question.

Scripts

Pour dynamiser une propriété par un script en C ou ScriptInfo, on utilise l'objet VB. La propriété ainsi dynamisée réagira dans Runtime conformément au script et sera commandée par un déclencheur. Pour paramétrer le déclencheur, on utilise l'objet Trigger.

Voir aussi

Objet VariableTrigger (Page 2048)

Objet VariableStateValue (Page 2045)

Objet Trigger (Page 2035)

Objet ScriptInfo (Page 2009)

Objet BinaryResultInfo (Page 1884)

Objet AnalogResultInfos (liste) (Page 1875)

Pour dynamiser une propriété par un script VB (Page 1691)

Pour dynamiser une propriété par un script en C (Page 1688)

Pour dynamiser une propriété par le dialogue Dynamique (Page 1685)

Pour dynamiser une propriété par liaison à une variable (Page 1684)

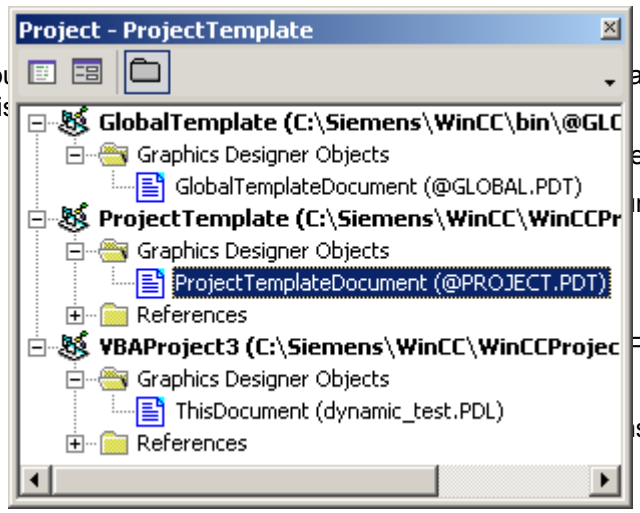
Création de dynamisations avec VBA (Page 1680)

Pour dynamiser une propriété par liaison à une variable

Introduction

Vou a dynamisation d'une propriété par liais

- e variable
- ne variable



Marche à suivre

1. F11> ou "Outils" > "Macros" > "Visual
2. s lequel vous voulez créer le code
3. Pour dynamiser une propriété d'objet avec une liaison directe à une variable, insérez p. ex. une procédure "AddDynamicAsVariableDirectToProperty()" dans le document. Dans cet exemple, on dynamise la propriété "Top" d'un cercle par la variable "Otto" :

```
Sub AddDynamicAsVariableDirectToProperty()
'VBA58
Dim objVariableTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1",
"HMICircle")
'Create dynamic at property "Top"
Set objVariableTrigger =
objCircle.Top.CreateDynamic(hmiDynamicCreationTypeVariableDirect,
"Otto")
'
'define cycle-time
With objVariableTrigger
.CycleType = hmiVariableCycleType_2s
End With
End Sub
```

4. Pour dynamiser une propriété d'objet avec une liaison indirecte à une variable, insérez p. ex. la procédure "AddDynamicAsVariableIndirectToProperty()" dans le document. L'exemple qui suit est celui de la dynamisation de la propriété "Left" d'un cercle par la variable "Anton" :
```
Sub AddDynamicAsVariableIndirectToProperty()
'VBA59
Dim objVariableTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle2",
"HMICircle")
'Create dynamic on property "Left":
Set objVariableTrigger =
objCircle.Left.CreateDynamic(hmiDynamicCreationTypeVariableIndirec
t, "Anton")
'
'Define cycle-time
With objVariableTrigger
.CycleType = hmiVariableCycleType_2s
End With
End Sub
```
5. Démarrez chaque procédure avec <F5>.

Voir aussi

Propriété CycleType (Page 2148)

Propriété VarName (Page 2467)

Objet VariableTrigger (Page 2048)

Méthode CreateDynamic (Page 1803)

Pour dynamiser une propriété par un script VB (Page 1691)

Pour dynamiser une propriété par un script en C (Page 1688)

Pour dynamiser une propriété par le dialogue Dynamique (Page 1685)

Dynamisation des propriétés d'une vue ou d'un objet (Page 1682)

Création de dynamisations avec VBA (Page 1680)

Pour dynamiser une propriété par le dialogue Dynamique

Introduction

Le dialogue Dynamique vous permet de dynamiser les propriétés des vues et des objets en fonction de plages de valeurs ou de l'état de variables déterminés. A cet effet, on dispose des options de plages de valeurs suivantes :

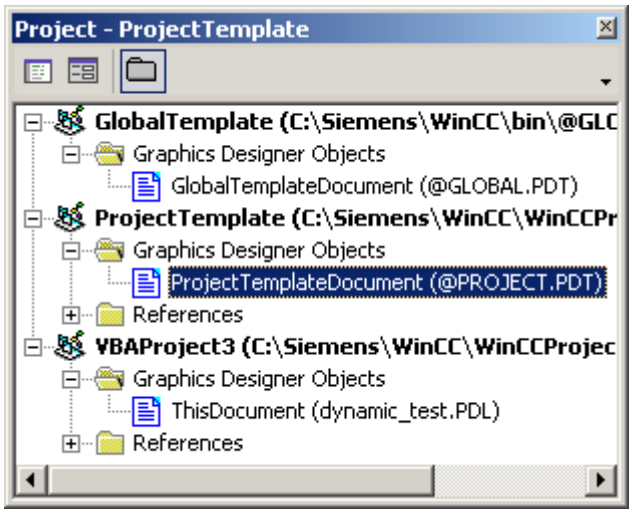
- Analogique
- Binaire

- Bit
- Direct

Dans VBA, on définit le type de plage de valeurs à l'aide de la propriété ResultType. On traitera ici de la dynamisation d'une propriété d'objet dont la plage de valeurs est analogique. Vous trouverez les autres renseignements sur la dynamisation par le dialogue Dynamique à la rubrique "Objet dialogue Dynamique" sous la référence VBA de cette documentation.

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor").
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



3. Pour dynamiser une propriété d'objet par le dialogue Dynamique, insérez p. ex. la procédure "AddDynamicDialogToCircleRadiusTypeAnalog()" dans le document. Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on crée trois plages de valeurs analogiques.
4. Démarrez la procédure avec <F5>.

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA60
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
'
'Create dynamic
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog, "'NewDynamic1'")
'
'Configure dynamic. "ResultType" defines the type of valuerange:
With objDynDialog
```

```
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.Add 50, 40
.AnalogResultInfos.Add 100, 80
.AnalogResultInfos.ElseCase = 100
End With
End Sub
```

Nouvelle méthode VBA pour la dynamisation avec le dialogue Dynamique

Une nouvelle méthode a été ajoutée pour des raisons d'optimisation.

- CreateDynamicDialog([Code as String], iResultType as Long) as HMIActionDynamic

Le paramètre "IResultType" a les constantes suivantes :

- hmiResultTypeDirect = 0
- hmiResultTypeAnalog= 1
- hmiResultTypeBool = 2
- hmiResultTypeBit = 3

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. Le dialogue Dynamique se voit attribuer un nom de variable et un "ResultType".

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA820
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
'Create Object
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("myCircle","HMICircle")
'Create dynamic (Tag "myTest" must be exist")
Set objDynDialog = objCircle.Radius.CreateDynamicDialog("'myTest'",1)
End Sub
```

Initialiser la propriété String

Avant la dynamisation, il vous faut initialiser la propriété "String" en lui attrituant un texte. Dans l'exemple ToolTip suivant, ceci s'effectue dans "objCircle.ToolTipText = "Text".

```
Sub Dyn()
'VBA823
Dim objCircle As HMICircle
Dim doc As Document
Dim objDynDialog As HMIDynamicDialog
Set doc = ActiveDocument
```

```
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle","HMICircle")
objCircle.ObjectName = "Circle1"
objCircle.BorderColor = RGB(255, 0, 0)
objCircle.BackColor = RGB(0, 255, 0)
objCircle.ToolTipText = "Text"
Set objDynDialog = objCircle.ToolTipText.CreateDynamic(hmiDynamicCreationTypeDynamicDialog, "'Var'")
End Sub
```

Voir aussi

Pour dynamiser une propriété par liaison à une variable (Page 1684)

Propriété ResultType (Page 2335)

Objet DynamicDialog (Page 1912)

Méthode CreateDynamic (Page 1803)

Pour dynamiser une propriété par un script VB (Page 1691)

Pour dynamiser une propriété par un script en C (Page 1688)

Dynamisation des propriétés d'une vue ou d'un objet (Page 1682)

Création de dynamisations avec VBA (Page 1680)

Propriété TooltipText (Page 2373)

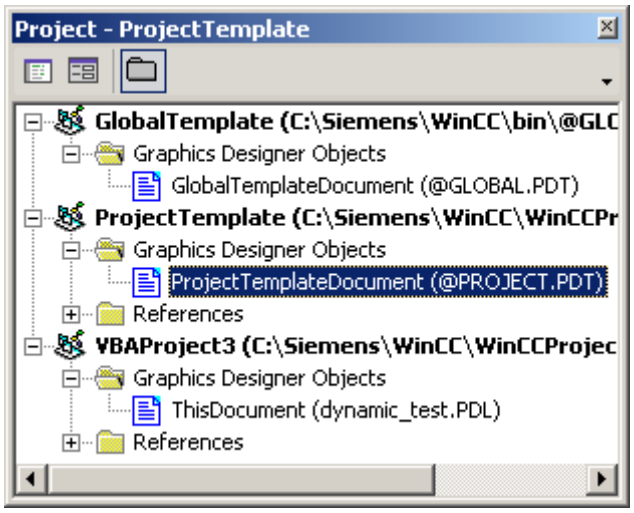
Pour dynamiser une propriété par un script en C

Introduction

Lorsque vous dynamisez une propriété avec un script en C, vous pouvez attribuer le code C à la propriété "SourceCode". Le code C est compilé en arrière-plan. La propriété "Compiled" retourne "True" lorsque le code C a été compilé correctement.

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor").
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



3. Pour dynamiser une propriété d'objet avec un script en C, insérez p. ex. la procédure "AddDynamicAsCSkriptToProperty()" dans le document. L'exemple qui suit est celui de l'augmentation de la hauteur d'un cercle de 5 pixels toutes les 2 secondes dans Runtime :
```
Sub AddDynamicAsCSkriptToProperty()
'VBA61
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
Dim strCode As String
strCode = "long lHeight;" & vbCrLf & "int check;" & vbCrLf
strCode = strCode & "GetHeight("" & "events.PDL"" & ","" &
"myCircle"" & ");" & vbCrLf
strCode = strCode & "lHeight = lHeight+5;" & vbCrLf
strCode = strCode & "check = SetHeight(""events.PDL"",
""myCircle"", lHeight );"
strCode = strCode & vbCrLf & "//Return-Type: BOOL" & vbCrLf
strCode = strCode & "return check;"
Set objCircle =
ActiveDocument.HMIObjects.AddHMIObject("myCircle", "HMICircle")
'Create dynamic for Property "Height":
Set objCScript =
objCircle.Height.CreateDynamic(hmiDynamicCreationTypeCScript)
'
'set Sourcecode and cycletime:
With objCScript
.SourceCode = strCode
.Trigger.Type = hmiTriggerTypeStandardCycle
.Trigger.CycleType = hmiCycleType_2s
.Trigger.Name = "Trigger1"
End With
End Sub
```
4. Démarrez la procédure avec <F5>.

Voir aussi

Propriété Trigger (Page 2377)

Propriété ScriptType (Page 2342)

Propriété SourceCode (Page 2357)

Propriété CycleType (Page 2148)

Objet ScriptInfo (Page 2009)

Méthode CreateDynamic (Page 1803)

Pour dynamiser une propriété par un script VB (Page 1691)

Pour dynamiser une propriété par le dialogue Dynamique (Page 1685)

Pour dynamiser une propriété par liaison à une variable (Page 1684)

Dynamisation des propriétés d'une vue ou d'un objet (Page 1682)

Création de dynamisations avec VBA (Page 1680)

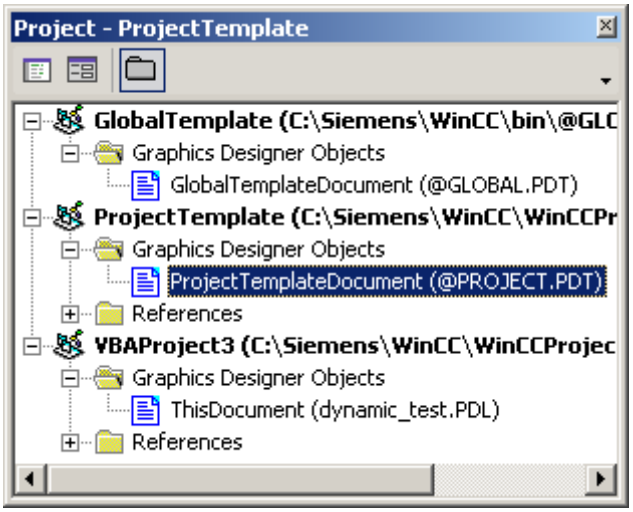
Pour dynamiser une propriété par un script VB

Introduction

Lorsque vous dynamisez une propriété avec un script VB, vous pouvez attribuer le code VB à la propriété "SourceCode". Le script VB est compilé en arrière-plan. La propriété "Compiled" retourne "True" lorsque la syntaxe du code VB est correcte.

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor").
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



3. Pour dynamiser une propriété d'objet avec un script VB, insérez p. ex. la procédure "AddDynamicAsVBSkriptToProperty()" dans le document. L'exemple qui suit est celui de l'augmentation du rayon d'un cercle de 5 pixels toutes les 2 secondes dans Runtime :
```
Sub AddDynamicAsVBSkriptToProperty()
'VBA62
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle
Dim strCode As String
strCode = "Dim myCircle" & vbCrLf & "Set myCircle = "
strCode = strCode &
"HMIRuntime.ActiveScreen.ScreenItems(""myCircle"")"
strCode = strCode & vbCrLf & "myCircle.Radius = myCircle.Radius +
5"
Set objCircle =
ActiveDocument.HMIObjects.AddHMIObject("myCircle", "HMICircle")
'
'Create dynamic of property "Radius":
Set objVBScript =
objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
'
'Set SourceCode and cycletime:
With objVBScript
.SourceCode = strCode
.Trigger.Type = hmiTriggerTypeStandardCycle
.Trigger.CycleType = hmiCycleType_2s
.Trigger.Name = "Trigger1"
End With
End Sub
```
4. Démarrez la procédure avec <F5>.

Voir aussi

Pour dynamiser une propriété par un script en C (Page 1688)

Propriété Trigger (Page 2377)

Propriété SourceCode (Page 2357)

Propriété CycleType (Page 2148)

Objet ScriptInfo (Page 2009)

Méthode CreateDynamic (Page 1803)

Pour dynamiser une propriété par le dialogue Dynamique (Page 1685)

Pour dynamiser une propriété par liaison à une variable (Page 1684)

Dynamisation des propriétés d'une vue ou d'un objet (Page 1682)

Création de dynamisations avec VBA (Page 1680)

3.3.5.3 Configuration d'actions commandées par événement avec VBA

Configuration d'actions commandées par événement avec VBA

Introduction

Avec VBA, vous pouvez configurer des actions sur des vues ou des objets qui seront déclenchées à l'occurrence d'événements prédéfinis. Par exemple, dans Runtime, un clic de souris sur un objet peut appeler un script en C dont la valeur retournée sera utilisée pour dynamiser une propriété d'objet. Les méthodes de dynamisation disponibles sont les suivantes :

- Liaison directe
- Scripts

Les événements qui sont utilisés pour configurer des actions commandées par événement interviennent exclusivement dans Runtime et n'ont aucun rapport avec les gestionnaires d'événements VBA.

Principe

Pour configurer avec VBA des actions commandées par événement, on utilise la propriété Events. L'utilisation de cette propriété est différente selon que vous souhaitez configurer une action sur un objet, une vue ou une propriété.

Configuration d'une action sur un objet ou une vue

L'action que vous configurez sur un objet ou une vue sera déclenchée à l'occurrence d'un événement prédéfini, p. ex. lors d'un clic de souris sur l'objet. Pour configurer une action sur un objet avec VBA, on utilise la propriété "Events(Index)", dans laquelle "Index" désigne l'événement déclencheur :

```
Sub AddActionToObjectTypeCScript()
'VBA63
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
'Create circle. Click on object executes an C-action
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB", "HMICircle")
Set objEvent = objCircle.Events(1)
Set objCScript = objEvent.Actions.AddAction(hmiActionCreationTypeCScript)
'
'Assign a corresponding custom-function to the property "SourceCode":
objCScript.SourceCode = ""
End Sub
```

Configuration d'une action sur une propriété

L'action que vous configurez sur une propriété d'une vue ou d'un objet sera déclenchée à la modification de la valeur de cette propriété. Pour configurer une action sur une propriété avec

VBA, on utilise la propriété "Events(1)", dans laquelle l'index "1" correspond à l'événement "En cas de modification" :

```
Sub AddActionToPropertyTypeCScript()
'VBA64
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
'Create circle. Changing of the Property
'"Radius" should be activate C-Aktion:
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB", "HMICircle")
Set objEvent = objCircle.Radius.Events(1)
Set objCScript = objEvent.Actions.AddAction(hmiActionCreationTypeCScript)
'
'Assign a corresponding custom-function to the property "SourceCode":
objCScript.SourceCode = ""
End Sub
```

Liaison directe

Pour configurer une liaison directe, on utilise l'objet DirectConnection.

Scripts

Lorsqu'un événement doit déclencher une action C ou VB, on utilise l'objet ScriptInfo.

Voir aussi

Pour configurer avec VBA une action VB déclenchée à l'occurrence d'un évènement (Page 1699)

Propriété Events (Page 2159)

Objet ScriptInfo (Page 2009)

Objet Event (Page 1923)

Pour configurer avec VBA une action C à l'occurrence d'un événement (Page 1697)

Pour configurer une liaison directe avec VBA (Page 1694)

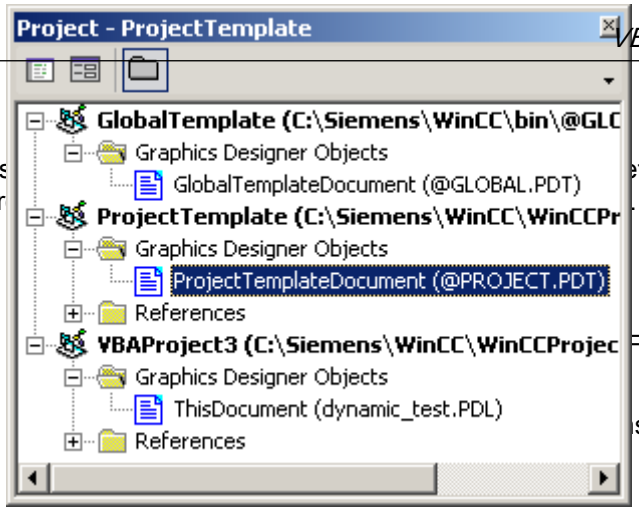
Gestion des événements (Page 1704)

Création de dynamisations avec VBA (Page 1680)

Pour configurer une liaison directe avec VBA

Introduction

Les instructions suivantes se rapportent à la configuration d'une liaison directe à l'aide de deux propriétés d'objet. Vous trouverez des renseignements supplémentaires sur la configuration



des ette documentation, aux rubriques "Pr

Marche à suivre

1. F11> ou "Outils" > "Macros" > "Visual

2. s lequel vous voulez créer le code

3. Pour ajouter une liaison directe à une propriété d'objet, insérez p. ex. la procédure "AddDirectConnectionToObject()" dans le document. Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub DirectConnection()
'VBA65
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleA As HMIRectangle
Dim objEvent As HMIEvent
Dim objDConnection As HMIDirectConnection
'
'Create objects:
Set objRectangleA =
ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A",
"HMIRectangle")
Set objRectangleA =
ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A",
"HMIRectangle")
Set objButton =
ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Text = "SetPosition"
End With
'
'Directconnection is initiated by mouseclick:
Set objDConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectC
onnection)
With objDConnection
'Sourceobject: Property "Top" of Rectangle_A
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Destinationobject: Property "Left" of Rectangle_B
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
```

```
End Sub
```

4. Démarrez la procédure avec <F5>.

Voir aussi

Propriété ObjectName (Page 2293)

Propriété AutomationName (Page 2070)

Objet SourceLink (Page 2016)

Objet DestLink (Page 1904)

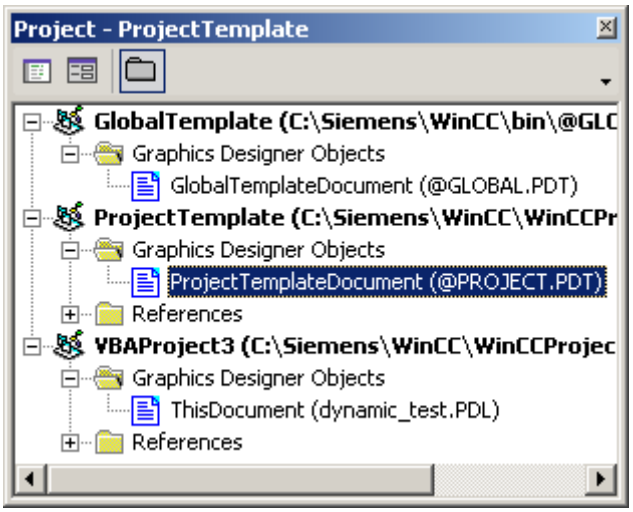
Objet DirectConnection (Page 1906)

Configuration d'actions commandées par événement avec VBA (Page 1693)

Pour configurer avec VBA une action C à l'occurrence d'un événement

Marche à suivre

1. Dans Graphics Designer, ouvrez l'Editeur VBA (<Alt+F11> ou "Outils" > "Macros" > "Visual Basic Editor")
2. Dans l'Explorateur de projet, ouvrez le document dans lequel vous voulez créer le code VBA :



3. Pour configurer avec VBA une action C à l'occurrence d'un événement, insérez p. ex. la procédure "CreateCActionToClickedEvent()" dans le document. Dans l'exemple qui suit, on insère un bouton et un cercle dans la vue active. Dans Runtime, la hauteur augmente à chaque clic sur le bouton :

```
Sub CreateCActionToClickedEvent()
'VBA66
Dim objButton As HMIButton
Dim objCircle As HMICircle
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim strCode As String
strCode = "long lHeight;" & vbCrLf & "int check;" & vbCrLf
strCode = strCode & "lHeight = GetHeight (""events.PDL"",
""myCircle"");"
strCode = strCode & vbCrLf & "lHeight = lHeight+5;" & vbCrLf &
"check = "
strCode = strCode & "SetHeight(""events.PDL"",
""myCircle"",lHeight);"
strCode = strCode & vbCrLf & "//Return-Type: Void"
Set objCircle =
ActiveDocument.HMIObjects.AddHMIObject("myCircle", "HMICircle")
Set objButton =
ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircle
.Top = 100
.Left = 100
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Text = "Increase height"
End With
'Configure directconnection:
Set objCScript =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeCScript
)
With objCScript
'
'Note: Replace "events.PDL" with your picturename
.SourceCode = strCode
End With
End Sub
```

4. Démarrez la procédure avec <F5>.

Voir aussi

Objet ScriptInfo (Page 2009)

Obj

Obj

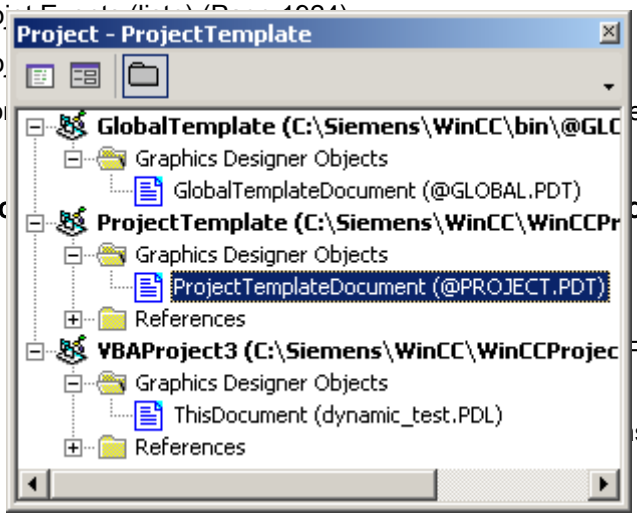
Co ec VBA (Page 1693)

Pour configurer avec d'un évènement

Marche à suivre

1. F11> ou "Outils" > "Macros" > "Visual

2. s lequel vous voulez créer le code



3. Pour configurer dans VBA une action VB déclenchée par événement, insérez p. ex. la procédure "CreateVBActionToClickedEvent()" dans le document. Dans l'exemple qui suit, on insère un bouton et un cercle dans la vue active. Dans Runtime, le rayon du cercle augmente à chaque clic sur le bouton :

```
Sub CreateVBActionToClickedEvent()
'VBA67
Dim objButton As HMIButton
Dim objCircle As HMICircle
Dim objEvent As HMIEvent
Dim objVBScript As HMIScriptInfo
Dim strCode As String
strCode = "Dim myCircle" & vbCrLf & "Set myCircle = "
strCode = strCode &
"HMIRuntime.ActiveScreen.ScreenItems(""Circle_VB"")"
strCode = strCode & vbCrLf & "myCircle.Radius = myCircle.Radius +
5"
Set objCircle =
ActiveDocument.HMIObjects.AddHMIObject("Circle_VB", "HMICircle")
Set objButton =
ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircle
.Top = 100
.Left = 100
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 120
.Text = "Increase Rayon"
End With
'Define event and assign sourcecode:
Set objVBScript =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeVBScrip
t)
With objVBScript
.SourceCode = strCode
End With
End Sub
```

4. Démarrez la procédure avec <F5>.

Voir aussi

Objet Actions (liste) (Page 1872)

Objet ScriptInfo (Page 2009)

Objet Events (liste) (Page 1924)

Configuration d'actions commandées par événement avec VBA (Page 1693)

3.3.5.4 Edition de déclencheurs

Edition de déclencheurs

Introduction

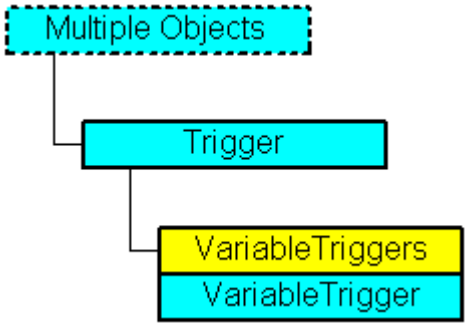
On utilise les déclencheurs dans le cadre de la dynamisation d'objets graphiques et pour déclencher des actions liées à des propriétés d'objet. Les déclencheurs peuvent être par exemple :

- Variables : Modification, franchissement du seuil supérieur ou inférieur d'une valeur de variable,
- Cycle standard : exécution cyclique de l'action. Les cycles peuvent être réglés de 250 ms à 1 heure. Vous pouvez aussi disposer de cycles utilisateur personnalisés,
- Cycle de vue : Le déclenchement utilisé est un déclenchement cyclique. Ce cycle vous offre la possibilité de déterminer de façon centralisée les cycles de toutes les actions configurées dans une vue, des connections variables et des dialogues dynamiques.
- Cycle fenêtres: Le déclenchement utilisé est un déclenchement cyclique. Cette valeur est valable pour toutes les actions utilisées dans fenêtre vue, pour les connections variables et les dialogues dynamiques qui ont été configurés en mode déclencheur "cycle fenêtre".

Pour configurer une action qui doit s'exécuter à l'occurrence d'un événement sur un objet graphique, l'événement en question est le déclencheur.

Edition de déclencheurs avec VBA

Pour éditer un déclencheur avec VBA, on utilise l'objet Trigger. Si le déclencheur doit être une variable, utilisez l'objet VariableTrigger :



On définit le type de déclencheur par la propriété Type. Pour éditer la variable servant de déclencheur, on utilise la propriété "VariableTriggers".

Voir aussi

Exemples d'édition de déclencheurs avec VBA (Page 1702)

Objet VariableTrigger (Page 2048)

Objet Trigger (Page 2035)

Objet ScriptInfo (Page 2009)

Exemples d'édition de déclencheurs avec VBA

Introduction

Les quatre exemples qui suivent se rapportent à la création avec VBA des déclencheurs suivants :

- Cycle standard
- Variable
- Cycle vue
- Cycle fenêtre

Tous les exemples se rapportent à l'insertion dans la vue active d'un cercle dont le rayon est dynamisé par une action VB.

Vous trouverez la marche à suivre pour dynamiser une propriété par liaison à une variable à la rubrique "Dynamisation d'une propriété par liaison à une variable" dans cette documentation.

Exemple 1 : Cycle standard

```
Sub DynamicWithStandardCycle()
'VBA68
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_Standard", "HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
With objVBScript
.Trigger.Type = hmiTriggerTypeStandardCycle
'"CycleType"-specification is necessary:
.Trigger.CycleType = hmiCycleType_10s
.Nom.Trigger = "VBA_StandardCycle"
.SourceCode = ""
End With
End Sub
```

Exemple 2 : Variable

```
Sub DynamicWithVariableTriggerCycle()
```

```
'VBA69
Dim objVBScript As HMIScriptInfo
Dim objVarTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_VariableTrigger",
"HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
With objVBScript
Set objVarTrigger = .Trigger.VariableTriggers.Add("VarTrigger", hmiVariableCycleType_10s)
.SourceCode = ""
End With
End Sub
```

Exemple 3 : Cycle vue

```
Sub DynamicWithPictureCycle()
'VBA70
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_Picture", "HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
With objVBScript
.Trigger.Type = hmiTriggerTypePictureCycle
.Nom.Trigger = "VBA_PictureCycle"
.SourceCode = ""
End With
End Sub
```

Exemple 4 : Cycle fenêtre

```
Sub DynamicWithWindowCycle()
'VBA71
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_Window", "HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
With objVBScript
.Trigger.Type = hmiTriggerTypeWindowCycle
.Trigger.Name = "VBA_WindowCycle"
.SourceCode = ""
End With
End Sub
```

Voir aussi

Objet VariableTrigger (Page 2048)

Objet Trigger (Page 2035)

Objet ScriptInfo (Page 2009)

Edition de déclencheurs (Page 1701)

3.3.6 Gestion des événements

Introduction

Dans le Graphics Designer, certaines actions (p. ex. l'ouverture d'une vue) sont liées à des événements. Un événement peut déclencher l'exécution d'instructions à l'aide d'un gestionnaire d'événement VBA prédéfini.

Les événements surviennent seulement lors d'une configuration dans le Graphics Designer et ne sont pas disponibles dans Runtime. Ces événements ne doivent pas être confondus avec les événements (comme un clic de souris ou une modification de propriété) liés à des objets graphiques ou à des vues.

Remarque

Quand Graphics Designer est ouvert, des événements sont aussi

déclenchés par d'autres éditeurs. Ceci concerne p. ex. la modification de propriétés de vues dans WinCCExplorer. Fermez Graphics Designer avant d'entreprendre des modifications sur des vues dans d'autres éditeurs. Vous éviterez ainsi l'apparition impromptue d'événements.

Remarque

Si vous ouvrez une vue dans Graphics Designer, vous déclenchez non pas seulement l'"événement DocumentOpened" de la vue active mais aussi celle du "Project Template" et du "Global Template". Le code VBA de l'"événement DocumentOpened" est donc exécuté en double.

Vous devez intercepter ce comportement avec le gestionnaire d'évènements.

Principe

Dans le cadre de la gestion d'événements, on distingue les événements avec ou sans transmission. On reconnaît un événement avec transmission à son paramètre "CancelForwarding". Un événements sans transmission ne possède pas ce paramètre. En l'occurrence, l'événement est envoyé à la vue active, puis transmis au modèle général ("Global Template").

Un événement avec transmission est donc toujours transmis au document "Modèle général" par l'intermédiaire du document "Modèle projet".

Interdire la transmission

Vous pouvez interdire la transmission d'un événement en réglant le paramètre "CancelForwarding" sur "True" dans le gestionnaire d'événements VBA :

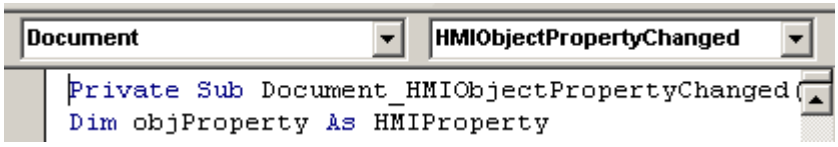
```
Sub Document_HMIObjectPropertyChanged(ByVal Property As IHMIProperty, CancelForwarding As
Boolean)
'VBA72
CancelForwarding = True
MsgBox "Object's property has been changed!"
End Sub
```

Evénements spécifiques à une vue ou d'une application

Indépendamment du caractère avec ou sans transmission des événements mentionné ci-dessus, le Graphics Designer fait la distinction entre les événements spécifiques à une vue et les événements spécifiques à une application :

Evénements spécifiques à une vue

Les événements spécifiques à une vue répondent toujours aux actions qui interviennent sur la vue active dans le Graphics Designer. Ces actions peuvent être, p. ex., une modification d'une propriété d'objet ou l'enregistrement de la vue active. Vous trouverez une liste des événements spécifiques à une vue qui sont disponibles en sélectionnant "Document" dans l'Editeur VBA :



Evénements spécifiques à une application

Les événements spécifiques à une application agissent sur les actions qui interviennent dans l'application "Graphics Designer". Ces actions peuvent être, p. ex., le démarrage du Graphics Designer ou la création d'un objet dans la bibliothèque des modules.

Pour pouvoir disposer des événements spécifiques de l'application, on écrit dans l'Editeur VBA l'instruction suivante au début du document (de préférence le "Modèle projet" ou le "Modèle général") :

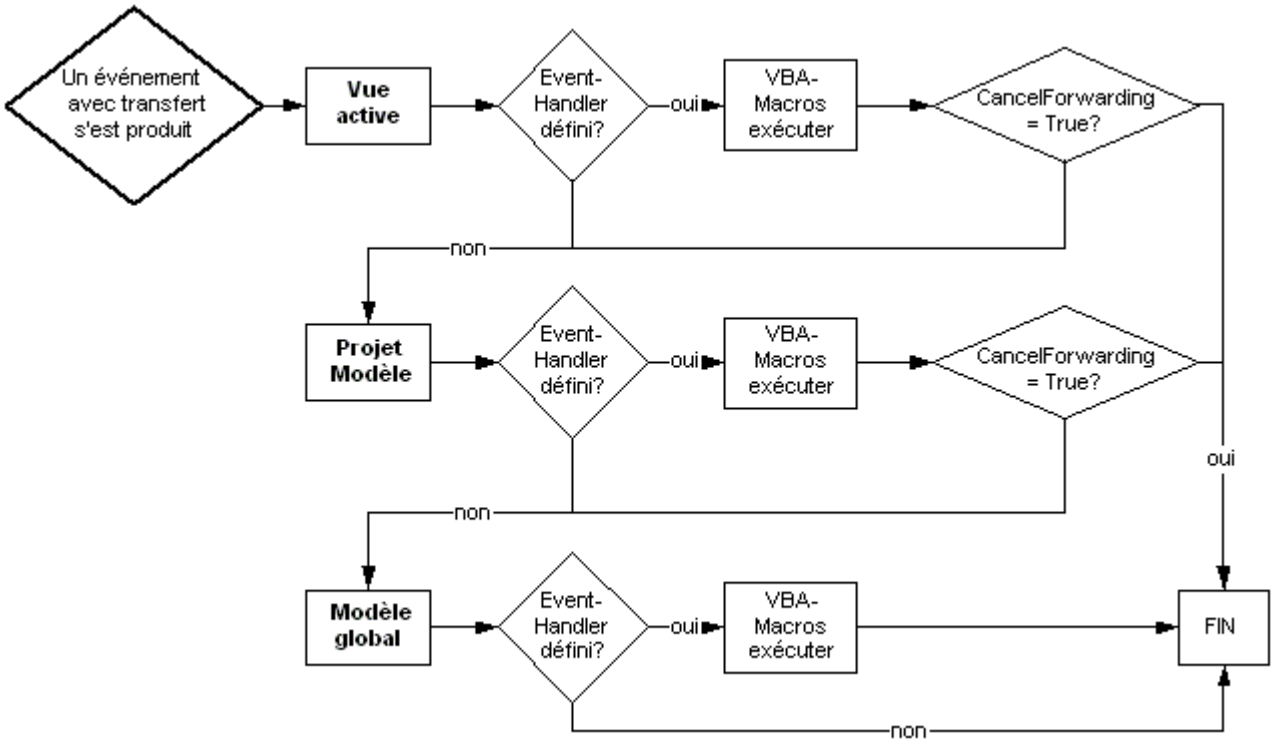
```
Dim WithEvents <Name> As grafexe.Application
```

Par cette instruction, seuls les événements spécifiques de l'application qui se trouvent dans la liste du Graphics Designer peuvent être sélectionnés :

```
objGDApplication                    Started
Dim WithEvents objGDApplication As grafexe.Application

Private Sub objGDApplication_Started()
MsgBox ("Graphics Designer has been started")
End Sub
```

Exemple 1 : Occurrence d'un événement avec transmission

La figure représente la séquence opérationnelle à l'occurrence d'un événement avec transmission :

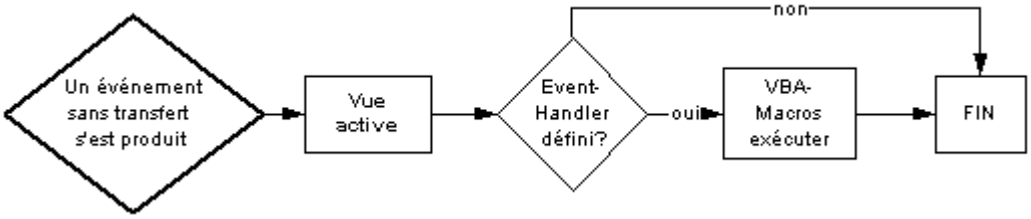


Remarque

Il existe des événements qui sont spécifiques à la fois d'une vue et de l'application (p. ex.: BeforeDocumentSave). A l'occurrence d'un événement de ce type, le système vérifie si le gestionnaire d'événement spécifique àl'application correspondant avait été bien défini. Ce n'est qu'après cette vérification que la séquence ci-dessus est lancée.

Exemple 2 : Occurrence d'un événement avec transmission

La figure représente la séquence opérationnelle à l'occurrence d'un événement sans transmission :



Désactivation de la gestion d'événements

Vous pouvez désactiver la gestion d'événements en réglant la propriété "DisableVBAEvents" de l'objet Application sur "True".

Voir aussi

Propriété DisableVBAEvents (Page 2152)

Organisation du code VBA dans un projet WinCC (Page 1607)

3.3.7 Accès aux applications externes avec VBA

3.3.7.1 Accès aux applications externes avec VBA

Introduction

VBA vous permet d'accéder à des programmes compatibles VBA, p. ex. les produits de la suite Microsoft-Office. Vous avez ainsi, p. ex., la possibilité de lire les valeurs d'un tableau Excel et de les attribuer à des propriétés d'objet.

Remarque

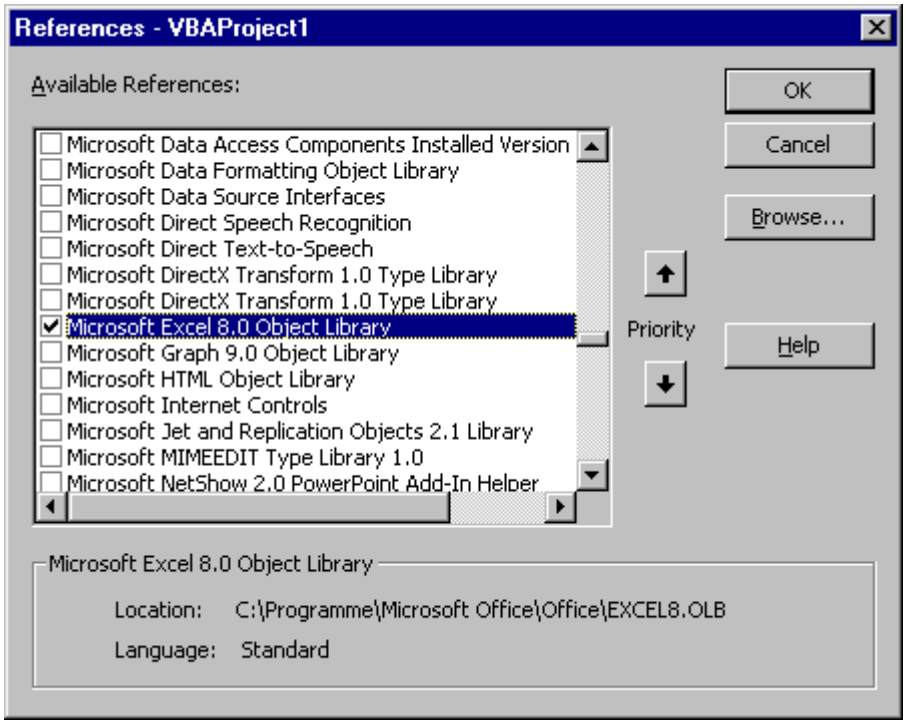
Pas d'utilisation directe d'Unicode dans Excel-VBA et Word-VBA

Dans Excel-VBA et Word-VBA, il n'est possible d'utiliser les caractères Unicode que par le biais de la fonction <ChrW(unicode-id).

Déclaration de l'application externe

Vous devez lier l'application externe à l'Editeur VBA pour pouvoir disposer de sa bibliothèque d'objets. A cet effet, dans l'Editeur VBA, sélectionnez dans le menu "Tools" la commande

"References". Dans le dialogue "References", sélectionnez ensuite la bibliothèque d'objets souhaitée :



Remarque

Dans l'Explorateur de projets de l'Editeur VBA, vous devez lier l'application externe à tous les projets qui doivent faire appel à cette application.

Voir aussi

Exemple : Accès à MS Excel avec VBA (Page 1708)

3.3.7.2 Exemple : Accès à MS Excel avec VBA

Introduction

Les trois exemples qui suivent montrent l'accès à MS Excel. Pour que ces exemples fonctionnent, la bibliothèque d'objets de MS Excel doit être liée par une référence.

Remarque

Pas d'utilisation directe d'Unicode dans Excel-VBA et Word-VBA

Dans Excel-VBA et Word-VBA, il n'est possible d'utiliser les caractères Unicode que par le biais de la fonction <ChrW(unicode-id).

Exemple 1

L'exemple qui suit est celui de l'exportation de la liste des objets par défaut du Graphics Designer vers un formulaire Excel. L'exportation tient compte des propriétés d'objet et des dynamisations éventuelles. En outre, le type de données VBA est affiché.

```
Sub ExportDefObjListToXLS()
'VBA73
'Microsoft Excel Object Library needs to be referenced
Dim objGDApplication As grafexe.Application
Dim objHMIObject As grafexe.HMIObject
Dim objProperty As grafexe.HMIProperty
Dim objXLS As Excel.Application
Dim objWSheet As Excel.Worksheet
Dim objWBook As Excel.Workbook
Dim rngSelection As Excel.Range
Dim lRow As Long
Dim lRowGroupStart As Long

'define local errorhandler
On Local Error GoTo LocErrTrap

'Set references to the applications Excel and GraphicsDesigner
Set objGDApplication = grafexe.Application
Set objXLS = New Excel.Application

'Create workbook
Set objWBook = objXLS.Workbooks.Add()
objWBook.SaveAs objGDApplication.ApplicationDataPath & "DefaultObjekte.xls"

'Adds new worksheet to the new workbook
Set objWSheet = objWBook.Worksheets.Add
objWSheet.Name = "DefaultObjekte"
lRow = 1

'Every object of the DefaultHMIObjects-collection will be written
'to the worksheet with their objectproperties.
'For better overview the objects will be grouped.
For Each objHMIObject In objGDApplication.DefaultHMIObjects
DoEvents
objWSheet.Cells(lRow, 1).value = objHMIObject.ObjectName
objWSheet.Cells(lRow, 2).value = objHMIObject.Type
lRow = lRow + 1
lRowGroupStart = lRow
```

```
For Each objProperty In objHMIObject.Properties
'Write displayed name and automationname of property
'into the worksheet
objWSheet.Cells(lRow, 2).value = objProperty.DisplayName
objWSheet.Cells(lRow, 3).value = objProperty.Name
'Write the value of property, datatype and if their dynamicable
'into the worksheet
If Not IsEmpty(objProperty.value) Then _
        objWSheet.Cells(lRow, 4).value = objProperty.value
objWSheet.Cells(lRow, 5).value = objProperty.IsDynamicable
objWSheet.Cells(lRow, 6).value = TypeName(objProperty.value)
objWSheet.Cells(lRow, 7).value = VarType(objProperty.value)
lRow = lRow + 1
Next objProperty
'Select and groups the range of object-properties in the worksheet
Set rngSelection = objWSheet.Range(objWSheet.Rows(lRowGroupStart), _
                    objWSheet.Rows(lRow - 1))
rngSelection.Select
rngSelection.Group
Set rngSelection = Nothing
'Insert empty row
lRow = lRow + 1
Next objHMIObject
objWSheet.Columns.AutoFit
Set objWSheet = Nothing
objWBook.Save
objWBook.Close
Set objWBook = Nothing
objXLS.Quit
Set objXLS = Nothing
Set objGDApplication = Nothing
Exit Sub
LocErrTrap:
MsgBox Err.Description, , Err.Source
Resume Next
End Sub
```

Exemple 2

L'exemple qui suit est celui de l'exportation de tous les objets de la vue active vers un tableau Excel. L'importation tient compte des propriétés Position X, Position Y, Largeur, Hauteur et Niveau :

```
Sub ExportObjectListToXLS()
'VBA74
Dim objGDApplication As grafexe.Application
Dim objDoc As grafexe.Document
Dim objHMIObject As grafexe.HMIObject
Dim objProperty As grafexe.HMIProperty
Dim objXLS As Excel.Application
Dim objWSheet As Excel.Worksheet
Dim objWBook As Excel.Workbook
Dim lRow As Long
```

```
'Define local errorhandler
On Local Error GoTo LocErrTrap

'Set references on the applications Excel and GraphicsDesigner
Set objGDApplication = grafexe.Application
Set objDoc = objGDApplication.ActiveDocument
Set objXLS = New Excel.Application

'Create workbook
Set objWBook = objXLS.Workbooks.Add()
objWBook.SaveAs objGDApplication.ApplicationDataPath & "Export.xls"

'Create worksheet in the new workbook and write headline
'The name of the worksheet is equivalent to the documents name
Set objWSheet = objWBook.Worksheets.Add
objWSheet.Name = objDoc.Name
objWSheet.Cells(1, 1) = "Nom de l'objet"
objWSheet.Cells(1, 2) = "Type d'objet"
objWSheet.Cells(1, 3) = "ProgID"
objWSheet.Cells(1, 4) = "Position X"
objWSheet.Cells(1, 5) = "Position Y"
objWSheet.Cells(1, 6) = "Largeur"
objWSheet.Cells(1, 7) = "Hauteur"
objWSheet.Cells(1, 8) = "Niveau"
lRow = 3

'Every objects will be written with their objectproperties width,
'height, pos x, pos y and layer to Excel. If the object is an
'ActiveX-Control the ProgID will be also exported.
For Each objHMIObject In objDoc.HMIObjects
DoEvents
objWSheet.Cells(lRow, 1).value = objHMIObject.ObjectName
objWSheet.Cells(lRow, 2).value = objHMIObject.Type
If UCase(objHMIObject.Type) = "HMIACTIVEXCONTROL" Then
objWSheet.Cells(lRow, 3).value = objHMIObject.ProgID
End If
objWSheet.Cells(lRow, 4).value = objHMIObject.Left
objWSheet.Cells(lRow, 5).value = objHMIObject.Top
objWSheet.Cells(lRow, 6).value = objHMIObject.Width
objWSheet.Cells(lRow, 7).value = objHMIObject.Height
objWSheet.Cells(lRow, 8).value = objHMIObject.Layer
lRow = lRow + 1
Next objHMIObject
objWSheet.Columns.AutoFit
Set objWSheet = Nothing
objWBook.Save
objWBook.Close
Set objWBook = Nothing
objXLS.Quit
Set objXLS = Nothing
Set objDoc = Nothing
Set objGDApplication = Nothing
Exit Sub
LocErrTrap:
MsgBox Err.Description, , Err.Source
```

```
Resume Next
End Sub
```

Exemple 3

L'exemple qui suit est celui de l'importation d'objets depuis le tableau Excel créé dans l'exemple 2. L'importation tient compte des propriétés Position X, Position Y, Largeur, Hauteur et Niveau :

```
Sub ImportObjectListFromXLS()
'VBA75
Dim objGDApplication As grafexe.Application
Dim objDoc As grafexe.Document
Dim objHMIObject As grafexe.HMIObject
Dim objXLS As Excel.Application
Dim objWSheet As Excel.Worksheet
Dim objWBook As Excel.Workbook
Dim lRow As Long
Dim strWorkbookName As String
Dim strWorksheetName As String
Dim strSheets As String

'define local errorhandler
On Local Error GoTo LocErrTrap

'Set references on the applications Excel and GraphicsDesigner
Set objGDApplication = Application
Set objDoc = objGDApplication.ActiveDocument
Set objXLS = New Excel.Application

'Open workbook. The workbook have to be in datapath of GraphicsDesigner
strWorkbookName = InputBox("Name of workbook:", "Import of objects")
Set objWBook = objXLS.Workbooks.Open(objGDApplication.ApplicationDataPath &
strWorkbookName)
If objWBook Is Nothing Then
MsgBox "Open workbook fails!" & vbCrLf & "This function is cancled!", vbCritical, "Import
od objects"
Set objDoc = Nothing
Set objGDApplication = Nothing
Set objXLS = Nothing
Exit Sub
    End If

'Read out the names of all worksheets contained in the workbook
For Each objWSheet In objWBook.Sheets
strSheets = strSheets & objWSheet.Name & vbCrLf
Next objWSheet
strWorksheetName = InputBox("Name of table to import:" & vbCrLf & strSheets, "Import of
objects")
Set objWSheet = objWBook.Sheets(strWorksheetName)
lRow = 3

'Import the worksheet as long as in actual row the first column is empty.
```

```
'Add with the outreaded data new objects to the active document and
'assign the values to the objectproperties
With objWSheet
While (.Cells(lRow, 1).value <> vbNullString) And (Not IsEmpty(.Cells(lRow, 1).value))
'Add the objects to the document as its objecttype,
'do nothing by groups, their have to create before.
If (UCase(.Cells(lRow, 2).value) = "HMIGROUP") Then
Else
  If (UCase(.Cells(lRow, 2).value) = "HMIACTIVEXCONTROL") Then
    Set objHMIObject = objDoc.HMIObjects.AddActiveXControl(.Cells(lRow,
1).value, .Cells(lRow, 3).value)
    Else
    Set objHMIObject = objDoc.HMIObjects.AddHMIObject(.Cells(lRow, 1).value, .Cells(lRow,
2).value)
  End If
  objHMIObject.Left = .Cells(lRow, 4).value
  objHMIObject.Top = .Cells(lRow, 5).value
  objHMIObject.Width = .Cells(lRow, 6).value
  objHMIObject.Height = .Cells(lRow, 7).value
  objHMIObject.Layer = .Cells(lRow, 8).value
    End If
Set objHMIObject = Nothing
lRow = lRow + 1
Wend
End With
objWBook.Close
Set objWBook = Nothing
objXLS.Quit
Set objXLS = Nothing
Set objDoc = Nothing
Set objGDApplication = Nothing
Exit Sub
LocErrTrap:
MsgBox Err.Description, , Err.Source
Resume Next
End Sub
```

3.4 AddIns

3.4.1 AddIns

Introduction

Un AddIn est un code non visualisable enregistré sous forme de DLL. Les AddIns vous permettent de disposer de nouvelles fonctions par l'enregistrement de la DLL correspondante dans le système d'exploitation et par chargement dans Graphics Designer.

Si vous êtes utilisateur, vous pouvez mettre à profit des AddIns qui proposent des fonctions spécifiquement adaptées à l'utilisation considérée. Par exemple, si vous travaillez sur divers ordinateurs de projet et que vous utilisez souvent des macros VBA, vous pouvez réunir ces dernières dans un ou plusieurs VBA. Si vous changez d'ordinateur, il vous suffit de copier l'AddIn pour pouvoir retrouver vos fonctions habituelles sur le nouveau poste.

Si vous êtes développeur, vous pouvez créer des MS avec la bibliothèque de programmes dans un environnement de développement, comme p. ex. AddIns Visual Studio 6.0, et protéger ainsi vos codes contre l'intrusion.

Les AddIns dans Graphics Designer

Dans Graphics Designer, vous pouvez utiliser tous les AddIns qui ont été développés pour cet éditeur et qui sont dans enregistrés dans le système d'exploitation de l'ordinateur de projet.

Si vous avez souvent besoin des fonctions d'un AddIn, vous pouvez le charger automatiquement au démarrage de Graphics Designer.

Si vous n'avez plus besoin des fonctions d'un AddIns, vous pouvez le décharger à tout moment.

Voir aussi

Pour charger un AddIn dans Graphics Designer (Page 1716)

Exemple : Création d'addins (Page 1718)

Ajouts d'addIns (Page 1714)

3.4.2 Ajouts d'addIns

Introduction

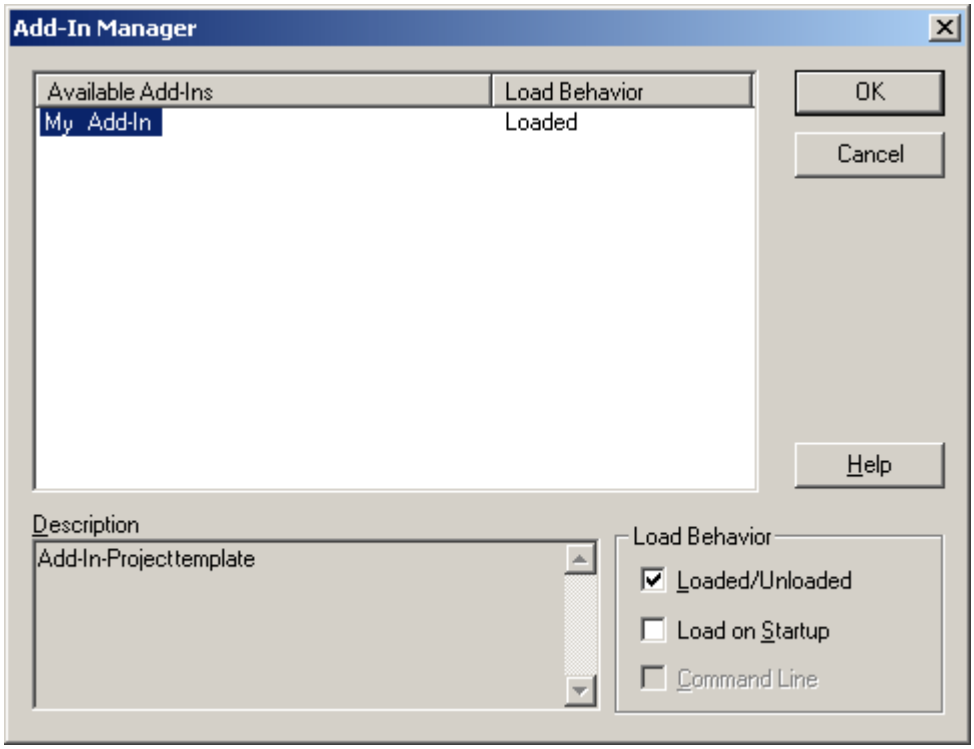
Dans Graphics Designer, on utilise AddIn Manager pour définir le comportement de chargement des AddIns qui pourront être utilisés dans Graphics Designer.

Conditions

- Un AddIn doit être enregistré dans le système d'exploitation, p. ex. à l'invite du programme à l'aide de la commande "regsvr32 filename.dll".
- Pour enregistrer des AddIn VBA, "Microsoft Visual Basic for Applications" doit être installé. L'installation s'effectue comme suit :
  - Microsoft Office : lors de l'installation de produits Microsoft Office, p.ex. MS Excel ou MS Word, Visual Basic for Applications est installé automatiquement.
  - Installation ultérieure à partir du programme setup de Microsoft Office : Dans le setup de Microsoft Office, vous pouvez installer de manière ciblée Visual Basic for Applications uniquement.
  - Téléchargement de l'environnement VBA Runtime : Microsoft propose les liens suivants pour télécharger l'environnement VBA Runtime :
    - fichier "VBRun60.exe" pour V6.0 : http://support.microsoft.com/kb/192461/ (http://support.microsoft.com/kb/192461/)
    - fichier "VBRun60sp6.exe" pour V6.0 SP6 : http://support.microsoft.com/kb/290887/ (http://support.microsoft.com/kb/290887/)

Démarrer l'AddIn Manager

Pour lancer AddIn Manager, sélectionnez dans Graphics Designer la commande "Macros" > "AddIn Manager" :



Chargement automatique d'un AddIn

Si l'AddIn contient de nouvelles fonctions dont vous avez toujours besoin dans Graphics Designer, vous pouvez le charger automatiquement à l'ouverture de Graphics Designer.

A cet effet, sélectionnez l'AddIn dans le gestionnaire d'AddIns et cochez la case de commande "Load on Startup".

Remarque

Selon la programmation de l'AddIn, sa fonction peut être entrée comme option de menu dans "Outils > Macros > AddIn". Vous pourrez alors lancer la fonction par un clic de souris.

Chargement et déchargement manuel d'un AddIn

Vous pouvez charger un AddIn manuellement si vous n'avez besoin de ses fonctions que pour une utilisation particulière (p. ex. pour une routine de contrôle).

Pour charger ou décharger manuellement un AddIn, sélectionnez-le dans AddIn-Manager et cochez la case d'option "Loaded/Unloaded".

Voir aussi

Pour charger un AddIn dans Graphics Designer (Page 1716)

Exemple : Création d'addins (Page 1718)

AddIns (Page 1714)

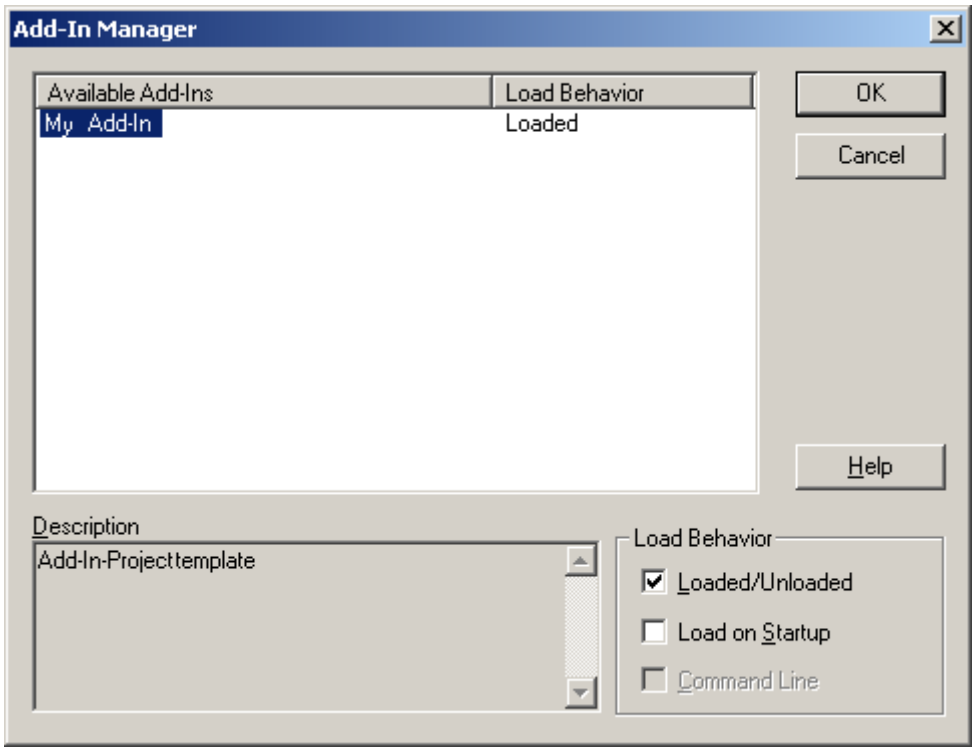
3.4.3 Pour charger un AddIn dans Graphics Designer

Condition

Un AddIn doit être enregistré dans le système d'exploitation, p. ex. à l'invite du programme à l'aide de la commande "regsvr32 filename.dll".

Marche à suivre

1. Démarrez Graphics Designer et ouvrez le projet dans lequel vous souhaitez insérer un AddIn.
2. Pour ouvrir AddIn Manager, sélectionnez la commande de menu "Outils" > AddIn Manager". AddIn Manager s'ouvre. Sous "Available AddIns" sont listés tous les AddIns avec leur état courant de chargement :



3. Pour chaque AddIn, vous devez déterminer s'il doit être chargé, et à quel moment. A cet effet, sélectionnez l'AddIn souhaité et cochez la case correspondante sous "Load Behavior".
4. Pour décharger un AddIn, sélectionnez l'AddIn concerné et décochez la case de commande Load/Unload sous "Load Behavior".
5. Cliquez sur OK.

Résultat

Selon la programmation de l'AddIn, sa fonction sera ajoutée à la liste du menu "Outils" > "AddIn" ou réagira à un gestionnaire d'événement dans Graphics Designer.

Pour que l'AddIn soit lancé en liaison avec un gestionnaire d'événement (p. ex. événement Started), la case de commande "On Startup" doit être cochée pour cet AddIn.

Voir aussi

Ajouts d'addIns (Page 1714)

AddIns (Page 1714)

3.4.4 Exemple : Création d'addins

3.4.4.1 Exemple : Création d'addins

Introduction

dans cette documentation, nous vous donnons un exemple de création d'un AddIns avec Visual Basic 6.0 qui soit utilisable dans Graphics Manager.

Conditions

MS Visual Studio 6.0 doit être installé sur l'ordinateur de projet.

Vous devez posséder des connaissances de programmation si vous souhaitez vous baser sur le code d'exemple pour créer vos propres AddIns.

Exemple : modèle de programme pour Visual Basic 6.0

Utilisez le gestionnaire d'événement "AddInInstance_OnConnection" pour ouvrir une session de Graphics Designer. L'identification de l'application est indispensable pour que l'AddIn puisse accéder à Graphics Designer.

Voir aussi

Exemple : Créer un addin avec Visual Basic 6.0 (Page 1718)

Ajouts d'addIns (Page 1714)

3.4.4.2 Exemple : Créer un addin avec Visual Basic 6.0

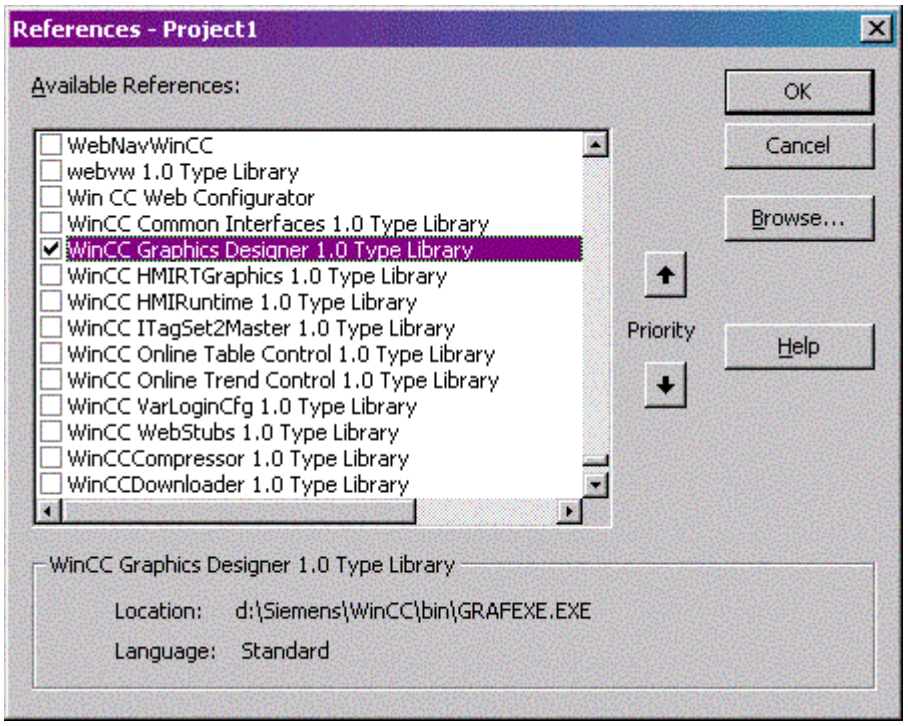
Introduction

Le code programme de cet exemple crée le fichier "MyAddIn.DLL". Pour que l'AddIn fonctionne dans Graphics Designer, vous devez cocher la case de commande "Load on Startup" dans l'AddIn Manager de Graphics Designer. A cet effet, vous pouvez aussi utiliser la fonction "LoadOnStartup" dans l'AddIn.

A l'ouverture de Graphics Designer, l'AddIn crée un menu personnalisé. En outre, vous pouvez appeler la fonction contenue dans l'AddIn par le menu "Outils" > "AddIns".

Condition

Pour pouvoir créer un AddIn qui fonctionne à partir du code d'exemple, "MS Visual Studio 6.0" doit être installé sur votre ordinateur. En outre, vous devez avoir référence la bibliothèque WinCC Graphics Designer 1.0 Type Library dans "MS Visual Studio 6.0" :



Marche à suivre

1. Ouvrez "MS Visual Studio 6.0" et créez un nouveau projet. Dans le dialogue "Nouveau projet", sélectionnez l'option AddIn et cliquez sur OK pour créer le projet.
2. Dans l'Explorateur de projets, ouvrez le dossier "Designer" et faites un double-clic sur l'option "Connect". Le dialogue "Connect (AddIn Designer)" s'ouvre.
3. Sous "Application", sélectionnez l'option "Graphics Designer" et sélectionnez le "Comportement de chargement au démarrage". Refermez le dialogue "Connect (AddIn Designer)".
4. Dans l'Explorateur de projets, ouvrez le dossier "Designer" et sélectionnez la commande "Afficher code" pour l'option "Connect" du menu contextuel.

5. Remplacez tout le code de programme par le code suivant :

```
Option Explicit
'-----------------------------------------------------
'Member Variables
'-----------------------------------------------------
'Reference to the AddIn Connection
Dim WithEvents ThisAddIn As grafexe.AddinHook
'Reference to the Grafics Designer Application
Dim WithEvents GrafApp As grafexe.Application
'-----------------------------------------------------
'WithEvents AddInInstance IDTExtensibility2 (automatic)
'-----------------------------------------------------
'-------------------------------------------------------
'This method connects the AddIn To the Graphic Designer Application
'-----------------------------------------------------
Private Sub AddinInstance_OnConnection(ByVal Application As
Object, _
                                        ByVal ConnectMode As
AddInDesignerObjects.ext_ConnectMode, _
                                        ByVal AddInInst As Object,
custom() As Variant)
On Error GoTo AddinInstance_OnConnection_Error

    '-------------------------------------------------------
    ' Hook up to the Graphics Designer application.IAddInHookEvents
interface.
    ' It is necessary referencing the application this AddIn hooks
up to
    '-------------------------------------------------------
    Dim GDApplication As grafexe.Application
    Set GDApplication = Application
    If (Not GDApplication Is Nothing) Then
        '-----------------------------------------------------
        ' Explanation on filters ( first parameter to
AddIns.Attach() )
        '
        ' sbAddInFilterExecute : AddIn is not shown in the AddIn-
Menu
        '                       (ThisAddIn_Execute() is NOT called)
        ' sbAddInFilterNone    : AddIn is shown in the AddIn-Menu
and by
        '                       clicking on the AddIn's menu entry
ThisAddIn_Execute()
        '                        is called (see the figure below)
        '-----------------------------------------------------
        Set ThisAddIn =
GDApplication.Addins.Attach(sbAddinFilterNone, "Create Rectangle")
        Set GrafApp = GDApplication

        RegisterApplicationMenus
    End If
```

```
    Exit Sub

AddinInstance_OnConnection_Error:
    MsgBox Err.Description
End Sub
'-------------------------------------------------------------
'This method removes the Add-In from VB by event disconnect
'-------------------------------------------------------------
Private Sub AddinInstance_OnDisconnection(ByVal RemoveMode As
AddInDesignerObjects.ext_DisconnectMode, _
                                          custom() As Variant)
On Error GoTo AddinInstance_OnDisconnection_Error

    If (RemoveMode = ext_dm_UserClosed) Then
        RemoveApplicationMenus
    End If

    ' ------------------------------------------------------------
    ' Release reference to IAddInHookEvents interface - Important
    ' ------------------------------------------------------------
    Set ThisAddIn = Nothing
    Set GrafApp = Nothing
    Exit Sub

AddinInstance_OnDisconnection_Error:
    MsgBox Err.Description
End Sub
'-----------------------------------------------------------------
'This method describes the 2nd way to make AddIn functions
available in Graphics Designer
'
'By adding an application menu in Graphics Designer the menu click
events can be catched by
'the MenuItemClicked event from the application object
'-----------------------------------------------------------------
Private Sub RegisterApplicationMenus()
    Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
    Set objDocMenu = GrafApp.CustomMenus.InsertMenu(1, "DocMenu1",
"Doc_Menu_1")
    Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1,
"dmItem1_1", "My first menu entry")
    Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2,
"dmItem1_2", "My second menu entry")

    Set objMenuItem = Nothing
    Set objDocMenu = Nothing
End Sub
'----------------------------------------------------------------
'This method removes the AddIn menus available in Graphics Designer
'----------------------------------------------------------------
Private Sub RemoveApplicationMenus()
```

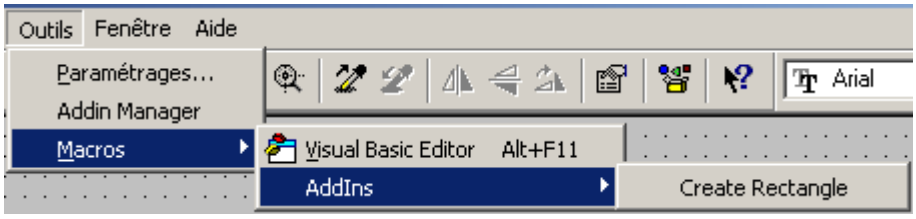
```
    Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem

    For Each objMenuItem In
GrafApp.CustomMenus("DocMenu1").MenuItems
    Set objMenuItem = Nothing
Next objMenuItem

    GrafApp.CustomMenus("DocMenu1").Delete

    Set objMenuItem = Nothing
    Set objDocMenu = Nothing
End Sub
Private Sub AddinInstance_Terminate()
    ' ---------------------------------------------------
    ' Release reference to IAddInHookEvents interface - Important
    ' ---------------------------------------------------
    Set ThisAddIn = Nothing
    Set GrafApp = Nothing
End Sub
Private Sub GrafApp_MenuItemClicked(ByVal MenuItem As
grafexe.IHMIMenuItem)

Select Case MenuItem.Key
        Case "dmItem1_1"
            TestCall1
        Case "dmItem1_2"
            TestCall2
        Case Else
            Debug.Assert False
End Select
End Sub
'-------------------------------------------------------------
'You can call both of the following procedures by clicking the menu
item in the "DocMenu1"
'-------------------------------------------------------------
Sub TestCall1()
    Call MsgBox("AddIn Menu: dmItem1_1 Clicked", vbInformation,
"GrafApp_MenuItemClicked")
End Sub
Sub TestCall2()
    Call MsgBox("AddIn Menu: dmItem1_2 Clicked", vbInformation,
"GrafApp_MenuItemClicked")
End Sub
'-------------------------------------------------------------
'Registering an AddInHook creates an object which event
'can be executed by clicking "Extras\Macros\AddIns\<Registered
Name>"
'-------------------------------------------------------------
Private Sub ThisAddIn_Execute()
    MsgBox ("AddIn : Execute! Will create a new Rectangle now")
    Dim NewShape As HMIObject
```

```
    Set NewShape =
GrafApp.ActiveDocument.HMIObjects.AddHMIObject("Rectangle1",
"HMIRectangle")
    With NewShape
.Top = 40
.Left = 40
        .BackColor = 255
End With
    MsgBox (NewShape.ObjectName)
End Sub
```

6. Générez l'AddIn et chargez-le dans Graphics Designer.

Résultat

A la prochaine ouverture, le menu personnalisé "DocMenu1" sera inséré dans Graphics Designer. Dans le menu "Outils" > "AddIns" apparaît l'option "Create Rectangle" qui permet d'insérer un rectangle dans la vue active :



Voir aussi

Pour charger un AddIn dans Graphics Designer (Page 1716)

Exemple : Création d'addins (Page 1718)

3.5 Référence VBA

3.5.1 Le modèle d'objet du Graphics Designer

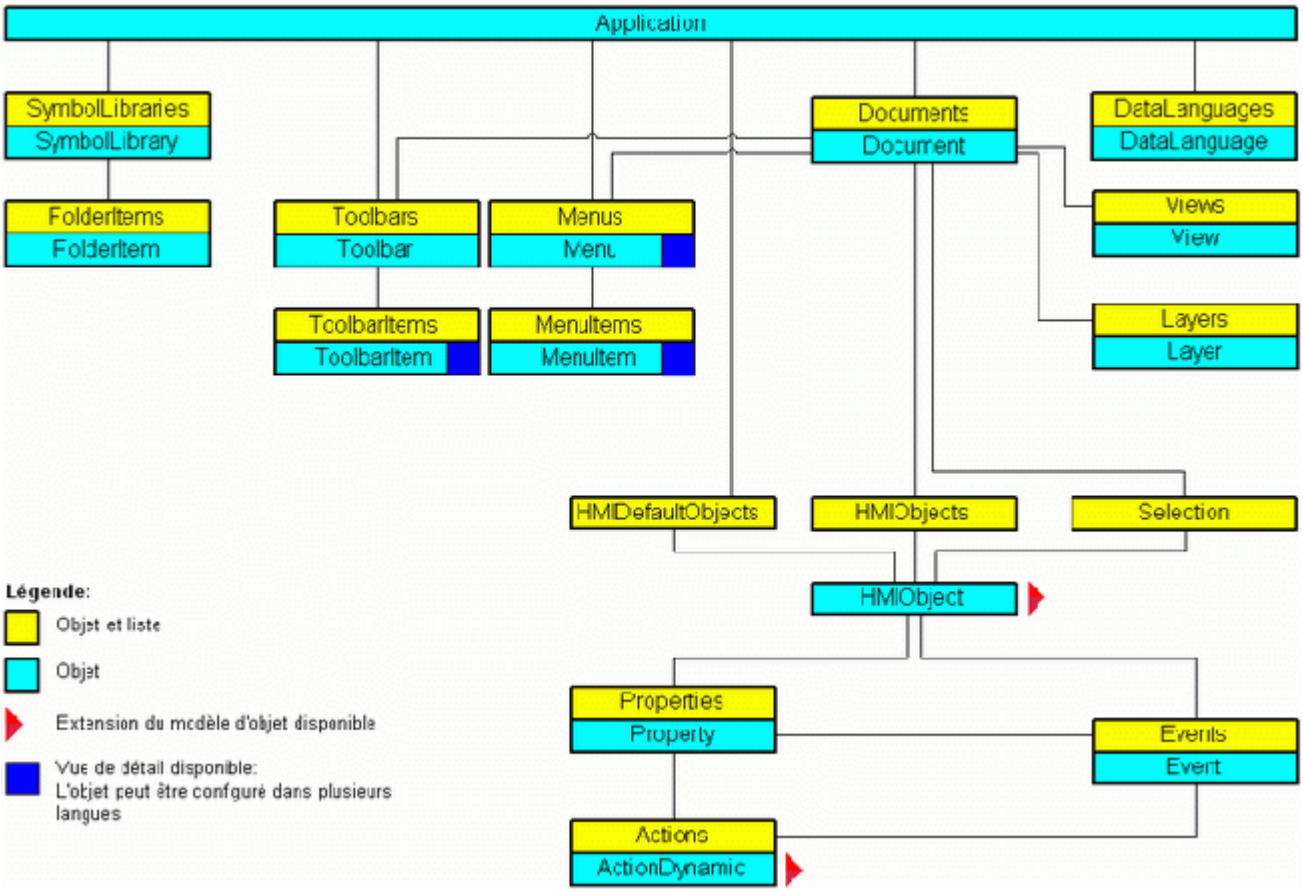
3.5.1.1 Référence VBA

Modèle d'objet VBA

Lorsque vous cliquez sur un nom d'objet, vous obtenez une description détaillée.

Remarque

Dans ce qui suit, le préfixe "HMI" est omis dans les descriptions. N'oubliez pas que, dans le code, les objets doivent être précédés du préfixe "HMI", p. ex. : "HMISymbolLibrary".



Voir aussi

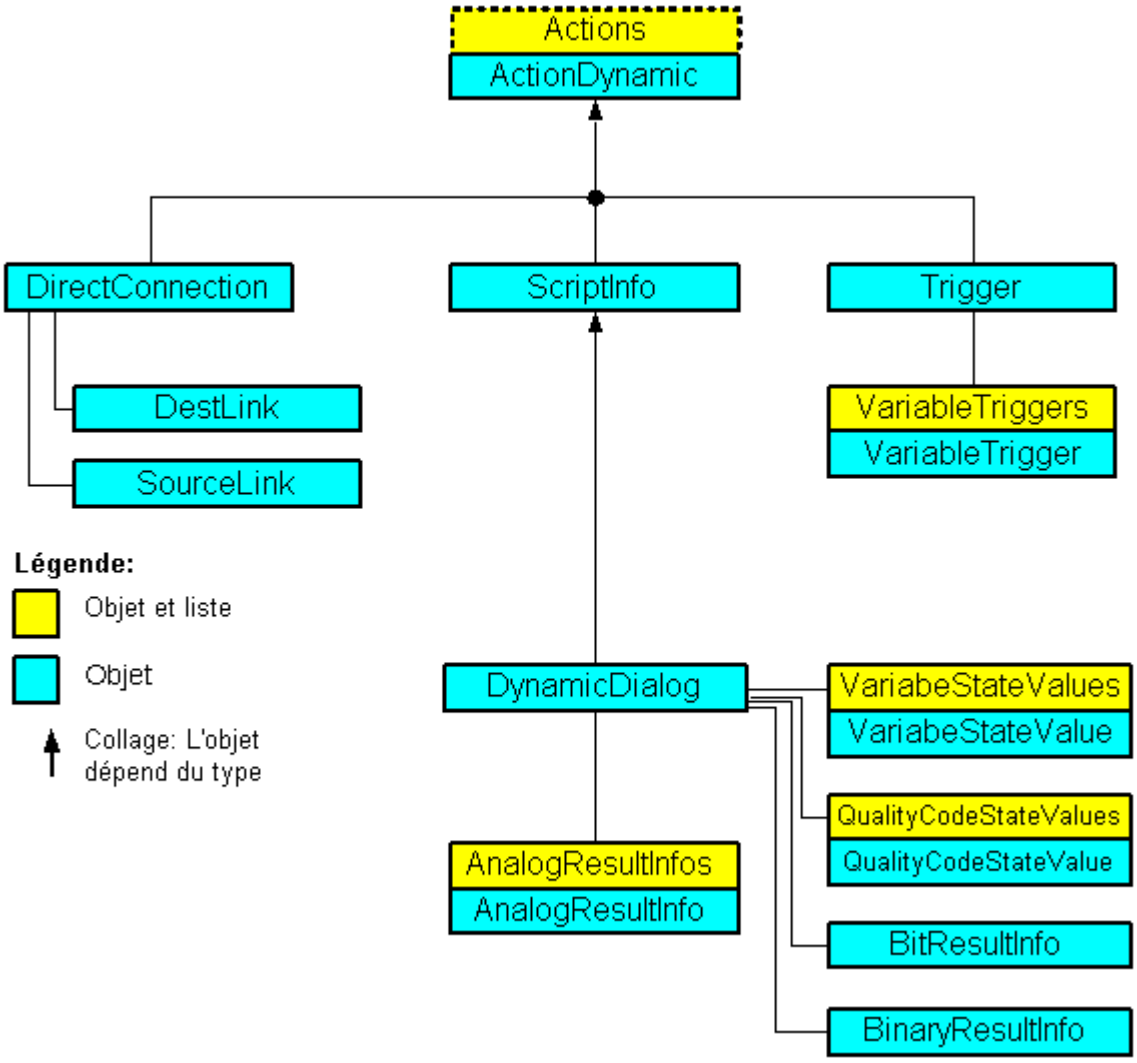
Objet Events (liste) (Page 1924)
Objet SymbolLibraries (liste) (Page 2024)
Objet Actions (liste) (Page 1872)
Objet Application (Page 1876)
Objet DataLanguage (Page 1902)
Objet DataLanguages (liste) (Page 1903)
Objet Document (Page 1908)
Objet Documents (liste) (Page 1911)
Objet Event (Page 1923)
Objet HMIDefaultObjects (liste) (Page 1939)
Objet HMIObject (Page 1943)
Objets HMIObjects (liste) (Page 1945)
Objet FolderItem (Page 1927)
Objet FolderItems (liste) (Page 1929)
Référence VBA : ActionDynamic (Page 1726)
Référence VBA : HMIObjects (Page 1728)
Référence VBA : Languages (Page 1730)
Objet Layer (Page 1955)
Objet Layers (Liste) (Page 1956)
Objet Menu (Page 1964)
Objet Menus (liste) (Page 1965)
Objet MenuItem (Page 1967)
Objet MenuItems (liste) (Page 1970)
Objet Properties (liste) (Page 1992)
Objet Toolbar (Page 2028)
Objet Toolbars (liste) (Page 2029)
Objet ToolbarItem (Page 2031)
Objet ToolbarItems (liste) (Page 2034)
Objet View (Page 2050)
Objet Views (liste) (Page 2052)
Objet Selection (liste) (Page 2010)
Objet SymbolLibrary (Page 2023)
Objet Property (Page 1993)

3.5.1.2 Référence VBA : ActionDynamic

Modèle d'objet VBA : ActionDynamic

"ActionDynamic" constitue l'interface avec les dynamiques et les actions comme les scripts, le dialogue Dynamique, la liaison directe et les déclencheurs.

Lorsque vous cliquez sur un nom d'objet, vous obtenez une description détaillée.



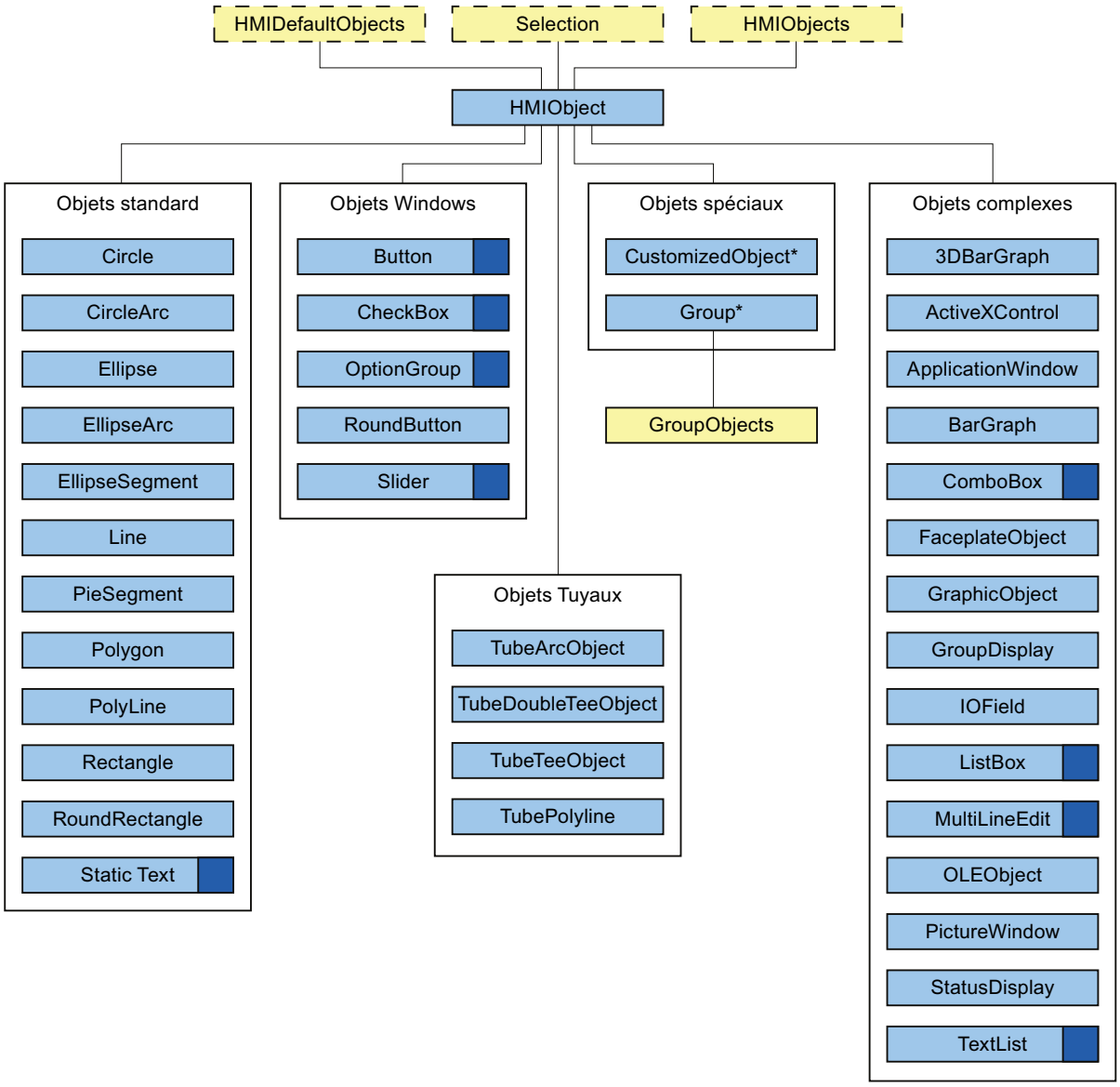
Voir aussi

Référence VBA (Page 1724)
Objet AnalogResultInfo (Page 1874)
Objet AnalogResultInfos (liste) (Page 1875)
Objet BinaryResultInfo (Page 1884)
Objet BitResultInfo (Page 1885)
Objet Actions (liste) (Page 1872)
Objet DestLink (Page 1904)
Objet DirectConnection (Page 1906)
Objet DynamicDialog (Page 1912)
Objets QualityCodeStateValue (Page 1995)
Objet QualityCodeStateValues (listage) (Page 1997)
Objet ScriptInfo (Page 2009)
Objet SourceLink (Page 2016)
Objet Trigger (Page 2035)
Objet VariableStateValue (Page 2045)
Objet VariableStateValues (liste) (Page 2046)
Objet VariableTrigger (Page 2048)
Objet VariableTriggers (liste) (Page 2049)

3.5.1.3 Référence VBA : HMIObjects

Modèle d'objet VBA : HMIObjects

Lorsque vous cliquez sur un nom d'objet, vous obtenez une description détaillée.



Objet et liste

Objet

Vue de détail disponible.
L'objet peut être configuré en plusieurs langues.

* Non contenu dans la liste DefaultObjects.

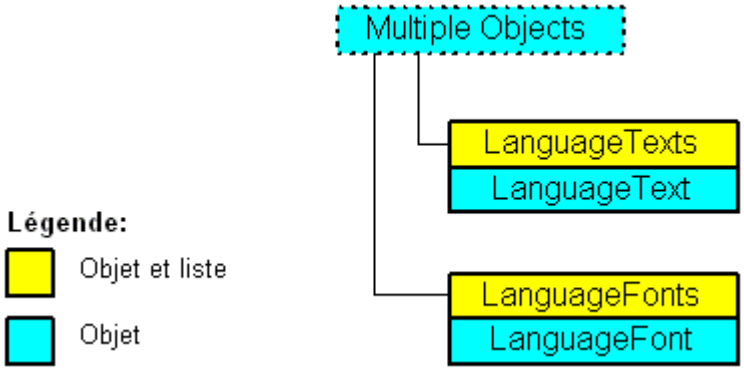
Voir aussi

Référence VBA (Page 1724)
Objet PolyLine (Page 1989)
Objet GroupDisplay (Page 1935)
Objet 3DBarGraph (Page 1867)
Objet ActiveXControl (Page 1873)
Objet ApplicationWindow (Page 1879)
Objet Button (Page 1886)
Objet CheckBox (Page 1889)
Objet Circle (Page 1890)
Objet CircularArc (Page 1893)
Objet Line (Page 1958)
Objet OLEObject (Page 1975)
Objet OptionGroup (Page 1977)
Objet PictureWindow (Page 1980)
Objet PieSegment (Page 1983)
Objet Polygon (Page 1986)
Objet Property (Page 1993)
Objet Rectangle (Page 2000)
Objet RoundButton (Page 2003)
Objet RoundRectangle (Page 2006)
Objet Slider (Page 2013)
Objet StaticText (Page 2017)
Objet StatusDisplay (Page 2020)
Objet TextList (Page 2025)
Objet Ellipse (Page 1914)
Objet EllipseArc (Page 1917)
Objet EllipseSegment (Page 1920)
Objet GraphicObject (Page 1931)
Objet Group (Page 1934)
Objet HMIDefaultObjects (liste) (Page 1939)
Objet HMIObject (Page 1943)
Objets HMIObjects (liste) (Page 1945)
Objet IOField (Page 1947)
Objet BarGraph (Page 1881)
Objet GroupedObjects (liste) (Page 1938)
Référence VBA : Languages (Page 1730)
Objet SelectionA (liste) (Page 2010)
Objet CustomizedObject (Page 1900)
Objet FaceplateObject (Page 1926)

3.5.1.4 Référence VBA : Languages

Modèle d'objet VBA : Languages

Lorsque vous cliquez sur un nom d'objet, vous obtenez une description détaillée.



Voir aussi

Référence VBA (Page 1724)
Objet LanguageFont (Page 1950)
Objet LanguageFonts (liste) (Page 1951)
Objet LanguageText (Page 1953)
Objet LanguageTexts (liste) (Page 1954)

3.5.1.5 Evénements

A-D

Evénement Activated

Description

Intervient à l'activation d'une vue dans Graphics Designer, p. ex. lors d'un basculement entre deux images.

Syntaxe

```
Document_Activated(CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré à l'activation de la vue :

```
Private Sub Document_Activated(CancelForwarding As Boolean)
'VBA76
MsgBox "The document got the focus." & vbCrLf &_
"This event (Document_Activated) is raised by the document itself"
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Gestion des événements (Page 1704)

Evénement BeforeClose

Description

Intervient immédiatement avant la fermeture d'une vue.

Syntaxe

```
Document_BeforeClose(Cancel As Boolean, CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré avant la fermeture de la vue :

```
Private Sub Document_BeforeClose(Cancel As Boolean, CancelForwarding As Boolean)
'VBA77
MsgBox "Event Document_BeforeClose is raised"
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeDocumentClose

Description

Intervient immédiatement avant la fermeture d'une vue.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_BeforeDocumentClose(Document As HMIDocument, Cancel
As Boolean)
```

Paramètres

Paramètre (type de données)	Description
Document (HMIDocument)	La vue qui va être fermée.
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, un message est délivré avant la fermeture de la vue :

```
Private Sub objGDApplication_BeforeDocumentClose(ByVal Document As IHMIDocument, Cancel As
Boolean)
'VBA78
MsgBox "The document " & Document.Name & " will be closed after press ok"
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeDocumentSave

Description

Intervient immédiatement avant l'enregistrement d'une vue.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_BeforeDocumentSave(Document As HMIDocument, Cancel
As Boolean)
```

Paramètres

Paramètre (type de données)	Description
Document (HMIDocument)	La vue qui va être fermée.
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, un message est délivré avant la fermeture de la vue :

```
Private Sub objGDApplication_BeforeDocumentSave(ByVal Document As IHMIDocument, Cancel As
Boolean)
'VBA79
MsgBox Document.Name & "-saving will start after press ok."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeHMIObjectDelete

Description

Intervient immédiatement avant la suppression d'un objet d'une vue.

Syntaxe

```
BeforeHMIObjectDelete(ByVal HMIObject As IHMIObject, Cancel As
Boolean, CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
HMIObject (IHMIObject)	Identifie l'objet à supprimer.
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message apparaît indiquant l'objet qui va être supprimé :

```
Private Sub Document_BeforeHMIObjectDelete(ByVal HMIObject As IHMIObject, Cancel As
Boolean, CancelForwarding As Boolean)
'VBA80
Dim strObjName As String
Dim strAnswer As String
'
'"strObjName" contains the name of the deleted object
strObjName = HMIObject.ObjectName
strAnswer = MsgBox("Are you sure to delete " & strObjName & "?", vbYesNo)
If strAnswer = vbNo Then
'if pressed "No" -> set Cancel to true for prevent delete
Cancel = True
    End If
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeLibraryFolderDelete

Description

Intervient immédiatement avant la suppression d'un dossier de la bibliothèque des modules.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_BeforeLibraryFolderDelete(LibObject As
HMIFolderItem, Cancel As Boolean)
```

Paramètres (option)

Paramètre (type de données)	Description
LibObject (HMIFolderItem)	Le dossier qui va être supprimé.
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple suivant, une alarme est émise avant qu'un dossier soit effacé de la bibliothèque des modules :

```
Private Sub objGDApplication_BeforeLibraryFolderDelete(ByVal LibObject As HMIFolderItem,
Cancel As Boolean)
'VBA81
MsgBox "The library-folder " & LibObject.Name & " will be delete..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeLibraryObjectDelete

Description

Intervient immédiatement avant la suppression d'un objet de la bibliothèque des modules.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_BeforeLibraryObjectDelete(LibObject As
HMIFolderItem, Cancel As Boolean)
```

Paramètres (option)

Paramètre (type de données)	Description
LibObject (HMIFolderItem)	L'objet qui va être supprimé.
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple suivant, une alarme est émise avant qu'un dossier soit effacé de la bibliothèque des modules :

```
Private Sub objGDApplication_BeforeLibraryObjectDelete(ByVal LibObject As HMIFolderItem,
Cancel As Boolean)
'VBA82
MsgBox "The object " & LibObject.Name & " will be delete..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeQuit

Description

Intervient immédiatement avant la fermeture de Graphics Designer.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_BeforeQuit(Cancel As Boolean)
```

Paramètres

Paramètre (type de données)	Description
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple suivant, une alarme est émise peu avant que Graphics Designer soit fermé.

```
Private Sub objGDApplication_BeforeQuit(Cancel As Boolean)
'VBA83
MsgBox "The Graphics Designer will be shut down"
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeSave

Description

Intervient immédiatement avant l'enregistrement d'une vue.

Syntaxe

```
Document_BeforeSave(Cancel As Boolean, CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré avant la fermeture de la vue :

```
Private Sub Document_BeforeSave(Cancel As Boolean, CancelForwarding As Boolean)
'VBA84
MsgBox "The document will be saved..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement BeforeVisibleFalse

Description

Intervient immédiatement avant le passage de l'application Graphics Designer de visible à invisible.

Syntaxe

```
Document_BeforeVisibleFalse(Cancel As Boolean, CancelForwarding As
Boolean)
```

Paramètres

Paramètre (type de données)	Description
Cancel (Boolean)	TRUE si l'exécution de la commande doit être annulée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Réglage par défaut : "False".

Exemple

--

Voir aussi

Référence VBA (Page 1724)

Evénement ConnectionEvent

Description

Intervient lorsque deux objets sont liés par le connecteur.

Syntaxe

```
ConnectionEvent(eConnEventType, HMIConnector, HMIConnectedObject,
CancelProcess, CancelForwarding)
```

Paramètres (option)

Paramètre (type de données)	Description
eConnEventType (HMIConnectionEventType)	--
HMIConnector (HMIObject)	--
HMIConnectedObject (HMIObject)	--
CancelProcess (Boolean)	TRUE si l'exécution de la commande doit être annulée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

--

Voir aussi

Référence VBA (Page 1724)

Evénement DataLanguageChanged

Description

Intervient lors d'un changement de la langue de travail.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_DataLanguageChanged(lCID As Long)
```

Paramètres

Paramètre (type de données)	Description
lCID (Long)	L'identifiant de la langue de travail

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, la nouvelle langue de travail configurée est appliquée :

```
Private Sub objGDApplication_DataLanguageChanged(ByVal lCID As Long)
```

```
'VBA87
MsgBox "The datalanguage is changed to " & Application.CurrentDataLanguage & "."
End Sub
```

Voir aussi

Configuration multilingue avec VBA (Page 1617)

Référence VBA (Page 1724)

Evénement DesktopLanguageChanged

Description

Intervient lors d'un changement de la langue d'interface.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_DesktopLanguageChanged(lCID As Long)
```

Paramètres

Paramètre (type de données)	Description
lCID (Long)	L'identifiant de la langue d'interface

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, la nouvelle langue d'interface configurée est appliquée :

```
Private Sub objGDApplication_DesktopLanguageChanged(ByVal lCID As Long)
'VBA88
MsgBox "The desktop-language is changed to " & Application.CurrentDesktopLanguage & "."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Configuration multilingue avec VBA (Page 1617)

Evénement DocumentActivated

Description

Intervient à l'activation d'une vue dans Graphics Designer, p. ex. lors d'un basculement entre deux images.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_DocumentActivated(Document As HMIDocument)
```

Paramètres

Paramètre (type de données)	Description
Document (HMIDocument)	L'image activée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, un message est délivré indiquant la vue qui a été activée :

```
Private Sub objGDApplication_DocumentActivated(ByVal Document As IHMIDocument)
'VBA89
MsgBox "The document " & Document.Name & " got the focus." & vbCrLf &_
"This event is raised by the application."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement DocumentCreated

Description

Intervient à la création d'une nouvelle vue dans Graphics Designer.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_DocumentCreated(Document As HMIDocument)
```

Paramètres

Paramètre (type de données)	Description
Document (HMIDocument)	La vue créée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

dans l'exemple suivant, le nom de la nouvelle vue est appliqué :

```
Private Sub objGDApplication_DocumentCreated(ByVal Document As IHMIDocument)
'VBA90
MsgBox Document.Name & " will be created."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement DocumentOpened

Description

Intervient à l'ouverture d'une vue.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_DocumentOpened(Document As HMIDocument)
```

Paramètres

Paramètre (type de données)	Description
Document (HMIDocument)	La vue qui a été ouverte.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, un message est délivré indiquant la vue qui a été ouverte :

```
Private Sub objGDApplication_DocumentOpened(ByVal Document As IHMIDocument)
'VBA91
MsgBox Document.Name & " is opened."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement DocumentSaved

Description

Intervient à l'enregistrement d'une vue dans Graphics Designer.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_DocumentSaved(Document As HMIDocument)
```

Paramètres

Paramètre (type de données)	Description
Document (HMIDocument)	La vue qui a été enregistrée.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, un message est délivré indiquant la vue qui a été enregistrée :

```
Private Sub objGDApplication_DocumentSaved(ByVal Document As IHMIDocument)
'VBA92
MsgBox Document.Name & " is saved."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement DocumentPropertyChanged

Description

Intervient à la modification d'une propriété de vue.

Syntaxe

```
Document_DocumentPropertyChanged(ByVal Property As IHMIProperty,
CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
Property (IHMIProperty)	Identifie la propriété qui a été modifiée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple suivant, un message est délivré indiquant la propriété de vue qui a été modifiée :

```
Private Sub Document_DocumentPropertyChanged(ByVal Property As IHMIProperty,
CancelForwarding As Boolean)
'VBA93
Dim strPropName As String
'"strPropName" contains the name of the modified property
strPropName = Property.Name
MsgBox "The picture-property " & strPropName & " is modified..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

F-Z

Evénement HMIObjectAdded

Description

Intervient à l'insertion d'un objet.

Syntaxe

```
Document_HMIObjectAdded(ByVal HMIObject As IHMIObject,
CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
HMIObject (IHMIObject)	Identifie l'objet qui a été inséré.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré indiquant l'objet qui a été inséré :

```
Private Sub Document_HMIObjectAdded(ByVal HMIObject As IHMIObject, CancelForwarding As
Boolean)
'VBA94
Dim strObjName As String
'
'"strObjName" contains the name of the added object
strObjName = HMIObject.ObjectName
MsgBox "Object " & strObjName & " is added..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement HMIObjectMoved

Description

Intervient au déplacement d'un objet.

Syntaxe

```
Document_HMIObjectMoved(ByVal HMIObject As IHMIObject,
CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
HMIObject (IHMIObject)	Identifie l'objet qui a été déplacé.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré indiquant l'objet qui a été déplacé :

```
Private Sub Document_HMIObjectMoved(ByVal HMIObject As IHMIObject, CancelForwarding As
Boolean)
'VBA95
Dim strObjName As String
'
'"strObjName" contains the name of the moved object
strObjName = HMIObject.ObjectName
MsgBox "Object " & strObjName & " was moved..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement HMIObjectPropertyChanged

Description

Intervient à la modification d'une propriété d'objet.

Syntaxe

```
Document_HMIObjectPropertyChanged(ByVal Property As IHMIProperty,
CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
Property (IHMIProperty)	Identifie la propriété qui a été modifiée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple suivant, un message est délivré indiquant la propriété d'objet qui a été modifiée :

```
Private Sub Document_HMIObjectPropertyChanged(ByVal Property As IHMIProperty,
CancelForwarding As Boolean)
'VBA96
Dim strObjProp As String
Dim strObjName As String
Dim varPropValue As Variant
'
'"strObjProp" contains the name of the modified property
'"varPropValue" contains the new value
strObjProp = Property.Name
varPropValue = Property.value
'
'"strObjName" contains the name of the selected object,
'which property is modified
strObjName = Property.Application.ActiveDocument.Selection(1).ObjectName
MsgBox "The property " & strObjProp & " of object " & strObjName & " is modified... " &
vbCrLf & "The new value is: " & varPropValue
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement HMIObjectResized

Description

Intervient à la modification de la taille d'un objet.

Syntaxe

```
Document_HMIObjectResized(ByVal HMIObject As IHMIObject,
CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
HMIObject (IHMIObject)	Identifie l'objet dont la taille a été modifiée.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple suivant, un message est délivré indiquant que la taille d'un objet a été modifiée :

```
Private Sub Document_HMIObjectResized(ByVal HMIObject As IHMIObject, CancelForwarding As
Boolean)
'VBA97
Dim strObjName As String
'
'"strObjName" contains the name of the modified object
strObjName = HMIObject.ObjectName
MsgBox "The size of " & strObjName & " was modified..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement LibraryFolderRenamed

Description

Intervient au changement de nom d'un dossier de la bibliothèque des modules.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_LibraryFolderRenamed(LibObject As HMIFolderItem,
OldName As String)
```

Paramètres

Paramètre (type de données)	Description
LibObject (HMIFolderItem)	Le dossier dont le nom a été changé.
OldName (String)	Le nom du dossier avant changement.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple suivant, l'ancien et le nouveau nom du dossier sont édités :

```
Private Sub objGDApplication_LibraryFolderRenamed(ByVal LibObject As HMIFolderItem, ByVal
OldName As String)
'VBA98
MsgBox "The Library-folder " & OldName & " is renamed in: " & LibObject.DisplayName
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Evénement LibraryObjectRenamed

Description

Intervient au changement de nom d'un objet de la bibliothèque des modules.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_LibraryObjectRenamed(LibObject As HMIFolderItem,
OldName As String)
```

Paramètres

Paramètre (type de données)	Description
LibObject (HMIFolderItem)	L'objet dont le nom a été changé.
OldName (String)	Le nom de l'objet avant changement.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple suivant, l'ancien et le nouveau nom de l'objet sont édités :

```
Private Sub objGDApplication_LibraryObjectRenamed(ByVal LibObject As IHMIFolderItem, ByVal
OldName As String)
'VBA99
MsgBox "The object " & OldName & " is renamed in: " & LibObject.DisplayName
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Evénement LibraryObjectAdded

Description

Intervient à l'insertion d'un objet dans la bibliothèque des modules.

Syntaxe

```
HMIObjectPropertyChanged(ByVal Property As IHMIProperty,
CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
LibObject (IHMIFolderItem)	Identifie l'objet Library.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré lorsque l'objet a été inséré dans la bibliothèque des modules :

```
Private Sub Document_LibraryObjectAdded(ByVal LibObject As IHMIFolderItem,
CancelForwarding As Boolean)
'VBA100
Dim strObjName As String
'
'"strObjName" contains the name of the added object
strObjName = LibObject.DisplayName
MsgBox "Object " & strObjName & " was added to the picture."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement MenuItemClicked

Description

Intervient lors d'un clic sur une option d'un menu personnalisé.

Remarque

Cet événement est spécifique à une application et spécifique à un document.

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Syntaxe

```
Document_MenuItemClicked(ByVal MenuItem As IHMIMenuItem)
```

Paramètres

Paramètre (type de données)	Description
MenuItem (IHMIMenuItem)	Identifie le menu personnalisé.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, un message est délivré lors d'un clic sur la première option d'un menu personnalisé :

```
Private Sub Document_MenuItemClicked(ByVal MenuItem As IHMIMenuItem)
'VBA101
Dim objMenuItem As HMIMenuItem
Dim varMenuItemKey As Variant
Set objMenuItem = MenuItem
'
'"objMenuItem" contains the clicked menu-item
'"varMenuItemKey" contains the value of parameter "Key"
'from the clicked userdefined menu-item
```

```
varMenuItemKey = objMenuItem.Key
Select Case MenuItem.Key
Case "mItem1_1"
MsgBox "The first menu-item was clicked!"
End Select
End Sub
```

Voir aussi

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Référence VBA (Page 1724)

Evénement NewLibraryFolder

Description

Intervient à la création d'un dossier dans la bibliothèque des modules.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_NewLibraryFolder(LibObject As HMIFolderItem)
```

Paramètres

Paramètre (type de données)	Description
LibObject (HMIFolderItem)	Le nouveau dossier créé.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple suivant, le nom du nouveau dossier est appliqué :

```
Private Sub objGDApplication_NewLibraryFolder(ByVal LibObject As IHMIFolderItem)
'VBA102
MsgBox "The library-folder " & LibObject.DisplayName & " was added."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Evénement NewLibraryObject

Description

Intervient à la création d'un objet dans la bibliothèque des modules.

Syntaxe

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

```
objGDApplication_NewLibraryObject(LibObject As HMIFolderItem)
```

Paramètres

Paramètre (type de données)	Description
LibObject (HMIFolderItem)	Le nouvel objet créé.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
```

```
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple suivant, le nom du nouvel objet est appliqué :

```
Private Sub objGDApplication_NewLibraryObject(ByVal LibObject As IHMIFolderItem)
'VBA103
MsgBox "The object " & LibObject.DisplayName & " was added."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Evénement Opened

Description

Intervient à l'ouverture d'une vue.

Syntaxe

```
Document_Opened(CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré à l'ouverture de la vue :

```
Private Sub Document_Opened(CancelForwarding As Boolean)
'VBA104
MsgBox "The Document is open now..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement Saved

Description

Intervient après enregistrement d'une vue.

Syntaxe

```
Document_Saved(CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message est délivré indiquant que la vue a été enregistrée :

```
Private Sub Document_Saved(CancelForwarding As Boolean)
'VBA105
MsgBox "The document is saved..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement SelectionChanged

Description

Intervient à un changement de sélection.

Syntaxe

```
Document_SelectionChanged(CancelForwarding As Boolean)
```

Paramètres

Paramètre (type de données)	Description
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Dans l'exemple qui suit, un message apparaît à la sélection d'un nouvel objet :

```
Private Sub Document_SelectionChanged(CancelForwarding As Boolean)
'VBA106
MsgBox "The selection is changed..."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement Started

Description

Intervient au démarrage de Graphics Designer.

Syntaxe

```
objGDApplication_Started()
```

Remarque

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Dans l'exemple qui suit, la désignation "objGDApplication" est utilisée pour <Name>.

Paramètres

--

Exemple

Déclarer une application.

```
Dim WithEvents objGDApplication As grafexe.Application
```

Poser la variable d'événement.

```
Private Sub Document_Opened(CancelForwarding As Boolean)
 Set objGDApplication = Me.Application
End Sub
```

Interroger l'événement "Started" et éditer l'alarme.

```
Private Sub objGDApplication_Started()
'VBA107
'This event is raised before objGDApplication_Started()
 MsgBox "The Graphics Designer is started!"
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement ToolbarItemClicked

Description

Intervient lors d'un clic sur une icône dans une barre d'outils personnalisée.

Remarque

Cet événement est spécifique à une application et spécifique à un document.

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Syntaxe

```
Document_ToolbarItemClicked(ByVal ToolbarItem As IHMIToolbarItem)
```

Paramètres

Paramètre (type de données)	Description
ToolbarItem (IHMIToolbarItem)	Identifie l'icône.

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, un message est délivré lors d'un clic sur la première icône personnalisée :

```
Private Sub Document_ToolbarItemClicked(ByVal ToolbarItem As IHMIToolbarItem)
'VBA108
Dim objToolbarItem As HMIToolbarItem
Dim varToolbarItemKey As Variant
Set objToolbarItem = ToolbarItem
'
'"varToolbarItemKey" contains the value of parameter "Key"
'from the clicked userdefined toolbar-item
varToolbarItemKey = objToolbarItem.Key
'
Select Case varToolbarItemKey
Case "tItem1_1"
MsgBox "The first Toolbar-Icon was clicked!"
End Select
End Sub
```

Voir aussi

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Référence VBA (Page 1724)

Evénement ViewCreated

Description

Intervient à la copie d'une vue.

Remarque

Cet événement est spécifique à une application et spécifique à un document.

Pour que l'événement spécifique de l'application soit disponible dans le projet, l'application Graphics Designer doit être déclarée. A cet effet, on utilise l'instruction suivante :

Dim WithEvents <Name> As grafexe.Application

Syntaxe

```
Document_ViewCreated(ByVal pView As IHMIView, CancelForwarding As
Boolean)
```

Paramètres

Paramètre (type de données)	Description
pView (IHMIView)	Identifie la copie de la vue.
CancelForwarding (Boolean)	TRUE si l'événement ne doit pas être transmis. Le réglage par défaut est "False".

Exemple

Pour que l'exemple ci-dessous s'applique, vous devez utiliser la procédure suivante :

```
Private Sub SetApplication()
'This procedure have to execute with "F5" first
Set objGDApplication = grafexe.Application
End Sub
```

Dans l'exemple qui suit, le nombre de copies de la vue effectuées est sorti après la création de la première copie.

```
Private Sub Document_ViewCreated(ByVal pView As IHMIView, CancelForwarding As Boolean)
'VBA109
Dim iViewCount As Integer
'
'To read out the number of views
iViewCount = pView.Application.ActiveDocument.Views.Count
```

```
MsgBox "A new copy of the picture (number " & iViewCount & ") was created."
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Evénement WindowStateChange

Description

Intervient à la modification de la taille d'une fenêtre (p.ex. de "mini" à "maxi").

Syntaxe

```
objGDApplication_WindowStateChanged()
```

Paramètres (option)

--

Exemple

Dans l'exemple suivant, un message est délivré lorsque la taille de la fenêtre est modifiée.

```
Private Sub objGDApplication_WindowStateChanged()
'VBA110
MsgBox "The state of the application-window is changed!"
End Sub
```

Voir aussi

Référence VBA (Page 1724)

3.5.1.6 Méthodes

A-C

Méthode Activate

Description

Active l'objet indiqué.

Syntaxe

Expression.`Activate()`

Expression

Impératif. Une expression qui renvoie un objet du type "Application" ou "View".

Paramètres

--

Exemple

Dans l'exemple qui suit, la vue active est copiée et la copie est ensuite activée :

```
Sub CreateAndActivateView()
'VBA111
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
End Sub
```

Voir aussi

Objet View (Page 2050)
Objet Application (Page 1876)
Référence VBA (Page 1724)

Méthode Add

Description

Ajoute un élément à un listage.

Les listages auxquels peut être appliquée la méthode Add sont indiqués dans le tableau qui suit. Les paramètres et la syntaxe de la méthode Add sont donnés à la rubrique "Méthodes".

Listage	Effet de la méthode Add
Listage AnalogResultInfos	Ajoute au dialogue Dynamique une nouvelle plage de valeurs analogiques.
Listage Documents	Crée une nouvelle vue dans Graphics Designer.
Listage GroupedObjects	Ajoute un nouvel objet à un objet-groupe.
Listage Toolbars	Crée une nouvelle barre d'outils personnalisée.
Listage VariableTriggers	Crée un nouveau déclencheur de variable.
Listage Views	Crée une copie de la vue indiquée.

Voir aussi

Méthode Add (listage Views) (Page 1773)

Méthode Add (listage VariableTriggers) (Page 1772)

Méthode Add (listage CustomToolbars) (Page 1768)

Méthode Add (listage GroupedObjects) (Page 1770)

Méthode Add (listage Documents) (Page 1769)

Méthode Add (listage AnalogResultInfos) (Page 1766)

Méthode Add (listage AnalogResultInfos)

Description

Ajoute au dialogue Dynamique une nouvelle plage de valeurs analogiques.

Syntaxe

Expression.`Add(RangeTo, ResultValue)`

Expression

Impératif. Une expression qui renvoie un objet du type "AnalogResultInfos".

Paramètres

Paramètre (type de données)	Description
RangeTo (Variant)	La plage de valeurs qui provoque la modification de la propriété.
ResultValue (Variant)	La valeur affectée à la propriété d'objet lorsque la plage de valeurs est atteinte.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. On indique un nom de variable et on définit trois plages de valeurs analogiques :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA112
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.Add 50, 40
.AnalogResultInfos.Add 100, 80
.AnalogResultInfos.ElseCase = 100
End With
End Sub
```

La figure représente le dialogue Dynamique après exécution de la procédure :

Voir aussi

Objet DynamicDialog (Page 1912)

Objet AnalogResultInfos (liste) (Page 1875)

Méthode CreateDynamic (Page 1803)

Pour dynamiser une propriété par le dialogue Dynamique (Page 1685)

Méthode Add (listage CustomToolbars)

Description

Crée une nouvelle barre d'outils personnalisée. On distingue les barres d'outils personnalisées spécifiques d'une application et celles spécifiques d'une vue :

- Barre d'outils spécifique à une application : Est liée à Graphics Designer et reste visible lorsque toutes les vues dans Graphics Designer sont fermées. Placez le code VBA dans le document "GlobalTemplateDocument" ou "ProjectTemplateDocument" et utilisez la propriété "Application".
- Barre d'outils spécifique à une vue : est lié à une vue et n'est visible que tant que cette vue est active. Placez le code VBA dans le document "ThisDocument" de la vue considérée et utilisez la propriété "ActiveDocument".

Syntaxe

Expression.`Add(Key)`

Expression

Impératif. Une expression qui renvoie un objet du type "CustomToolbars".

Paramètres

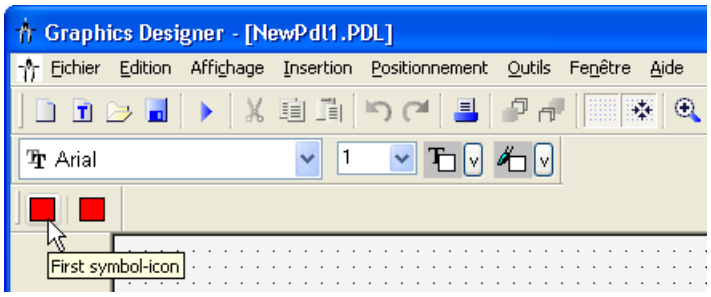
Paramètre (type de données)	Description
Key (Variant)	Identifie la barre d'outils personnalisée. Pour "Key", utilisez des désignations univoques (p. ex. "DocToolbar1").

Exemple

Dans l'exemple qui suit, on crée dans la vue active une barre d'outils personnalisée à deux icônes séparées par une ligne de séparation :

```
Sub AddDocumentSpecificCustomToolbar()
'VBA115
Dim objToolbar As HMIToolbar
Dim objToolbarItem As HMIToolbarItem

Set objToolbar = ActiveDocument.CustomToolbars.Add("DocToolbar")

'Add toolbar-items to the userdefined toolbar
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(1, "tItem1_1", "My first
Symbol-Icon")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(3, "tItem1_3", "My second
Symbol-Icon")
'
'Insert seperatorline between the two tollbaritems
```

```
Set objToolbarItem = objToolbar.ToolbarItems.InsertSeparator(2, "tSeparator1_2")
End Sub
```



Voir aussi

Objet Toolbars (liste) (Page 2029)

Méthode InsertToolbarItem (Page 1831)

Méthode InsertSeparator (Page 1828)

Méthode InsertFromMenuItem (Page 1823)

Référence VBA (Page 1724)

Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode Add (listage Documents)

Description

Crée une nouvelle vue dans Graphics Designer.

Syntaxe

Expression.`Add [HMIOpenDocumentType]`

Expression

Impératif. Une expression qui renvoie un objet du type "Documents".

Paramètres

Paramètre (type de données)	Description
HMIOpenDocumentType (HMIDocumentType)	Définit le mode d'ouverture de la vue : • HMIDocumentTypeVisible : ouvre la vue pour traitement direct. Si vous n'indiquez pas le paramètre, ceci est le mode par défaut. • HMIDocumentTypeInvisible : ouvre la vue en invisible, c'est-à-dire qu'elle ne sera pas affichée dans Graphics Designer. Vous ne pouvez accéder à la vue que par l'intermédiaire du listage Documents et vous pouvez la régler en visible par la propriété Hide.

Exemple

Dans l'exemple qui suit, on crée une nouvelle vue dans Graphics Designer :

```
Sub AddNewDocument()
'VBA113
Application.Documents.Add hmiOpenDocumentTypeVisible
End Sub
```

Voir aussi

Propriété Hide (Page 2202)

Objet Documents (liste) (Page 1911)

Référence VBA (Page 1724)

Méthode Add (listage GroupedObjects)

Description

Ajoute un objet existant à l'objet-groupe indiqué.

Syntaxe

Expression.`Add(Index)`

Expression

Impératif. Une expression qui renvoie un objet du type "GroupedObjects".

Paramètres

Paramètre (type de données)	Description
Indice (Variant)	L'objet à ajouter. Vous pouvez utiliser le numéro d'index ou le nom de l'objet.

Exemple

Dans cet exemple, on crée l'objet groupe "MyGroup" à partir d'objets. On ajoute ensuite un segment d'ellipse à l'objet groupe :

```
Sub CreateGroup()
'VBA114
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipseSegment As HMIEllipseSegment
Dim objGroup As HMIGroup

Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With

MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup

'Set name for new group-object
'The name identifies the group-object
objGroup.ObjectName = "My Group"

'Add new object to active document...
Set objEllipseSegment = ActiveDocument.HMIObjects.AddHMIObject("EllipseSegment",
"HMIEllipseSegment")
Set objGroup = ActiveDocument.HMIObjects("My Group")

'...and add it to the group:
objGroup.GroupedHMIObjects.Add ("EllipseSegment")
End Sub
```

Voir aussi

Objet GroupedObjects (liste) (Page 1938)

Méthode Add (listage VariableTriggers)

Description

Crée un nouveau déclencheur de variable.

Syntaxe

Expression.`Add(VarName, Type)`

Expression

Impératif. Une expression qui renvoie un objet du type "VariableTriggers".

Paramètres

Paramètre (type de données)	Description
VarName (String)	Le nom de la variable qui doit être utilisée comme déclencheur. Notez bien que vous devez créer la variable dans le dialogue de sélection de variable.
Type (CycleType)	Le type de cycle, à sélectionner dans une liste de l'Editeur VBA si vous utilisez cette méthode.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé par un déclenchement de variable :

```
Sub DynamicWithVariableTriggerCycle()
'VBA69
Dim objVBScript As HMIScriptInfo
Dim objVarTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_VariableTrigger",
"HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
With objVBScript
Set objVarTrigger = .Trigger.VariableTriggers.Add("VarTrigger", hmiVariableCycleType_10s)
.SourceCode = ""
End With
End Sub
```

Voir aussi

Objet VariableTriggers (liste) (Page 2049)

Référence VBA (Page 1724)

Méthode Add (listage Views)

Description

Crée une copie de la vue indiquée.

Syntaxe

Expression.`Add()`

Expression

Impératif. Une expression qui renvoie un objet du type "Views".

Paramètres

--

Exemple

Dans l'exemple qui suit, la vue active est copiée et la copie est ensuite activée :

```
Sub CreateViewAndActivateView()
'VBA117
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
End Sub
```

Voir aussi

Objet Views (liste) (Page 2052)

Référence VBA (Page 1724)

Méthode AddAction

Description

Configure une action liée à un objet ou à une propriété qui sera déclenchée à l'occurrence d'un événement déterminé.

Syntaxe

Expression.`Methode(HMIActionCreationType)`

Expression

Impératif. Une expression qui renvoie un objet du type "Actions".

Paramètres

Paramètre (type de données)	Description
HMIActionCreationType (Variant)	Définit l'action : • hmiActionCreationTypeCScript : Configure une action en C • hmiActionCreationTypeVBScript : Configure une action en VBS • hmiActionCreationTypeDirectConnection : Configure une liaison directe

Exemple

Dans l'exemple qui suit, on configure une action VBS liée à la modification du rayon d'un cercle :

```
Sub AddActionToPropertyTypeVBScript()
'VBA118
Dim objEvent As HMIEvent
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle
'Create circle in picture. By changing of property "Radius"
'a VBS-action will be started:
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB", "HMICircle")
Set objEvent = objCircle.Radius.Events(1)
Set objVBScript = objEvent.Actions.AddAction(hmiActionCreationTypeVBScript)
End Sub
```

Voir aussi

Objet Event (Page 1923)

Objet Actions (liste) (Page 1872)

Méthode AddActiveXControl

Description

Ajoute un nouvel objet ActiveXControl au listage "HMIObjects". L'objet est inséré dans le coin supérieur gauche de la vue indiquée.

Syntaxe

Expression.`AddActiveXControl("ObjectName", "ProgID")`

Expression

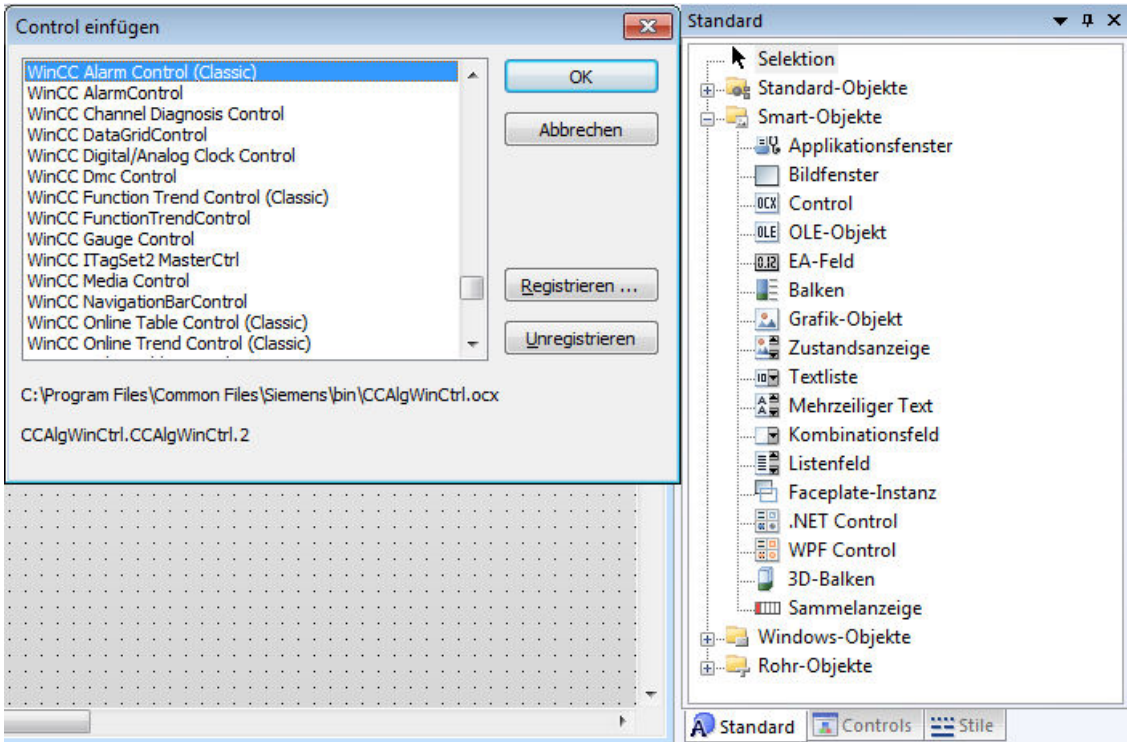
Impératif. Une expression qui renvoie un objet du type "HMIObjects".

Paramètres

Paramètre (type de données)	Description
ObjectName (String)	Le nom de l'objet. L'objet est accessible par son nom dans un listage.
ProgID (String)	Le contrôle ActiveX qui doit être inséré.

Détermination de ProgID

Vous définissez la ProgID pour un contrôle ActiveX en insérant l'objet de contrôle dans votre vue dans la "Palette d'objets" du Graphics Designer dans l'onglet Standard sous "Objets complexes". Dans le dialogue "Ajouter un contrôle", le chemin et la ProgID du contrôle sélectionné sont affichés :



Vous voyez dans le tableau suivant une liste des ProgID des contrôles WinCC qui sont installés par WinCC :

Nom du contrôle WinCC	ProgID
Siemens HMI Symbol Library	SiemensHMI.SymbolLibrary.1
WinCC AlarmControl	CCAxAlarmControl.AxAlarmControl.1
WinCC Digital/Analog Clock Control	DACLOCK.DaclockCtrl.1
WinCC FunctionTrendControl	CCAxFunctionTrendControl.AxFunctionTrendControl.1
WinCC Gauge Control	XGAUGE.XGaugeCtrl.1
WinCC Media Control	CCMediaControl.CCMediaControl.1
WinCC OnlineTableControl	CCAxOnlineTableControl.AxOnlineTableControl.1
WinCC OnlineTrendControl	CCAxOnlineTrendControl.AxOnlineTrendControl.1
WinCC Push Button Control	PBUTTON.PbuttonCtrl.1
WinCC Slider Control	SLIDER.SliderCtrl.1
WinCC RulerControl	CCAxTrendRulerControl.AxRulerControl.1
WinCC UserArchiveControl	CCAxUserArchiveControl.AxUserArchiveControl.1

Exemple

Dans l'exemple suivant, le contrôle ActiveX "WinCC Gauge Control" est inséré dans la vue active :

```
Sub AddActiveXControl()
'VBA119
Dim objActiveXControl As HMIActiveXControl
Set objActiveXControl = ActiveDocument.HMIObjects.AddActiveXControl("WinCC_Gauge",
"XGAUGE.XGaugeCtrl.1")
With ActiveDocument
.HMIObjects("WinCC_Gauge").Top = 40
.HMIObjects("WinCC_Gauge").Left = 40
End With
End Sub
```

Remarque

Après l'exécution de la méthode le Graphics Designer n'est pas terminé complètement. Le fichier "Grafexe.exe" reste dans la mémoire. Pour redémarrer le Graphics Designer vous devez quitter l'application "Grafexe.exe" du gestionnaire des tâches.

Voir aussi

ActiveX Controls (Page 1663)

Objets HMIObjects (liste) (Page 1945)

Objet ActiveXControl (Page 1873)

Référence VBA (Page 1724)

Méthode AddDotNetControl

Description

Ajoute un nouvel objet "Control .Net" à la liste "HMIObjects".

Syntaxe

```
Expression.AddDotNetControl(ObjectName, ControlType, InGAC,
AssemblyInfo)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIObjects".

Paramètre

Paramètre (type de données)	Description
ObjectName (String)	Le nom de l'objet. L'objet est accessible par son nom dans un listage.
ControllType (String)	L'espace de noms de l'objet.
InGAC (String)	TRUE: L'objet est enregistré dans le Global Assembly Cache. FALSE: L'objet n'est pas enregistré dans le Global Assembly Cache.
AssemblyInfo (String)	Si "InGAC=TRUE", l'information suivante est entrée : Assembly Version Culture PublicKeyToken Si "InGAC=FALSE", seul le chemin de l'objet est entré dans "Assembly".

Exemple

Dans l'exemple suivant, l'objet ".NET-Control" est inséré dans la vue active à partir du Global Assembly Cache :

```
'VBA851
Dim DotNetControl As HMIDotNetControl
Set DotNetControl = ActiveDocument.HMIObjects.AddDotNetControl("MyVBAControl",
"System.Windows.Forms.Label", True,"Assembly=System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089")
```

Méthode AddFolder

Description

Crée un nouveau dossier dans la bibliothèque des modules. L'objet FolderItem du type "Folder" est ajouté au listage FolderItems.

Le nouveau dossier ainsi créé se voit attribué le nom interne "FolderX", "X" étant son numéro d'ordre à partir de 1. On utilise ce nom interne pour accéder au nouveau dossier dans le listage FolderItems.

Syntaxe

Expression.AddFolder(DefaultName)

Expression

Impératif. Une expression qui renvoie un objet du type "FolderItems".

Paramètres

Paramètre (type de données)	Description
DefaultName (String)	Le nom du dossier à créer.

Exemple

Dans l'exemple qui suit, le dossier Mon dossier est créé dans la "Bibliothèque du projet" :

```
Sub AddNewFolderToProjectLibrary()
'VBA120
Dim objProjectLib As HMISymbolLibrary
Set objProjectLib = Application.SymbolLibraries(2)
objProjectLib.FolderItems.AddFolder ("My Folder")
End Sub
```

Voir aussi

Objet SymbolLibrary (Page 2023)

Objet FolderItems (liste) (Page 1929)

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Méthode AddFromClipboard

Description

Copie un objet depuis le presse-papier dans un dossier de la bibliothèque du projet. L'objet FolderItem du type "Item" est ajouté au listage FolderItems.

Remarque

Le presse-papier doit contenir des objets provenant de Graphics Designer. Les autres contenus (p.ex. du texte ASCII) ne sont pas insérés.

Syntaxe

Expression.`AddFromClipboard(DefaultName)`

Expression

Impératif. Une expression qui renvoie un objet du type "FolderItems".

Paramètres

Paramètre (type de données)	Description
DefaultName (String)	Le nom que doit porter l'objet inséré dans la bibliothèque des modules.

Exemple

Dans l'exemple qui suit, l'objet "PC" se trouvant dans la "Bibliothèque globale" est copié dans le dossier "Mon dossier 3" de la "bibliothèque du projet" :

```
Sub CopyObjectFromGlobalLibraryToProjectLibrary()
'VBA121
Dim objGlobalLib As HMISymbolLibrary
Dim objProjectLib As HMISymbolLibrary
Set objGlobalLib = Application.SymbolLibraries(1)
Set objProjectLib = Application.SymbolLibraries(2)
objProjectLib.FolderItems.AddFolder ("My Folder3")
'
'copy object from "Global Library" to clipboard
With objGlobalLib
.FolderItems(2).Folder.Item(2).Folder.Item(1).CopyToClipboard
End With
'
'paste object from clipboard into "Project Library"
objProjectLib.FolderItems(objProjectLib.FindByDisplayName("My
Folder3").Name).Folder.AddFromClipBoard ("Copy of PC/PLC")
End Sub
```

Voir aussi

Objet FolderItems (liste) (Page 1929)
Objet SymbolLibrary (Page 2023)
Référence VBA (Page 1724)
Accès à la bibliothèque des modules avec VBA (Page 1640)

Méthode AddHMIObject

Description

Ajoute un nouvel objet simple, complexe ou Windows au listage "HMIObjects". Cet objet est inséré dans le coin supérieur gauche de la vue indiquée.

Remarque

Pour insérer une commande ActiveX, on utilise la méthode AddActiveXControl.

Pour insérer un objet OLE, on utilise la méthode AddOLEObject.

Syntaxe

Expression.`AddHMIObject("ObjectName", "ProgID")`

Expression

Impératif. Une expression qui renvoie un objet du type "HMIObjects".

Paramètres

Paramètre (type de données)	Description
ObjectName (String)	Le nom de l'objet. L'objet est accessible par son nom dans un listage.
ProgID (String)	Le type d'objet à insérer. Pour obtenir la ProgID, on écrit le préfixe "HMI" devant le nom d'objet VBA (p. ex. "HMICircle" ou "HMIRectangle").

Exemple

Dans l'exemple qui suit, un cercle est inséré avec la couleur de fond "Rouge" dans la vue active :

```
Sub AddCircleToActiveDocument()
'VBA122
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("VBA_Circle", "HMICircle")
```

```
objCircle.BackColor = RGB(255, 0, 0)
End Sub
```

Voir aussi

Objet PieSegment (Page 1983)
Objet TextList (Page 2025)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet PictureWindow (Page 1980)
Objet OptionGroup (Page 1977)
Objets HMIObjects (liste) (Page 1945)
Objet Line (Page 1958)
Objet IOField (Page 1947)
Objet GraphicObject (Page 1931)
Objet EllipseArc (Page 1917)
Objet EllipseSegment (Page 1920)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)
Objet BarGraph (Page 1881)
Objet ApplicationWindow (Page 1879)
Méthode AddOLEObject (Page 1783)
Méthode AddActiveXControl (Page 1774)
Référence VBA (Page 1724)

Méthode AddItem

Description

Copie un objet de la vue indiquée dans un dossier de la bibliothèque des modules. L'objet FolderItem du type "Item" est ajouté au listage FolderItems.

Syntaxe

Expression.`Folder.AddItem "DefaultName", pHMIObject`

Expression

Impératif. Une expression qui renvoie un objet du type "FolderItems".

Paramètres

Paramètre (type de données)	Description
DefaultName (String)	Le nom que doit porter l'objet inséré dans la bibliothèque des modules.
pHMIObject (HMIObject)	L'objet à insérer dans la bibliothèque des modules à partir de la vue indiquée.

Exemple

Dans l'exemple qui suit, un cercle doit être copié dans la "bibliothèque de projet". A cet effet, on insère le cercle dans la vue active et on crée le dossier "Mon dossier2" dans la "bibliothèque du projet" :

```
Sub VBA123()
'VBA123
Dim objProjectLib As HMISymbolLibrary
Dim objCircle As HMICircle

Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle", "HMICircle")
Set objProjectLib = Application.SymbolLibraries(2)
objProjectLib.FolderItems.AddFolder ("My Folder2")
objProjectLib.FindByDisplayName("My Folder2").Folder.AddItem "ProjectLib Circle",
ActiveDocument.HMIObjects("Circle")
End Sub
```

Voir aussi

Objet FolderItems (liste) (Page 1929)

Objet SymbolLibrary (Page 2023)

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Méthode AddOLEObject

Description

Ajoute un nouvel objet OLE au listage "HMIObjects". Cet objet est inséré dans le coin supérieur gauche de la vue indiquée.

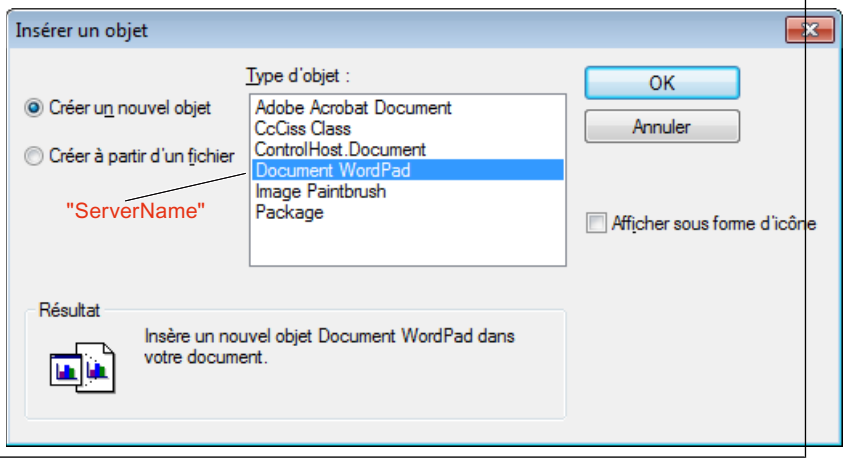
Syntaxe

```
Expression.AddOLEObject(ObjectName, ServerName, [CreationType],
[UseSymbol])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIObjects".

Paramètres

Paramètre (type de données)	Description
ObjectName (String)	Le nom de l'objet. L'objet est accessible par son nom dans un listage.
ServerName (String)	Le nom de l'application devant contenir l'objet OLE ou le nom du fichier avec indication du chemin. La valeur de "Nom du serveur" correspond au "Type d'objet" dans le dialogue "Insérer objet" : 

Paramètre (type de données)	Description
CreationType (HMIOLEObjectCreation Type-)	Détermine s'il faut créer l'objet OLE ou utiliser un fichier existant : • HMIOLEObjectCreationTypeDirect : correspond au paramètre "Nouveau". Cette option est utilisée si vous ne spécifiez pas le paramètre. • HMIOLEObjectCreationTypeByLink : correspond à l'option "Créer du fichier". qui crée une copie du fichier. Les modifications apportées à l'objet OLE ne sont pas répercutées au fichier original. On nomme le fichier à l'aide du paramètre "Nom du serveur". • HMIOLEObjectCreationTypeByLinkWithReference : Comme précédemment à la seule différence que les modifications dans l'objet OLE ont des répercussions sur le fichier d'origine. On nomme le fichier à l'aide du paramètre "Nom du serveur".
UseSymbol (Boolean)	VRAI si l'icône standard du type de fichier est utilisée. Un double-clic sur l'icône ouvre alors l'application correspondante. La valeur par défaut de ce paramètre est FAUX.

Exemple

Dans l'exemple qui suit, un objet OLE est inséré dans la vue active qui referme un document Wordpad :

```
Sub AddOLEObjectToActiveDocument()
'VBA124
Dim objOLEObject As HMIOLEObject
Set objOLEObject = ActiveDocument.HMIObjects.AddOLEObject("MS Wordpad Document",
"Wordpad.Document.1")
End Sub
```

Dans l'exemple qui suit, on utilise la méthode AddOLEObject avec le paramètre "HMIOLEObjectCreationTypeByLink" :

```
Sub AddOLEObjectByLink()
'VBA805
Dim objOLEObject As HMIOLEObject
Dim strFilename As String
'
'Add OLEObject by filename. In this case, the filename has to
'contain filename and path.
'Replace the definition of strFilename with a filename with path
'existing on your system
strFilename = Application.ApplicationDataPath & "Test.bmp"
Set objOLEObject = ActiveDocument.HMIObjects.AddOLEObject("OLEObject1", strFilename,
hmiOLEObjectCreationTypeByLink, False)
End Sub
```

Dans l'exemple qui suit, on utilise la méthode AddOLEObject avec le paramètre "HMIOLEObjectCreationTypeByLinkWithReference" :

```
Sub AddOLEObjectByLinkWithReference()
'VBA806
Dim objOLEObject As HMIOLEObject
Dim strFilename As String
'
'Add OLEObject by filename. In this case, the filename has to
'contain filename and path.
'Replace the definition of strFilename with a filename with path
'existing on your system
strFilename = Application.ApplicationDataPath & "Test.bmp"
Set objOLEObject = ActiveDocument.HMIObjects.AddOLEObject("OLEObject1", strFilename,
hmiOLEObjectCreationTypeByLinkWithReference, True)
End Sub
```

Voir aussi

Objet OLEObject (Page 1975)

Objets HMIObjects (liste) (Page 1945)

Référence VBA (Page 1724)

Méthode AddWPFControl

Description

Ajoute un nouvel objet "Control WPF" à la liste "HMIObjects".

Syntaxe

```
Expression.AddWPFControl(ObjectName, ControlType, InGAC,
AssemblyInfo)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIObjects".

Paramètre

Paramètre (type de données)	Description
ObjectName (String)	Le nom de l'objet. L'objet est accessible par son nom dans un listage.
ControllType (String)	L'espace de noms de l'objet.

Paramètre (type de données)	Description
InGAC (String)	TRUE: L'objet est enregistré dans le Global Assembly Cache. FALSE: L'objet n'est pas enregistré dans le Global Assembly Cache.
AssemblyInfo (String)	Si "InGAC=TRUE", l'information suivante est entrée : Assembly Version Culture PublicKeyToken Si "InGAC=FALSE", seul le chemin de l'objet est entré dans "Assembly".

Exemple

Dans l'exemple suivant, l'objet "WPF-Control" est inséré dans la vue active en dehors du Global Assembly Cache :

```
'VBA852
Dim WPFControl As HMIWPFControl
Set WPFControl = ActiveDocument.HMIObjects.AddWPFControl("MyWPFVBAControl",
"WinCCWPFControl.TestControl", False, "Assembly=Z:\TestControl\WinCCWPFControl.dll")
```

Méthode AlignBottom

Description

Aligne vers le bas les objets sélectionnés dans la vue indiquée. Cet alignement se fait par rapport à l'objet qui a été sélectionné en premier.

Syntaxe

Expression.`AlignBottom()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans deux positions différentes dans la vue actuelle, puis alignés vers le bas :

```
Sub AlignSelectedObjectsBottom()
'VBA125
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.AlignBottom
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode AlignLeft

Description

Aligne vers la gauche les objets sélectionnés dans la vue indiquée. Cet alignement se fait par rapport à l'objet qui a été sélectionné en premier.

Syntaxe

Expression.`AlignLeft()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans deux positions différentes dans la vue actuelle, puis alignés vers la gauche :

```
Sub AlignSelectedObjectsLeft()
'VBA126
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.AlignLeft
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode AlignRight

Description

Aligne vers la droite les objets sélectionnés dans la vue indiquée. Cet alignement se fait par rapport à l'objet qui a été sélectionné en premier.

Syntaxe

Expression.`AlignRight()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple suivant, deux objets sont insérés à deux positions dans la vue active qui sont ensuite alignées à droite :

```
Sub AlignSelectedObjectsRight()
'VBA127
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.AlignRight
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode AlignTop

Description

Aligne vers le haut les objets sélectionnés dans la vue indiquée. Cet alignement se fait par rapport à l'objet qui a été sélectionné en premier.

Syntaxe

Expression.`AlignTop()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans deux positions différentes dans la vue actuelle, puis alignés vers le haut :

```
Sub AlignSelectedObjectsTop()
'VBA128
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.AlignTop
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode ArrangeMinimizedWindows

Description

Dispose toutes les vues minimisées le long du bord inférieur de Graphics Designer.

Syntaxe

Expression.`ArrangeMinimizedWindows()`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

--

Exemple

Dans l'exemple qui suit, toutes les vues minimisées sont disposées le long du bord inférieur de Graphics Designer. Pour que cet exemple fonctionne, vous devez avoir minimisé plusieurs vues dans Graphics Designer :

```
Sub ArrangeMinimizedWindows()
'VBA129
Application.ArrangeMinimizedWindows
End Sub
```

Voir aussi

Objet Application (Page 1876)

Référence VBA (Page 1724)

Méthode BackwardOneLevel

Description

Déplace les objets sélectionnés d'un niveau d'arrière-plan de leur niveau actuel.

Syntaxe

Expression.`BackwardOneLevel()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère deux objets dans la vue active. L'objet inséré en dernier est ensuite déplacé d'un niveau à l'arrière-plan :

```
Sub MoveObjectOneLevelBackward()
'VBA173
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
```

```
.Left = 40
.Selected = False
End With
With objRectangle
.Top = 40
.Left = 40
.Width = 100
.Height = 50
.BackColor = RGB(255, 0, 255)
.Selected = True
End With
MsgBox "Objects created and selected!"
ActiveDocument.Selection.BackwardOneLevel
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode BringToFront

Description

Place les objets sélectionnés au premier plan de leur niveau actuel.

Remarque

Lorsque la méthode "BringToFront" est utilisée, l'ordre des objets HMI peut changer dans la liste HMIObjects.

Syntaxe

Impression.`BringToFront()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère deux objets dans la vue active. L'objet inséré en dernier est ensuite placé au premier plan :

```
Sub MoveObjectToFront()
'VBA198
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 40
.Left = 40
.Width = 100
.Height = 50
.BackColor = RGB(255, 0, 255)
.Selected = False
End With
MsgBox "The objects circle and rectangle are created" & vbCrLf & "Only the circle is
selected!"
ActiveDocument.Selection.BringToFront
MsgBox "The selection is moved to the front."
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode CascadeWindows

Description

Dispose en superposition partielle toutes les vues ouvertes dans Graphics Designer.

Syntaxe

Expression.`Methode(Parameter)`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

--

Exemple

Dans l'exemple qui suit, toutes les vues ouvertes dans Graphics Designer sont disposées en superposition partielle. Pour que cet exemple s'applique, vous devez avoir ouvert plusieurs vues dans Graphics Designer :

```
Sub CascadeWindows()
'VBA130
Application.CascadeWindows
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Objet Application (Page 1876)

Méthode CenterHorizontally

Description

Centre horizontalement les objets sélectionnés dans la vue indiquée.

Syntaxe

Expression.`CenterHorizontally()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans deux positions différentes dans la vue actuelle, puis centrés horizontalement :

```
Sub CenterSelectedObjectsHorizontally()
'VBA131
Dim objCircle As HMICircle
```

```
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.CenterHorizontally
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode CenterVertically

Description

Centre verticalement les objets sélectionnés dans la vue indiquée.

Syntaxe

Expression.`CenterVertically()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans deux positions différentes dans la vue actuelle, puis centrés verticalement :

```
Sub CenterSelectedObjectsVertically()
'VBA132
```

```
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.CenterVertically
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode CheckSyntax

Description

Vérifie la conformité de la syntaxe du script en C indiqué.

La méthode CheckSyntax s'utilise conjointement avec la propriété Compiled.

Syntaxe

Expression.`CheckSyntax(CheckOK, Error)`

Expression

Impératif. Une expression qui renvoie un objet du type "DynamicDialog".

Paramètres

Paramètre (type de données)	Description
CheckOK (Boolean)	TRUE si la syntaxe du script en C indiqué est conforme.
Error (String)	Le texte du message délivré en cas de non-conformité de la syntaxe du script en C.

Exemple

--

Voir aussi

Objet DynamicDialog (Page 1912)
Référence VBA (Page 1724)

Méthode Close

Description

Ferme la vue indiquée et la supprime du listage Documents.

Remarque

Les modifications qui n'auront pas été enregistrées seront perdues.

Syntaxe 1

Expression.`Close(FileName)`

Expression

Impératif. Une expression qui renvoie un objet du type "Documents".

Syntaxe 2

Expression.`Close()`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
FileName (String)	Le nom du fichier PDL à fermer.

Exemple

Dans l'exemple qui suit, on ferme la vue "Test.PDL". Pour que cet exemple fonctionne, vous devez avoir ouvert la vue "Test.PDL" :

```
Sub CloseDocumentUsingTheFileName()
'VBA134
Dim strFile As String
strFile = Application.ApplicationDataPath & "test.pdl"
Application.Documents.Close (strFile)
End Sub
Dans l'exemple qui suit, on ferme la vue actuelle dans Graphics Designer :
Sub CloseDocumentUsingActiveDocument()
'VBA135
ActiveDocument.Close
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété ActiveDocument (Page 2055)

Objet Documents (liste) (Page 1911)

Référence VBA (Page 1724)

Méthode CloseAll

Description

Ferme toutes les vues ouvertes dans Graphics Designer et les supprime du listage Documents.

Remarque

Les modifications qui n'auront pas été enregistrées seront perdues.

Syntaxe

Expression.`CloseAll()`

Expression

Impératif. Une expression qui renvoie un objet du type "Documents".

Paramètres

--

Exemple

Dans l'exemple qui suit, on ferme toutes les vues qui sont ouvertes dans Graphics Designer.

```
Sub CloseAllDocuments()
'VBA136
Application.Documents.CloseAll
End Sub
```

Voir aussi

Objet Documents (liste) (Page 1911)

Référence VBA (Page 1724)

Méthode ConvertToScript

Description

Convertit en script en C le dialogue Dynamique indiqué.

La conversion supprime l'objet DynamicDialog associé.

Remarque

La conversion est définitive et ne peut pas être annulée.

Syntaxe

Expression.`ConvertToScript()`

Expression

Impératif. Une expression qui renvoie un objet du type "DynamicDialog".

Paramètres

--

Exemple

Dans l'exemple qui suit, un cercle est inséré dans la vue active et son rayon est dynamisé dans le dialogue Dynamique. Le dialogue dynamique est ensuite converti en un script en C :

```
Sub ConvertDynamicDialogToScript()
'VBA137
Dim objDynDialog As HMIDynamicDialog
```

```
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
'
'Create dynamic
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
'
'configure dynamic. "ResultType" defines the valuerange-type:
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.Add 50, 40
.AnalogResultInfos.Add 100, 80
.AnalogResultInfos.ElseCase = 100
MsgBox "The dynamic-dialog will be changed into a C-script."
.ConvertToScript
End With
End Sub
```

Voir aussi

Objet DynamicDialog (Page 1912)

Référence VBA (Page 1724)

Méthode CopySelection

Description

Copie dans le presse-papier tous les objets de la vue sélectionnés.

Syntaxe

Expression.`CopySelection()`

Expression

Impératif. Une expression qui renvoie un objet du type "Document" ou "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, les deux objets sont insérées dans la vue active et sélectionnés. La sélection est copiée et insérée dans une nouvelle vue :

```
Sub CopySelectionToNewDocument()
```

```
'VBA138
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim iNewDoc As Integer
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
'Instead of "ActiveDocument.CopySelection" you can also write:
'"ActiveDocument.Selection.CopySelection".
ActiveDocument.CopySelection
Application.Documents.Add hmiOpenDocumentTypeVisible
iNewDoc = Application.Documents.Count
Application.Documents(iNewDoc).PasteClipboard
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété ActiveDocument (Page 2055)

Objet Selection (liste) (Page 2010)

Méthode PasteClipboard (Page 1843)

Méthode Add (listage Documents) (Page 1769)

Méthode Activate (Page 1765)

Référence VBA (Page 1724)

Méthode CopyToClipboard

Description

Copie dans le presse-papier un objet d'un dossier de la bibliothèque des modules.

Syntaxe

Expression.`CopyToClipboard()`

Expression

Impératif. Une expression qui renvoie un objet FolderItem; du type "Item".

Paramètres

--

Exemple

Dans l'exemple qui suit, l'objet "PC" se trouvant dans la Bibliothèque globale est copié dans le dossier "Mon dossier3" de la "bibliothèque du projet" :

```
Sub CopyObjectFromGlobalLibraryToProjectLibrary()
'VBA139
Dim objGlobalLib As HMISymbolLibrary
Dim objProjectLib As HMISymbolLibrary
Set objGlobalLib = Application.SymbolLibraries(1)
Set objProjectLib = Application.SymbolLibraries(2)
objProjectLib.FolderItems.AddFolder ("My Folder3")
'
'copy object from "Global Library" to clipboard
With objGlobalLib
.FolderItems(2).Folder.Item(2).Folder.Item(1).CopyToClipboard
End With
'
'paste object from clipboard into "Project Library"
objProjectLib.FolderItems(objProjectLib.FindByDisplayName("My
Folder3")).Folder.AddFromClipBoard ("Copy of PC/PLC")
End Sub
```

Voir aussi

Objet SymbolLibrary (Page 2023)

Objet FolderItem (Page 1927)

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Méthode CreateCustomizedObject

Description

Crée un objet utilisateur à partir des objets sélectionnés dans la vue indiquée. Vous devez ensuite configurer l'objet utilisateur dans le dialogue "Configuration".

Vous trouverez des renseignements complémentaires sur ce thème à la rubrique "Objets utilisateur" de cette documentation et au même intitulé de rubrique dans la documentation WinCC.

Syntaxe

Expression.`CreateCustomizedObject()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans deux positions différentes dans la vue actuelle, puis un objet utilisateur est créé :

```
Sub CreateCustomizedObject()
'VBA140
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objCustObject As HMICustomizedObject
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
Set objCustObject = ActiveDocument.Selection.CreateCustomizedObject
objCustObject.ObjectName = "myCustomizedObject"
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Objet CustomizedObject (Page 1900)

Référence VBA (Page 1724)

Objets utilisateur (Page 1676)

Méthode CreateDynamic

Description

Dynamise une propriété donnée.

Syntaxe

Expression.`CreateDynamic(DynamicType, [SourceCode])`

Expression

Impératif. Une expression qui renvoie un objet du type "Property".

Paramètres

Vous ne devez utiliser le paramètre "SourceCode" que si vous souhaitez dynamiser la propriété en question à l'aide du dialogue Dynamique.

Dans tous les autres types de dynamisation, vous pouvez négliger ce paramètre.

Paramètre (type de données)	Description
DynamicType (HMIDynamicCreationType)	Détermine le type de dynamisation : • hmiDynamicCreationTypeVariableDirect: Dynamisation avec une variable. • hmiDynamicCreationTypeVariableIndirect : dynamisation avec une variable. Dans ce type de dynamisation, on n'indique que le nom de la variable dont la valeur sera utilisée pour la dynamisation. • hmiDynamicCreationTypeScript : dynamisation par un script (C, VB). • hmiDynamicCreationTypeDynamicDialog : dynamisation avec la boîte de dialogue dynamique.
SourceCode (String)	Détermine la fonction ou la variable qui sera utilisée pour la dynamisation. Ajoutez des guillemets au nom de la variable : "Nom de variable"

Exemple

Dans cet exemple, on dynamise la propriété "Top" d'un cercle par la variable "NewDynamic" :

```
Sub AddDynamicAsVariableDirectToProperty()
'VBA141
Dim objVariableTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("MyCircle", "HMICircle")
'Make property "Top" dynamic:
Set objVariableTrigger = objCircle.Top.CreateDynamic(hmiDynamicCreationTypeVariableDirect,
"NewDynamic")
'
'Define cycle-time
With objVariableTrigger
.CycleType = hmiCycleType_2s
End With
```

```
End Sub
```

Voir aussi

Objet Property (Page 1993)

Méthode DeleteDynamic (Page 1809)

Référence VBA (Page 1724)

Méthode CreateGroup

Description

Crée un objet-groupe à partir des objets sélectionnés dans la vue indiquée.

Vous trouverez des renseignements complémentaires sur ce thème à la rubrique "Objets groupes" de cette documentation et au même intitulé de rubrique dans la documentation WinCC.

Syntaxe

Expression.`CreateGroup()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans deux positions différentes dans la vue actuelle, puis un objet-groupe est créé :

```
Sub CreateGroup()
'VBA142
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
```

```
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
objGroup.ObjectName = "myGroup"
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Objet Group (Page 1934)

Référence VBA (Page 1724)

Objets Groupe (Page 1668)

D-M

Méthode GetDeclutterObjectSize

Description

Lit les valeurs limites pour l'affichage/le masquage d'objets (Decluttering) dans la vue indiquée.

Syntaxe

Expression.`GetDeclutterObjectSize(Min, Max)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
Min (Long)	Limite inférieure de la plage en pixels.
Max (Long)	Limite supérieure de la plage en pixels.

Exemple

Dans l'exemple ci-dessous, les valeurs limites de Decluttering de la vue active sont lues et émises :

```
Sub ReadSettingsOfPicture()
'VBA848
Dim objectsize_min As Long, objectsize_max As Long

ActiveDocument.GetDeclutterObjectSize objectsize_min, objectsize_max
MsgBox objectsize_min & " " & objectsize_max

End Sub
```

Méthode Delete

Description

Supprime l'objet indiqué et le supprime du listage.

Syntaxe

Expression.`Delete()`

Expression

Impératif. Une expression qui renvoie un des types d'objet suivants :

- Assignment
- FolderItem
- LanguageText
- Menu
- MenuItem
- Object
- Toolbar
- ToolbarItem
- VariableTrigger
- View

Paramètres

--

Exemple

Dans l'exemple qui suit, on supprime le premier objet de la vue active. Pour que cet exemple fonctionne, vous devez avoir créé au moins un objet dans la vue active :

```
Sub ObjectDelete()
'VBA143
ActiveDocument.HMIObjects(1).Delete
End Sub
```

Voir aussi

Objet LanguageText (Page 1953)

Objet View (Page 2050)

Objet VariableTrigger (Page 2048)

Objet ToolbarItem (Page 2031)

Objet FolderItem (Page 1927)

Objet HMIObject (Page 1943)

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Référence VBA (Page 1724)

Méthode DeleteAll

Description

Supprime tous les objets sélectionnés dans la vue indiquée et les supprime des listages "Selection" et "HMIObjects".

Syntaxe

Expression.`DeleteAll()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés en deux positions différentes dans la vue actuelle, puis sélectionnés et supprimés :

```
Sub DeleteAllSelectedObjects()
'VBA145
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.DeleteAll
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode DeleteDynamic

Description

Supprime la dynamique de la propriété indiquée.

Syntaxe

Expression.`DeleteDynamic`

Expression

Impératif. Une expression qui renvoie un objet du type "Property".

Paramètres

--

Exemple

Dans l'exemple qui suit, une supprime une dynamique qui avait été créée par la méthode CreateDynamic :

```
Sub DeleteDynamicFromObjectMeinKreis()
'VBA146
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects("MyCircle")
objCircle.Top.DeleteDynamic
End Sub
```

Voir aussi

Objet Property (Page 1993)

Méthode CreateDynamic (Page 1803)

Référence VBA (Page 1724)

Méthode DeselectAll

Description

Désélectionne tous les objets sélectionné dans la vue indiquée et les supprime du listage Selection.

Syntaxe

Expression.`DeselectAll()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés à deux positions différentes dans la vue actuelle et sélectionnés. Ils sont ensuite désélectionnés :

```
Sub SelectObjectsAndDeselectThemAgain()
'VBA147
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
```

```
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects created and selected!"
ActiveDocument.Selection.DeselectAll
MsgBox "Objects deselected!"
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode Destroy

Description

Dissocie l'objet utilisateur indiqué. Les objets individuels sont conservés.

Syntaxe

Expression.`Destroy()`

Expression

Une expression qui renvoie un objet du type "CustomizedObject".

Paramètre

--

Exemple

Vous trouverez un exemple d'utilisation de la méthode Destroy dans cette documentation, à la rubrique "Configuration d'un objet utilisateur avec VBA".

Voir aussi

Objet CustomizedObject (Page 1900)

Méthode Destroy (Page 1811)

Méthode Delete (Page 1807)

Méthode CreateCustomizedObject (Page 1802)

Pour éditer un objet utilisateur avec VBA (Page 1677)

Méthode DuplicateSelection

Description

Duplique les objets sélectionnés dans la vue indiquée. Les objets ainsi créés sont ajoutés au listage HMIObjects. Le noms des nouveaux objets sont numérotés dans l'ordre à chaque duplication.

Si, p. ex., vous dupliquez un objet nommé "Cercle", le duplicata s'appellera "Cercle1". Si vous le dupliquez à nouveau, le duplicata s'appellera "Cercle2" et ainsi de suite.

Syntaxe

Expression.`DuplicateSelection()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés à deux positions différentes dans la vue actuelle et sélectionnés. Ils sont ensuite dupliqués :

```
Sub DuplicateSelectedObjects()
'VBA149
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
```

```
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects created and selected!"
ActiveDocument.Selection.DuplicateSelection
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Objets HMIObjects (liste) (Page 1945)

Référence VBA (Page 1724)

Méthode EvenlySpaceHorizontally

Description

Applique un espacement horizontal égal entre les objets sélectionnés dans la vue indiquée.

Syntaxe

Expression.`EvenlySpaceHorizontally()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, trois objets sont insérés dans trois positions différentes dans la vue actuelle et sélectionnés. On applique ensuite entre eux un espacement horizontal égal :

```
Sub EvenlySpaceObjectsHorizontally()
'VBA150
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
```

```
.Left = 0
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 42
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects created and selected!"
ActiveDocument.Selection.EvenlySpaceHorizontally
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Objet Selection (liste) (Page 2010)

Méthode EvenlySpaceVertically

Description

Applique un espacement vertical égal entre les objets sélectionnés dans la vue indiquée.

Syntaxe

Expression.`EvenlySpaceVertically()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, trois objets sont insérés dans trois positions différentes dans la vue actuelle et sélectionnés. On applique ensuite entre eux un espacement vertical égal :

```
Sub EvenlySpaceObjectsVertically()
```

```
'VBA151
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
.Left = 0
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 42
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects created and selected"
ActiveDocument.Selection.EvenlySpaceVertically
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode Export

Description

Enregistre la vue indiquée sous forme de fichier EMF.

Syntaxe

Expression.`Export(Type, Path)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
Type (HMIImportExportType)	Détermine le format d'enregistrement de la vue exportée.
Path (String)	Le chemin d'exportation de la vue. Ce chemin doit avoir été défini.

Exemple

```
Sub ExportAllPicturesAsPDL()
'VBA152
Dim iPictureCounter As Integer
Dim strPath As String

strPath = "C:\WinCC_PDL_Export\"

'Count Pictures in Graphics Designer...
For iPictureCounter = 1 To grafexe.Documents.Count
    '...and export each picture as PDL-file to specified path :
    grafexe.Documents(iPictureCounter).Export hmiImportExportTypePDL,
strPath
Next iPictureCounter
End Sub
```

Voir aussi

Objet View (Page 2050)

Objet Document (Page 1908)

Méthode Find

Description

Recherche des objets dans la vue indiquée et donne le résultat de la recherche sous forme d'un objet Collection. Les propriétés d'objet que vous pouvez rechercher sont les suivantes :

- Type
- Nom
- Propriété

Syntaxe

Expression.`Find([ObjectType], [ObjectName], [PropertyName])`

Expression

Impératif. Une expression qui renvoie un objet du type "HMIObjects".

Paramètres

Vous devez obligatoirement renseigner l'un des trois paramètres.

Paramètre (type de données)	Description
ObjectType (String)	Le type d'objet à rechercher. A cet effet, indiquez la "ProgID" de l'objet. Pour obtenir la ProgID, on écrit le préfixe "HMI" devant le nom d'objet VBA (p. ex. "HMICircle" ou "HMIRectangle").
ObjectName (String)	Le nom de l'objet à rechercher. Vous pouvez utiliser des caractères génériques (?, *) dans le nom de l'objet pour rechercher des objets dont les noms sont similaires.
PropertyName (String)	Le nom de la caractéristique d'objet à rechercher. A cet effet, indiquez le nom de propriété VBA (p.ex. "BackColor" au lieu de "Couleur de fond").

Exemple

Dans l'exemple qui suit, la recherche porte sur des objets du type "HMICircle" dans la vue active, avec sortie du résultat :

```
Sub FindObjectsByType()
'VBA153
Dim colSearchResults As HMICollection
Dim objMember As HMIObject
Dim iResult As Integer
Dim strName As String
Set colSearchResults = ActiveDocument.HMIObjects.Find(ObjectType:="HMICircle")
For Each objMember In colSearchResults
iResult = colSearchResults.Count
strName = objMember.ObjectName
MsgBox "Found: " & CStr(iResult) & vbCrLf & "objectname: " & strName)
Next objMember
End Sub
```

Remarque

Vous trouverez des renseignements complémentaires sur l'utilisation de la méthode Find dans cette documentation, à la rubrique "Edition d'objets simples, complexes et Windows".

Voir aussi

Propriété Type (Page 2378)

Propriété Name (Page 2289)

Objet Property (Page 1993)

Objets HMIObjects (liste) (Page 1945)

Vous éditez de la sorte les objets standard, complexes, Windows et tuyaux (Page 1657)

Référence VBA (Page 1724)

Méthode FindByDisplayName

Description

Recherche l'objet indiqué dans l'ensemble de la bibliothèque des modules. Le résultat de la recherche est fourni sous la forme d'un objet FolderItem.

Remarque

Le nom d'affichage de l'objet dépend de la langue. La recherche ne s'effectue que dans la langue configurée actuellement utilisée. La recherche se termine dès que le premier objet est trouvé.

Syntaxe

Expression.`FindByDisplayName(DisplayName)`

Expression

Impératif. Une expression qui renvoie un objet du type "SymbolLibrary" ou le listage "FolderItems".

Paramètres

Paramètre (type de données)	Description
DisplayName (String)	Le nom d'affichage de l'objet à rechercher dans la bibliothèque des modules.

Exemple

Dans l'exemple qui suit, on recherche l'objet "PC" dans la bibliothèque globale et on obtient son nom d'affichage :

```
Sub FindObjectInSymbolLibrary()
'VBA154
```

```
Dim objGlobalLib As HMISymbolLibrary
Dim objFItem As HMIFolderItem
Set objGlobalLib = Application.SymbolLibraries(1)
Set objFItem = objGlobalLib.FindByDisplayName("PC")
MsgBox objFItem.DisplayName
End Sub
```

Voir aussi

Objet FolderItem (Page 1927)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Méthode FlipHorizontally

Description

Retourne les objets sélectionnés dans la vue indiquée par rapport à l'axe horizontal médian.

La possibilité de retourner dépend du type d'objet (un objet OLE, p. ex., ne peut pas être retourné). Le retournement adapte les propriétés en conséquence. Par exemple, si vous retournez par rapport à l'axe horizontal médian un objet du type "StaticText", la valeur de la propriété "AlignmentTop" passe de "0" à "2".

Syntaxe

Expression.`FlipHorizontally()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, un objet StaticText est inséré dans la vue active et retourné par rapport à l'axe horizontal médian :

```
Sub FlipObjectHorizontally()
'VBA155
Dim objStaticText As HMIStaticText
Dim strPropertyName As String
Dim iPropertyValue As Integer
Set objStaticText = ActiveDocument.HMIObjects.AddHMIObject("Textfield", "HMIStaticText")
strPropertyName = objStaticText.Properties("Text").Name
With objStaticText
```

```
.Width = 120
.Text = "Sample Text"
.Selected = True
iPropertyValue = .AlignmentTop
MsgBox "Value of '" & strPropertyName & "' before flip: " & iPropertyValue
ActiveDocument.Selection.FlipHorizontally
iPropertyValue = objStaticText.AlignmentTop
MsgBox "Value of '" & strPropertyName & "' after flip: " & iPropertyValue
End With
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode FlipVertically

Description

Retourne les objets sélectionnés dans la vue indiquée par rapport à l'axe vertical médian.

La possibilité de retourner dépend du type d'objet (un objet OLE, p. ex., ne peut pas être retourné). Le retournement adapte les propriétés en conséquence. Par exemple, si vous retournez par rapport à l'axe vertical médian un objet du type "StaticText", la valeur de la propriété "AlignmentLeft" passe de "0" à "2".

Syntaxe

Expression.`FlipVertically()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, un objet StaticText est inséré dans la vue active et retourné par rapport à l'axe vertical médian :

```
Sub FlipObjectVertically()
'VBA156
Dim objStaticText As HMIStaticText
Dim strPropertyName As String
```

```
Dim iPropertyValue As Integer
Set objStaticText = ActiveDocument.HMIObjects.AddHMIObject("Textfield", "HMIStaticText")
strPropertyName = objStaticText.Properties("Text").Name
With objStaticText
.Width = 120
.Text = "Sample Text"
.Selected = True
.AlignmentLeft = 0
iPropertyValue = .AlignmentLeft
MsgBox "Value of '" & strPropertyName & "' before flip: " & iPropertyValue
ActiveDocument.Selection.FlipVertically
iPropertyValue = objStaticText.AlignmentLeft
MsgBox "Value of '" & strPropertyName & "' after flip: " & iPropertyValue
End With
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode ForwardOneLevel

Description

Déplace les objets sélectionnés d'un niveau d'avant-plan de leur niveau actuel.

Syntaxe

Expression.`ForwardOneLevel()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère deux objets dans la vue active. L'objet inséré en dernier est ensuite déplacé d'un niveau à l'avant-plan :

```
Sub MoveObjectOneLevelForward()
'VBA174
Dim objCircle As HMICircle
```

```
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 40
.Left = 40
.Width = 100
.Height = 50
.BackColor = RGB(255, 0, 255)
.Selected = False
End With
MsgBox "Objects created and selected!"
ActiveDocument.Selection.ForwardOneLevel
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode GetItemByPath

Description

Renvoie un objet FolderItem (dossier ou objet) dont le chemin d'accès interne dans la bibliothèque des modules est celui indiqué.

Remarque

Pour obtenir le chemin d'accès interne, sélectionnez "Copier le chemin" dans le menu contextuel de la bibliothèque des modules. Ceci a pour effet de copier dans le presse-papier le chemin interne du dossier ou de l'objet.

Syntaxe

Expression.`GetItemByPath(PathName)`

Expression

Impératif. Une expression qui renvoie un objet du type "SymbolLibrary".

Paramètres

Paramètre (type de données)	Description
PathName (String)	Le chemin d'accès à l'objet dans la bibliothèque des modules.

Exemple

Dans cet exemple, un objet est restitué depuis la bibliothèque globale et son nom est sorti :

```
Sub ShowDisplayName()
'VBA157
Dim objGlobalLib As HMISymbolLibrary
Dim objFItem As HMIFolderItem
Set objGlobalLib = Application.SymbolLibraries(1)
Set objFItem = objGlobalLib.GetItemByPath("\Folder1\Folder2\Object1")
MsgBox objFItem.DisplayName
End Sub
```

Voir aussi

Objet SymbolLibrary (Page 2023)

Objet FolderItem (Page 1927)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Méthode InsertFromMenuItem

Description

Insère une nouvelle icône d'outil dans une barre d'outils personnalisée existante qui se réfère à une option de menu existante d'un menu personnalisé.

On utilise cette méthode pour insérer dans un menu existant une barre d'outils comportant les mêmes options de commande.

Syntaxe

Expression.`InsertFromMenuItem(Position, Key, pMenuItem, DefaultToolTipText)`

Expression

Impératif. Une expression qui renvoie un objet du type "ToolbarItems".

Paramètres

Paramètre (type de données)	Description
Position (Long)	Définit la position de l'icône d'outil dans la barre d'outils personnalisée.
Key (Variant)	Identifie l'icône. Pour "Key", utilisez des désignations univoques (p.ex. "tItem1_1").
pMenuItem (HMIMenuItem)	L'objet MenuItem a référencer.
DefaultToolTipText (String)	Définit pour l'icône le texte d'info-bulle à afficher au passage de la souris sur l'icône d'outil.

Exemple

Dans cet exemple, on insère un menu personnalisé et une barre d'outils personnalisée dans la vue active. L'icône d'outil appelle l'entrée de menu "Hello World" du menu personnalisé :

```
Sub ToolbarItem_InsertFromMenuItem()
'VBA158
Dim objMenu As HMIMenu
Dim objToolbarItem As HMIToolbarItem
Dim objToolbar As HMIToolbar
Dim objMenuItem As HMIMenuItem
Set objMenu = Application.CustomMenus.InsertMenu(1, "Menu1", "TestMenu")
'
'**************************************************
'* Note:
'* The object-reference has to be unique.
'**************************************************
'
Set objMenuItem = Application.CustomMenus(1).MenuItems.InsertMenuItem(1, "MenuItem1",
"Hello World")
Application.CustomMenus(1).MenuItems(1).Macro = "HelloWorld"
Set objToolbar = Application.CustomToolbars.Add("Toolbar1")
Set objToolbarItem = Application.CustomToolbars(1).ToolbarItems.InsertFromMenuItem(1,
"ToolbarItem1", objMenuItem, "Call's Hello World of TestMenu")
End Sub

Sub HelloWorld()
MsgBox "Procedure 'HelloWorld()' is execute."
End Sub
```

Voir aussi

Objet ToolbarItems (liste) (Page 2034)

Méthode InsertSeparator (Page 1828)

Méthode Add (listage CustomToolbars) (Page 1768)

Référence VBA (Page 1724)

Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode InsertMenu

Description

Crée un nouveau menu personnalisé. On distingue ici les menus personnalisés spécifiques d'une application et ceux spécifiques d'une vue :

- Menu spécifique à une application : Est liée à Graphics Designer et reste visible lorsque toutes les vues dans Graphics Designer sont fermées. Placez le code VBA dans le document "GlobalTemplateDocument" ou "ProjectTemplateDocument" et utilisez la propriété "Application".
- Menu spécifique à une vue est lié à une vue et n'est visible que tant que cette vue est active. Placez le code VBA dans le document "ThisDocument" de la vue considérée et utilisez la propriété "ActiveDocument".

Syntaxe

Expression.`InsertMenu(Position, Key, DefaultLabel)`

Expression

Impératif. Une expression qui renvoie un objet du type "CustomMenus".

Paramètres

Paramètre (type de données)	Description
Position (Long)	Détermine la position du menu personnalisé dans la barre de menus. Toutefois, les menus spécifiques d'une vue sont toujours placés à droite des menus spécifiques d'une application.
Key (Variant)	Identifie le menu personnalisé. Pour "Key", utilisez des désignations univoques (p. ex. "DocMenu1").
DefaultLabel (String)	Le nom du menu personnalisé.

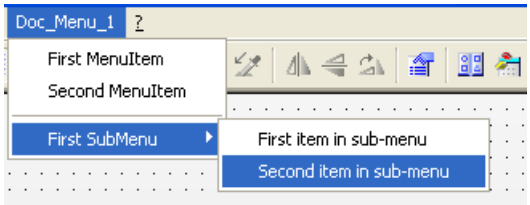
Exemple

Dans l'exemple qui suit, on crée dans la vue active un menu personnalisé à deux entrées et un sous-menu à deux entrées. Le sous-menu est visuellement caractérisé par une ligne de séparation :

```
Sub CreateDocumentMenus()
'VBA159
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objSubMenu As HMIMenuItem
'
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
'
```

```
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "First MenuItem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "Second MenuItem")
'
'Insert a dividing rule into custumized menu:
Set objMenuItem = objDocMenu.MenuItems.InsertSeparator(3, "dSeparator1_3")
'
Set objSubMenu = objDocMenu.MenuItems.InsertSubMenu(4, "dSubMenu1_4", "First SubMenu")
'
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(5, "dmItem1_5", "First item in sub-
menu")
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(6, "dmItem1_6", "Second item in sub-
menu")
End Sub
```

L'illustration représente la structure du menu ainsi obtenu.



Voir aussi

Objet Menus (liste) (Page 1965)
Méthode InsertSubmenu (Page 1829)
Méthode InsertSeparator (Page 1828)
Méthode InsertMenuItem (Page 1826)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode InsertMenuItem

Description

Insère une nouvelle entrée de menu dans un menu personnalisé.

Syntaxe

Expression.`InsertMenuItem(Position, Key, DefaultLabel)`

Expression

Impératif. Une expression qui renvoie un objet du type "MenuItems".

Paramètres

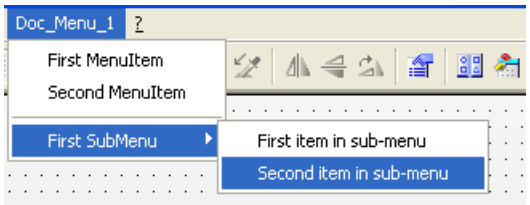
Paramètre (type de données)	Description
Position (Long)	Détermine la position du sous-menu dans le menu personnalisé.
Key (Variant)	Identifie le sous-menu Pour "Key", utilisez des désignations univoques (p.ex. "dSubMenu1_4").
DefaultLabel (String)	Nomme le sous-menu.

Exemple

Dans l'exemple qui suit, on crée dans la vue active un menu personnalisé à deux entrées et un sous-menu à deux entrées. Le sous-menu est visuellement caractérisé par une ligne de séparation :

```
Sub CreateDocumentMenus()
'VBA160
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objSubMenu As HMIMenuItem
'
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
'
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "First MenuItem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "Second MenuItem")
'
'Insert a dividing rule into customized menu:
Set objMenuItem = objDocMenu.MenuItems.InsertSeparator(3, "dSeparator1_3")
'
Set objSubMenu = objDocMenu.MenuItems.InsertSubMenu(4, "dSubMenu1_4", "First SubMenu")
'
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(5, "dmItem1_5", "First item in sub-
menu")
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(6, "dmItem1_6", "Second item in sub-
menu")
End Sub
```

L'illustration représente la structure du menu ainsi obtenu.



Voir aussi

Objet MenuItems (liste) (Page 1970)
Objet MenuItem (Page 1967)
Méthode InsertSubmenu (Page 1829)
Méthode InsertSeparator (Page 1828)
Méthode InsertMenu (Page 1825)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode InsertSeparator

Description

Insère une ligne de séparation dans un menu personnalisé ou dans une barre d'outils personnalisée.

Syntaxe

Expression.`InsertSeparator(Position, Key)`

Expression

Impératif. Une expression qui renvoie un objet du type "MenuItems" ou "ToolbarItems".

Paramètres

Paramètre (type de données)	Description
Position (Long)	Détermine la position de la ligne de séparation dans le menu ou la barre d'outils personnalisés.
Key (Variant)	Identifie la ligne de séparation. Pour "Key", utilisez des désignations univoques (p.ex. "tSeparator1_2")

Exemple

Dans l'exemple qui suit, on crée dans la vue active une barre d'outils personnalisée à deux icônes séparées par une ligne de séparation :

```
Sub AddDocumentSpecificCustomToolbar()
'VBA161
Dim objToolbar As HMIToolbar
Dim objToolbarItem As HMIToolbarItem
Set objToolbar = ActiveDocument.CustomToolbars.Add("DocToolbar")
'Add toolbar-item to userdefined toolbar
```

```
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(1, "tItem1_1", "First
symbol-icon")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(3, "tItem1_3", "Second
symbol-icon")
'
'Insert dividing rule between first and second symbol-icon
Set objToolbarItem = objToolbar.ToolbarItems.InsertSeparator(2, "tSeparator1_2")
End Sub
```

Voir aussi

Objet ToolbarItems (liste) (Page 2034)

Objet MenuItems (liste) (Page 1970)

Méthode InsertToolbarItem (Page 1831)

Référence VBA (Page 1724)

Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode InsertSubmenu

Description

Insère un sous-menu dans un menu personnalisé existant.

Syntaxe

Expression.`InsertSubmenu(Position, Key, DefaultLabel)`

Expression

Impératif. Une expression qui renvoie un objet du type "MenuItem".

Paramètres

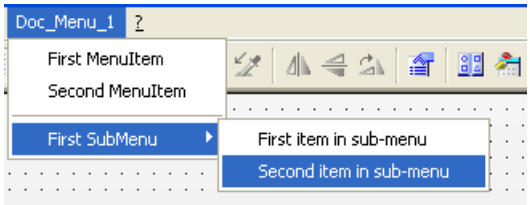
Paramètre (type de données)	Description
Position (Long)	Détermine la position du sous-menu dans le menu personnalisé.
Key (Variant)	Identifie le sous-menu Pour "Key", utilisez des désignations univoques (p.ex. "dSubMenu1_4").
DefaultLabel (String)	Nomme le sous-menu.

Exemple

Dans l'exemple qui suit, on crée dans la vue active un menu personnalisé à deux entrées et un sous-menu à deux entrées. Le sous-menu est visuellement caractérisé par une ligne de séparation :

```
Sub CreateDocumentMenus()
'VBA162
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objSubMenu As HMIMenuItem
'
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
'
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "First MenuItem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "Second MenuItem")
'
'Insert a dividing rule into customized menu:
Set objMenuItem = objDocMenu.MenuItems.InsertSeparator(3, "dSeparator1_3")
'
Set objSubMenu = objDocMenu.MenuItems.InsertSubMenu(4, "dSubMenu1_4", "First SubMenu")
'
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(5, "dmItem1_5", "First item in sub-
menu")
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(6, "dmItem1_6", "Second item in sub-
menu")
End Sub
```

L'illustration représente la structure du menu ainsi obtenu.



Voir aussi

Objet MenuItem (Page 1967)

Méthode InsertSeparator (Page 1828)

Méthode InsertMenuItem (Page 1826)

Méthode InsertMenu (Page 1825)

Référence VBA (Page 1724)

Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode InsertToolbarItem

Description

Insère une nouvelle icône d'outil dans une barre d'outils personnalisée existante.

Syntaxe

Expression.`InsertToolbarItem(Position, Key, DefaultToolTipText)`

Expression

Impératif. Une expression qui renvoie un objet du type "ToolbarItems".

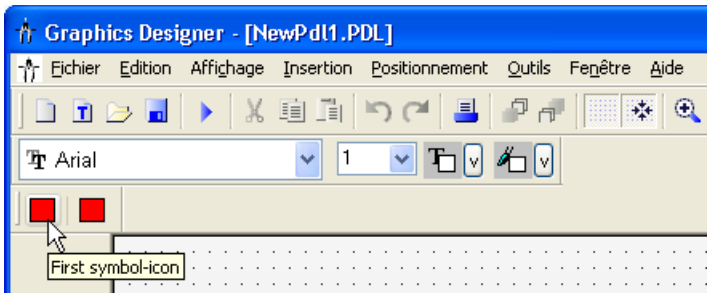
Paramètres

Paramètre (type de données)	Description
Position (Long)	Définit la position de l'icône d'outil dans la barre d'outils personnalisée.
Key (Variant)	Identifie l'icône. Pour "Key", utilisez des désignations univoques (p.ex. "tItem1_1").
DefaultToolTipText (String)	Définit pour l'icône le texte d'info-bulle à afficher au passage de la souris sur l'icône d'outil.

Exemple

Dans l'exemple qui suit, on crée dans la vue active une barre d'outils personnalisée à deux icônes séparées par une ligne de séparation :

```
Sub AddDocumentSpecificCustomToolbar()
'VBA163
Dim objToolbar As HMIToolbar
Dim objToolbarItem As HMIToolbarItem
Set objToolbar = ActiveDocument.CustomToolbars.Add("DocToolbar")
'Add toolbar-item to userdefined toolbar
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(1, "tItem1_1", "First
symbol-icon")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(3, "tItem1_3", "Second
symbol-icon")
'
'Insert dividing rule between first and second symbol-icon
Set objToolbarItem = objToolbar.ToolbarItems.InsertSeparator(2, "tSeparator1_2")
End Sub
```



Voir aussi

Objet ToolbarItems (liste) (Page 2034)

Méthode InsertSeparator (Page 1828)

Méthode Add (listage CustomToolbars) (Page 1768)

Référence VBA (Page 1724)

Création de menus et de barres d'outils personnalisés (Page 1620)

Méthode IsCSLayerVisible

Description

Retourne TRUE si le niveau CS indiqué est visible.

Syntaxe

Expression.`IsCSLayerVisible(Index)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
Indice (Variant)	Désigne le niveau CS. Plage des valeurs : 1 à 32. Le niveau 0 correspond au numéro d'indice "1".

Exemple

Dans l'exemple qui suit, on vérifie si le niveau CS n°1 de la vue active est visible et le résultat est sorti :

```
Sub IsCSLayerVisible()
```

```
'VBA164
Dim objView As HMIView
Dim strLayerName As String
Dim iLayerIdx As Integer
Set objView = ActiveDocument.Views(1)
objView.Activate
iLayerIdx = 2
strLayerName = ActiveDocument.Layers(iLayerIdx).Name
If objView.IsCSLayerVisible(iLayerIdx) = True Then
MsgBox "CS " & strLayerName & " is visible"
    Else
MsgBox "CS " & strLayerName & " is invisible"
    End If
End Sub
```

Voir aussi

Objet Document (Page 1908)

Référence VBA (Page 1724)

Edition de niveaux avec VBA (Page 1650)

Méthode IsRTLayerVisible

Description

Retourne TRUE si le niveau RT indiqué est visible.

Syntaxe

Expression.`IsRTLayerVisible(Index)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
Indice (Variant)	Désigne le niveau RT. Plage des valeurs : 1 à 32. Le niveau 0 correspond au numéro d'indice "1".

Exemple

Dans l'exemple qui suit, on vérifie si le niveau RT n°1 est visible et le résultat est renvoyé :

```
Sub RTLayerVisibility()
'VBA165
Dim strLayerName As String
Dim iLayerIdx As Integer
iLayerIdx = 2
strLayerName = ActiveDocument.Layers(iLayerIdx).Name
If ActiveDocument.IsRTLayerVisible(iLayerIdx) = True Then
MsgBox "RT " & strLayerName & " is visible"
    Else
MsgBox "RT " & strLayerName & " is invisible"
    End If
End Sub
```

Voir aussi

Objet Document (Page 1908)

Référence VBA (Page 1724)

Edition de niveaux avec VBA (Page 1650)

Méthode Item

Description

Renvoie un élément à partir d'un listage.

Syntaxe

Expression.`Item(Index)`

Expression

Impératif. Une expression qui renvoie un objet.

Paramètres

Paramètre (type de données)	Description
Indice (Variant)	Le nom ou le numéro d'indice d'un élément contenu dans un listage. Pour le nom, vous pouvez utiliser le nom de l'objet ou un numéro d'indice sous la forme d'une expression numérique (de 1 à la valeur de la propriété Count du listage). Si la valeur indiquée ne correspond à aucun élément du listage, une erreur intervient.

Exemple

Remarque

La méthode Item est la méthode par défaut pour les listages. C'est pourquoi les deux exemples qui suivent donnent le même résultat.

Dans ces deux exemples, le nom de la première vue est renvoyé dans Graphics Designer :

```
Sub ShowDocumentNameLongVersion()
'VBA166
Dim strDocName As String
strDocName = Application.Documents.Item(3).Name
MsgBox strDocName
End Sub

Sub ShowDocumentNameShortVersion()
'VBA167
Dim strDocName As String
strDocName = Application.Documents(3).Name
MsgBox strDocName
End Sub
```

Voir aussi

Objet VariableStateValues (liste) (Page 2046)
Propriété Count (Page 2141)
Objet Views (liste) (Page 2052)
Objet VariableTriggers (liste) (Page 2049)
Objet ToolbarItems (liste) (Page 2034)
Objet Toolbars (liste) (Page 2029)
Objet SymbolLibraries (liste) (Page 2024)
Objet Selection (liste) (Page 2010)
Objet Properties (liste) (Page 1992)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Objet MenuItems (liste) (Page 1970)
Objet Menus (liste) (Page 1965)
Objet Layers (Liste) (Page 1956)
Objet LanguageTexts (liste) (Page 1954)
Objet LanguageFonts (liste) (Page 1951)
Objet GroupedObjects (liste) (Page 1938)
Objet FolderItems (liste) (Page 1929)
Objet Events (liste) (Page 1924)
Objet Documents (liste) (Page 1911)
Objet DataLanguages (liste) (Page 1903)
Objet ConnectionPoints (liste) (Page 1898)
Objet AnalogResultInfos (liste) (Page 1875)
Objet Actions (liste) (Page 1872)
Référence VBA (Page 1724)

Méthode ItemByLcid

Description

Sélectionne la langue dont vous souhaitez configurer les paramètres de texte. Accès en lecture.

Remarque

Vous ne pouvez sélectionner que des langues avec lesquelles vous avez déjà travaillé.

Syntaxe

Expression.`ItemByLcid(LangID)`

Expression

Impératif. Une expression qui renvoie un objet du type "LanguageFonts".

Paramètres

Paramètre (type de données)	Description
LangID (Long)	L'identifiant de la langue. La liste des identificateurs de langue figure par exemple dans le fichier "Languages.csv" que vous trouverez dans l'indice de la documentation WinCC.

Exemple

L'exemple ci-dessous montre comment paramétrer pour un bouton les attributs de police pour le français et l'anglais. Comparé à l'anglais, une plus petite police à pas constant (Courier New, 12pt) apparaît en français sur le bouton :

```
Sub ExampleForLanguageFonts()
'VBA168
Dim objLangFonts As HMILanguageFonts
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
objButton.Text = "Hello"
Set objLangFonts = objButton.LDFonts
'
'To make fontsettings for english:
With objLangFonts.ItemByLCID(1033)
.Family = "Times New Roman"
.Bold = False
.Italic = True
.Underlined = False
.Size = 14
End With
'
'To make fontsettings for french:
With objLangFonts.ItemByLCID(1036)
.Family = "Courier New"
.Bold = True
.Italic = False
.Underlined = True
.Size = 12
End With

End Sub
```

Voir aussi

Objet LanguageFonts (liste) (Page 1951)

Méthode LoadDefaultConfig

Description

Charge le fichier des paramètres par défaut des objets. Le fichier PDD se trouve dans le dossier "GraCS" du projet en cours.

Syntaxe

Expression.`LoadDefaultConfig(FileName)`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

Paramètre (type de données)	Description
FileName (String)	Le nom du fichier PDD à charger.

Exemple

Dans l'exemple qui suit, on charge le fichier "Test.PDD". Pour que cet exemple fonctionne, il faut que le fichier ait déjà été enregistré. A cet effet, vous pouvez utiliser la méthode SaveDefaultConfig :

```
Sub LoadDefaultConfig()
'VBA169
Application.LoadDefaultConfig ("Test.PDD")
End Sub
```

Voir aussi

Objet Application (Page 1876)

Méthode SaveDefaultConfig (Page 1855)

Référence VBA (Page 1724)

Méthode MoveOneLayerDown

Description

Descend au niveau suivant l'objet sélectionné dans la vue spécifiée.

Syntaxe

Expression.`MoveOneLayerDown()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple suivant, un cercle est inséré dans la vue active au troisième niveau, puis descendu au niveau suivant.

```
Sub MoveObjectOneLayerDown()
'VBA170
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
.Layer = 3
MsgBox "Circle is inserted into layer" & Str(.Layer)
ActiveDocument.Selection.MoveOneLayerDown
MsgBox "Circle is moved into layer" & Str(.Layer)
End With
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode MoveOneLayerUp

Description

Monte au niveau suivant l'objet sélectionné dans la vue spécifiée.

Syntaxe

Expression.`MoveOneLayerUp()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple suivant, un cercle est inséré dans la vue active au troisième niveau, puis monté au niveau suivant.

```
Sub MoveObjectOneLayerUp()
'VBA171
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
.Layer = 3
MsgBox "Circle is inserted into layer" & Str(.Layer)
ActiveDocument.Selection.MoveOneLayerUp
MsgBox "Circle is moved into layer" & Str(.Layer)
End With
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode MoveSelection

Description

Déplace un ou plusieurs objets sélectionnés de la vue en fonction des coordonnées indiquées.

Remarque

Pour changer la position d'un ou de plusieurs objets sélectionnés, on utilise les propriétés "Left" et "Top".

Syntaxe

Expression.`MoveSelection(PosX, PosY)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document" ou "Selection".

Paramètres

Paramètre (type de données)	Description
PosX (Long)	Le nombre de pixels duquel la sélection doit être déplacée horizontalement.
PosY (Long)	Le nombre de pixels duquel la sélection doit être déplacée verticalement.

Exemple

Dans l'exemple qui suit, deux objets sont insérés à deux positions différentes dans la vue actuelle et sélectionnés. La sélection est ensuite déplacée de 30 pixels vers la droite et de 40 pixels vers le bas :

```
Sub MoveSelectionToNewPostion()
'VBA172
Dim nPosX As Long
Dim nPosY As Long
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
```

```
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
nPosX = 30
nPosY = 40
ActiveDocument.MoveSelection nPosX, nPosY
End Sub
```

Voir aussi

Propriété Top (Page 2374)
Propriété Left (Page 2253)
Objet Document (Page 1908)
Référence VBA (Page 1724)

O-Z

Méthode Open

Description

Ouvre une image présente dans Graphics Designer et l'ajoute au listage Documents.

Syntaxe

Expression.`Open(FileName, [HMIOpenDocumentType])`

Expression

Impératif. Une expression qui renvoie un objet du type "Documents".

Paramètres

Paramètre (type de données)	Description
FileName (String)	Le nom du fichier PDL à ouvrir. Si le fichier PDL n'a pas été enregistré dans le dossier "GraCS" du projet ouvert, vous devez en plus indiquer son chemin.
HMIOpenDocumentType (HMIDocumentType)	Définit le mode d'ouverture de la vue : • HMIDocumentTypeVisible : ouvre la vue pour traitement direct. Si vous n'indiquez pas le paramètre, ceci est le mode par défaut. • HMIDocumentTypeInvisible : ouvre la vue en invisible, c'est-à-dire qu'elle ne sera pas affichée dans Graphics Designer. Vous ne pouvez accéder à la vue que par l'intermédiaire du listage Documents et vous pouvez la régler en visible par la propriété Hide.

Exemple

Dans l'exemple qui suit, on ouvre la vue "Test". Pour que cet exemple fonctionne, une vue nommée "Test" doit déjà avoir été enregistrée dans le dossier "GraCS" du projet ouvert :

```
Sub OpenDocument()
'VBA175
Application.Documents.Open "Test.PDL", hmiOpenDocumentTypeVisible
End Sub
```

Voir aussi

Propriété Hide (Page 2202)

Objet Documents (liste) (Page 1911)

Référence VBA (Page 1724)

Méthode PasteClipboard

Description

Insère le contenu du presse-papier dans la vue indiquée.

Remarque

Le presse-papier doit contenir des objets provenant de Graphics Designer. Les autres contenus (p.ex. du texte ASCII) ne sont pas insérés.

Syntaxe

Expression.`PasteClipboard()`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

--

Exemple

Dans l'exemple qui suit, tous les objets sélectionnés de la vue active sont copiés dans le presse-papier puis insérés dans une nouvelle vue. Pour que cet exemple fonctionne, vous devez avoir sélectionné au moins un objet dans la vue active :

```
Sub CopySelectionToNewDocument()
'VBA176
Dim iNewDoc As String
ActiveDocument.CopySelection
Application.Documents.Add hmiOpenDocumentTypeVisible
iNewDoc = Application.Documents.Count
Application.Documents(iNewDoc).PasteClipboard
End Sub
```

Voir aussi

Propriété ActiveDocument (Page 2055)

Objet Document (Page 1908)

Méthode CopySelection (Page 1800)

Méthode Add (listage Documents) (Page 1769)

Méthode Activate (Page 1765)

Référence VBA (Page 1724)

Méthode PrintDocument

Description

Imprime la copie indiquée de la vue avec les paramètres d'impression actuels.

Syntaxe

Expression.`PrintDocument()`

Expression

Impératif. Une expression qui renvoie un objet du type "View".

Paramètres

--

Exemple

Dans l'exemple qui suit, on effectue une copie de la vue actuelle, puis cette copie est activée et imprimée :

```
Sub CreateAndPrintView()
'VBA177
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
objView.PrintDocument
End Sub
```

Voir aussi

Objet View (Page 2050)

Référence VBA (Page 1724)

Méthode PrintProjectDocumentation

Description

Imprime la documentation projet de la vue actuelle avec tous ses objets et ses propriétés par l'intermédiaire de l'éditeur de rapports (Report Designer) de WinCC.

Les paramètres d'impression (p. ex. Pages) doivent avoir été configurés dans le dialogue "Propriétés de travail d'impression". A cet effet, sélectionnez la commande de menu "Fichier" > "Configurer documentation de projet".

Remarque

La documentation projet est imprimée sur l'imprimante installée dans Report Designer. Vous pouvez configurer la mise en page en fonction de vos besoins dans Report Designer.

Syntaxe

Expression.`PrintProjectDocumentation()`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

--

Exemple

Dans l'exemple qui suit, on imprime la documentation projet de la vue active :

```
Sub ToPrintProjectDocumentation()
'VBA178
ActiveDocument.PrintProjectDocumentation
End Sub
```

Voir aussi

Objet Document (Page 1908)
Référence VBA (Page 1724)

Méthode Remove

Description

Supprime un objet d'un sélection d'objets ou d'un objet-groupe.

Syntaxe

Expression.`Remove(Index)`

Expression

Impératif. Une expression qui renvoie un objet du type "GroupedObjects" ou "Selection".

Paramètres

Paramètre (type de données)	Description
Indice (Variant)	Le nom ou le numéro d'indice de l'objet à supprimer. Pour le nom, vous pouvez utiliser le nom de l'objet ou un numéro d'indice sous la forme d'une expression numérique (de 1 à la valeur de la propriété Count du listage). Si la valeur indiquée ne correspond à aucun élément du listage, une erreur intervient.

Exemple

Dans l'exemple qui suit, on commence par insérer, puis sélectionner trois objets dans la vue active. Ensuite, on supprime un des objets de la sélection et on groupe les autres objets. C'est le premier objet de l'objet-groupe qui est supprimé :

```
Sub RemoveObjectFromGroup()
'VBA179
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
.Left = 0
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 42
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
.Width = 40
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
MsgBox "Group-object is created."
objGroup.GroupedHMIObjects.Remove ("sEllipse")
MsgBox "The ellipse is removed from group-object."
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)
Objet GroupedObjects (liste) (Page 1938)
Référence VBA (Page 1724)

Méthode Rotate

Description

Fait effectuer à l'objet sélectionné dans la vue indiquée une rotation de 90° dans le sens des aiguilles d'une montre.

Syntaxe

Expression.`Rotate()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, deux objets sont insérés dans la vue active, puis groupés. L'objet-groupe effectue ensuite une rotation et une seule :

```
Sub RotateGroupObject()
'VBA180
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objGroup As HMIGroup
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
With objRectangle
.Top = 30
.Left = 30
.Width = 80
.Height = 40
.Selected = True
End With
With objCircle
.Top = 30
.Left = 30
.BackColor = RGB(255, 255, 255)
.Selected = True
```

```
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
MsgBox "Group-object created."
objGroup.Selected = True
ActiveDocument.Selection.Rotate
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Objet Selection (liste) (Page 2010)

Méthode SameHeight

Description

Attribue la plus petite valeur disponible à la propriété "Height" de tous les objets sélectionnés dans la vue indiquée.

Syntaxe

Expression.`SameHeight()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère trois objets de taille différente dans la vue active. On sélectionne ensuite tous les objets et on les ramène tous à la même hauteur et à la même largeur :

```
Sub ApplySameHeightToSelectedObjects()
'VBA181
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
```

```
With objCircle
.Top = 30
.Left = 0
.Height = 15
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 42
.Height = 40
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
.Width = 40
.Height = 120
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.SameHeight
End Sub
```

Voir aussi

Propriété Height (Page 2201)

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode SameWidth

Description

Attribue la plus petite valeur disponible à la propriété "Width" de tous les objets sélectionnés dans la vue indiquée.

Syntaxe

Expression.`SameWidth()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère trois objets de taille différente dans la vue active. On sélectionne ensuite tous les objets et on les ramène tous à la même largeur :

```
Sub ApplySameWidthToSelectedObjects()
'VBA182
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
.Left = 0
.Width = 15
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 42
.Width = 40
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
.Width = 120
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.SameWidth
End Sub
```

Voir aussi

Propriété Width (Page 2472)

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode SameWidthAndHeight

Description

Attribue la plus petite valeur disponible aux propriétés "Height" et "Width" de tous les objets sélectionnés dans la vue indiquée.

Syntaxe

Expression.`SameWidthAndHeight()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère trois objets de taille différente dans la vue active. On sélectionne ensuite tous les objets et on les ramène tous à la même hauteur et à la même largeur :

```
Sub ApplySameWidthAndHeightToSelectedObjects()
'VBA183
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
.Left = 0
.Height = 15
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 42
.Width = 25
.Height = 40
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
.Width = 40
```

```
.Height = 120
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.SameWidthAndHeight
End Sub
```

Voir aussi

Propriété Width (Page 2472)

Propriété Height (Page 2201)

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode Save

Description

Enregistre la vue indiquée sous son nom actuel.

Syntaxe

Expression.`Save()`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

--

Exemple

Dans l'exemple qui suit, on enregistre la vue active dans Graphics Designer :

```
Sub SaveDocument()
'VBA184
ActiveDocument.Save
End Sub
```

Voir aussi

Propriété ActiveDocument (Page 2055)

Objet Document (Page 1908)

Référence VBA (Page 1724)

Méthode SaveAll

Description

Enregistre toutes les vues ouvertes dans Graphics Designer sous leur nom actuel.

Syntaxe

Expression.`SaveAll()`

Expression

Impératif. Une expression qui renvoie un objet du type "Documents".

Paramètres

--

Exemple

Dans l'exemple qui suit, on enregistre toutes les vues ouvertes dans Graphics Designer :

```
Sub SaveAllDocuments()
'VBA185
Application.Documents.SaveAll
End Sub
```

Voir aussi

Objet Documents (liste) (Page 1911)

Référence VBA (Page 1724)

Méthode SaveAs

Description

Enregistre la vue indiquée sous un nouveau nom.

Si une vue existant déjà doit être écrasée, assurez-vous avant d'appeler la méthode SaveAs que cette vue peut être écrasée. Vous devez à cet effet interroger la propriété LockedByCreatorID pour la vue à écraser. Faute de quoi une erreur est déclenchée dans VBA.

Syntaxe

Expression.`SaveAs(FileName)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
FileName (String)	Le nom de fichier sous lequel la vue doit être enregistrée.

Exemple

Dans l'exemple qui suit, la vue active est enregistrée sous le nom "Test2.PDL".

```
Sub SaveDocumentAs()
'VBA186
ActiveDocument.SaveAs ("Test2.PDL")
End Sub
```

Voir aussi

Propriété LockedByCreatorID (Page 2261)

Propriété ActiveDocument (Page 2055)

Objet Document (Page 1908)

Référence VBA (Page 1724)

Méthode SaveDefaultConfig

Description

Enregistre les paramètres par défaut des objets dans un fichier PDD. Le fichier PDD est enregistré dans le dossier "GraCS" du projet en cours.

Syntaxe

Expression.`SaveDefaultConfig(FileName)`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

Paramètre (type de données)	Description
FileName (String)	Le nom du fichier PDD.

Exemple

Dans l'exemple qui suit, les paramètres par défaut des objets sont enregistrés dans le fichier "Test.PDD".

```
Sub SaveDefaultConfig()
'VBA187
Application.SaveDefaultConfig ("Test.PDD")
End Sub
```

Voir aussi

Objet Application (Page 1876)

Méthode LoadDefaultConfig (Page 1838)

Référence VBA (Page 1724)

Méthode SelectAll

Description

Sélectionne tous les objets de la vue indiquée et les ajoute à la liste Selection.

Syntaxe

Expression.`SelectAll()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère trois objets dans la vue active, puis on les sélectionne.

```
Sub SelectAllObjectsInActiveDocument()
'VBA188
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
.Left = 0
.Height = 15
End With
With objRectangle
.Top = 80
.Left = 42
.Width = 25
.Height = 40
End With
With objEllipse
.Top = 48
.Left = 162
.Width = 40
.Height = 120
.BackColor = RGB(255, 0, 0)
End With
ActiveDocument.Selection.SelectAll
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode SendToBack

Description

Place les objets sélectionnés à l'arrière-plan de leur niveau actuel.

Remarque

Lorsque la méthode "SendToBack" est utilisée, l'ordre des objets HMI peut changer dans la liste HMIObjects.

Syntaxe

Impression.`SendToBack()`

Expression

Impératif. Une expression qui renvoie un objet du type "Selection".

Paramètres

--

Exemple

Dans l'exemple qui suit, on insère deux objets dans la vue active. L'objet inséré en premier est ensuite placé à l'arrière plan :

```
Sub SendObjectToBack()
'VBA197
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = False
End With
With objRectangle
.Top = 40
.Left = 40
.Width = 100
.Height = 50
.BackColor = RGB(255, 0, 255)
.Selected = True
End With
MsgBox "The objects circle and rectangle are created" & vbCrLf & "Only the rectangle is
selected!"
ActiveDocument.Selection.SendToBack
MsgBox "The selection is moved to the back."
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Référence VBA (Page 1724)

Méthode SetCSLayerVisible

Description

Affiche ou masque le niveau CS indiqué.

Syntaxe

Expression.`SetCSLayerVisible(Index, Val)`

Expression

Impératif. Une expression qui renvoie un objet du type "View".

Paramètres

Paramètre (type de données)	Description
Indice (Variant)	Détermine le niveau CS à afficher ou à masquer. Plage des valeurs : 1 à 32.
Val (Boolean)	TRUE si le niveau CS indiqué doit être visible.

Exemple

Dans l'exemple qui suit, on masque le deuxième niveau CS de la copie de la vue active :

```
Sub SetCSLayerVisible()
'VBA189
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
objView.SetCSLayerVisible 2, False
End Sub
```

Voir aussi

Objet Document (Page 1908)

Référence VBA (Page 1724)

Edition de niveaux avec VBA (Page 1650)

Méthode SetOpenContext

Description

On utilise la méthode SetOpenContext pour stocker le mot de passe d'une vue protégée par mot de passe (vue de process ou type de bloc d'affichage), puis pour ouvrir la vue.

Syntaxe

Expression.`SetOpenContext (Password)`

Expression

Impératif. Une expression qui renvoie un objet du type "Documents".

Paramètres

Paramètre (type de données)	Description
Password (String)	Mot de passe de la vue existante.

Exemple

Dans l'exemple suivant, plusieurs vues ("A.pdl", "B.pdl" et "C.pdl") s'ouvrent avec le même mot de passe "Test123". Pour ouvrir les vues, vous devez d'abord indiquer leur mot de passe. Pour que le mot de passe ne reste pas accessible, vous mettez fin à la méthode SetOpenContext avec une chaîne vide "".

```
Sub OpenProtectedPicture()
'VBAxxx
Documents.SetOpenContext ("Test123")
Documents.Open ("A.pdl")
Documents.Open ("B.pdl")
Documents.Open ("C.pdl")
Documents.SetOpenContext ("")
End Sub
```

Méthode SetDeclutterObjectSize

Description

Détermine la plage de grandeur pour l'affichage/le masquage d'objets dans la vue indiquée. Les objets sont masqués si la hauteur et la largeur de l'objet dépassent la plage de grandeur spécifiée.

La propriété "ObjectSizeDecluttering" doit être définie sur VRAI.

Syntaxe

Expression.`SetDeclutterObjectSize(Min, Max)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
Min (Long)	Limite inférieure de la plage en pixels.
Max (Long)	Limite supérieure de la plage en pixels.

Exemple

Dans l'exemple qui suit, on définit les paramètres du niveau le plus bas de la vue active :

```
Sub ConfigureSettingsOfLayer
'VBA190
Dim objLayer As HMILayer
Set objLayer = ActiveDocument.Layers(1)
With objLayer
'Configure "Layer 0"
.MinZoom = 10
.MaxZoom = 100
.Name = "Configured with VBA"
End With
'Define decluttering of objects:
With ActiveDocument
.LayerDecluttering = True
.ObjectSizeDecluttering = True
.SetDeclutterObjectSize 50, 100
End With
End Sub
```

Voir aussi

Propriété ObjectSizeDecluttering (Page 2295)

Objet Document (Page 1908)

Référence VBA (Page 1724)

Méthode SetRTLayerVisible

Description

Affiche ou masque le niveau RT indiqué.

Syntaxe

Expression.`SetRTLayerVisible(Index, Val)`

Expression

Impératif. Une expression qui renvoie un objet du type "Document".

Paramètres

Paramètre (type de données)	Description
Indice (Variant)	Détermine le niveau RT à afficher ou à masquer. Plage des valeurs de 1 à 32.
Val (Boolean)	TRUE si le niveau RT indiqué doit être visible.

Exemple

Dans l'exemple qui suit, on masque le premier niveau RT de la vue active :

```
Sub SetRTLayerVisibleWithVBA()
'VBA191
ActiveDocument.SetRTLayerVisible 1, False
End Sub
```

Voir aussi

Objet Document (Page 1908)

Référence VBA (Page 1724)

Edition de niveaux avec VBA (Page 1650)

Méthode ShowPropertiesDialog

Description

Ouvre le dialogue "Propriétés de l'objet".

Syntaxe

Expression.`ShowPropertiesDialog()`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

--

Exemple

Dans l'exemple suivant, on ouvre le dialogue "Propriétés de l'objet" :

```
Sub ShowPropertiesDialog()
'VBA192
Application.ShowPropertiesDialog
End Sub
```

Voir aussi

Objet Application (Page 1876)
Référence VBA (Page 1724)

Méthode ShowSymbolLibraryDialog

Description

Ouvre la bibliothèque des modules.

Syntaxe

Expression.`ShowSymbolLibraryDialog()`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

--

Exemple

Dans l'exemple qui suit, on ouvre la bibliothèque des modules :

```
Sub ShowSymbolLibraryDialog()
'VBA193
Application.ShowSymbolLibraryDialog
End Sub
```

Voir aussi

Objet Application (Page 1876)
Référence VBA (Page 1724)

Méthode ShowTagDialog

Description

Ouvre le dialogue "Variables".

Syntaxe

Expression.`ShowTagDialog()`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

--

Exemple

Dans l'exemple qui suit, on ouvre le dialogue "Variables" :

```
Sub ShowTagDialog()
'VBA194
Application.ShowTagDialog
End Sub
```

Voir aussi

Objet Application (Page 1876)
Référence VBA (Page 1724)

Méthode TileWindowsHorizontally

Description

Dispose horizontalement toutes les vues ouvertes dans Graphics Designer.

Syntaxe

Expression.`Methode()`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

--

Exemple

Dans l'exemple suivant, on dispose horizontalement toutes les vues ouvertes dans Graphics Designer. Pour que cet exemple s'applique, vous devez avoir ouvert plusieurs vues dans Graphics Designer :

```
Sub TileWindowsHorizontally()
'VBA195
Application.TileWindowsHorizontally
End Sub
```

Voir aussi

Objet Application (Page 1876)

Référence VBA (Page 1724)

Méthode TileWindowsVertically

Description

Dispose verticalement toutes les vues ouvertes dans Graphics Designer.

Syntaxe

Expression.`Methode()`

Expression

Impératif. Une expression qui renvoie un objet du type "Application".

Paramètres

--

Exemple

Dans l'exemple suivant, on dispose verticalement toutes les vues ouvertes dans Graphics Designer. Pour que cet exemple s'applique, vous devez avoir ouvert plusieurs vues dans Graphics Designer :

```
Sub TileWindowsVertically()
'VBA196
```

```
Application.TileWindowsVertically
End Sub
```

Voir aussi

Objet Application (Page 1876)

Référence VBA (Page 1724)

Méthode Ungroup

Description

Dissocie un objet-groupe. Les objets individuels sont conservés.

Syntaxe

Expression.`Ungroup(Parameter)`

Expression

Impératif. Une expression qui renvoie un objet du type "Group".

Paramètres

--

Exemple

Dans l'exemple qui suit, on crée trois objets dans la vue active, puis on les réunit sous forme d'un objet-groupe. L'objet-groupe est ensuite déplacé, puis dissocié.

```
Sub DissolveGroup()
'VBA199
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
.Left = 0
.Selected = True
End With
With objRectangle
.Top = 80
```

```
.Left = 42
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
.Width = 40
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
MsgBox "Group-object is created."
With objGroup
.Left = 120
.Top = 300
MsgBox "Group-object is moved."
.UnGroup
MsgBox "Group is dissolved."
End With
End Sub
```

Voir aussi

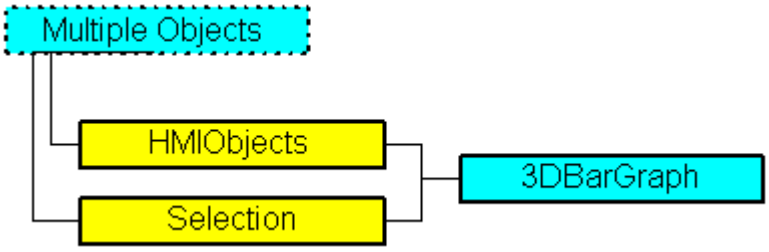
Objet Group (Page 1934)
Méthode CreateGroup (Page 1805)
Référence VBA (Page 1724)
Objets Groupe (Page 1668)

3.5.1.7 Objets et listes

0-9, A-C

Objet 3DBarGraph

Description



Représente l'objet "Bargraphes tridimensionnels". L'objet 3DBarGraph est un élément des liste suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.

Désignation VBA de l'objet

HMI3DBarGraph

Utilisation

Pour créer un nouvel objet "Bargraphes tridimensionnels" dans une vue, on utilise la méthode Add :

```
Sub Add3DBarGraph()
'VBA200
Dim obj3DBarGraph As HMI3DBarGraph
Set obj3DBarGraph = ActiveDocument.HMIObjects.AddHMIObject("3DBar", "HMI3DBarGraph")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub Edit3DBarGraph()
'VBA201
Dim obj3DBarGraph As HMI3DBarGraph
Set obj3DBarGraph = ActiveDocument.HMIObjects("3DBar")
obj3DBarGraph.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

```
Sub ShowNameOfFirstSelectedObject()
'VBA202
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Propriétés de l'objet

Un objet Bargraphes tridimensionnels possède les propriétés suivantes :

- AngleAlpha
- AngleBeta

- Axe
- BorderColor
- Background
- BarDepth
- BarHeight
- BarWidth
- BaseX
- BaseY
- BorderColor
- BorderStyle
- BorderWidth
- Direction
- Height
- Layer00Checked
- Layer00Color
- Layer00Value
- Layer01Checked
- Layer01Color
- Layer01Value
- Layer02Checked
- Layer02Color
- Layer02Value
- Layer03Checked
- Layer03Color
- Layer03Value
- Layer04Checked
- Layer04Color
- Layer04Value
- Layer05Checked
- Layer05Color
- Layer05Value
- Layer06Checked
- Layer06Color
- Layer06Value
- Layer07Checked

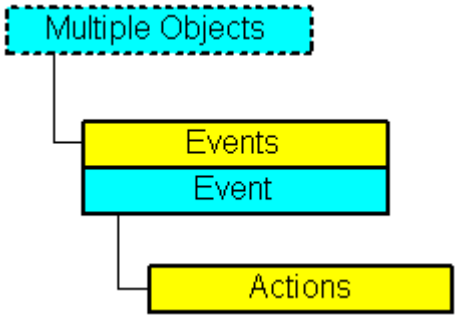
- Layer07Color
- Layer07Value
- Layer08Checked
- Layer08Color
- Layer08Value
- Layer09Checked
- Layer09Color
- Layer09Value
- Layer10Checked
- Layer10Color
- Layer10Value
- Layer
- Left
- LightEffect
- Max
- Min
- Nom
- Opération
- PasswordLevel
- PredefinedAngles
- Process
- ToolTipText
- Top
- Visible
- Width
- ZeroPointValue

Voir aussi

Objet Selection (liste) (Page 2010)
Propriété Layer09Color (Page 2242)
Propriété Layer01Checked (Page 2225)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété ZeroPointValue (Page 2477)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété Process (Page 2323)
Propriété PredefinedAngels (Page 2321)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Min (Page 2286)
Propriété Max (Page 2269)
Propriété LightEffect (Page 2254)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Layer10Value (Page 2245)
Propriété Layer10Color (Page 2244)
Propriété Layer10Checked (Page 2243)
Propriété Layer09Value (Page 2243)
Propriété Layer09Checked (Page 2241)
Propriété Layer08Value (Page 2241)
Propriété Layer08Color (Page 2240)
Propriété Layer08Checked (Page 2239)
Propriété Layer07Value (Page 2239)
Propriété Layer07Color (Page 2238)
Propriété Layer07Checked (Page 2237)
Propriété Layer06Value (Page 2237)
Propriété Layer06Color (Page 2236)
Propriété Layer06Checked (Page 2235)
Propriété Layer05Value (Page 2235)
Propriété Layer05Color (Page 2234)
Propriété Layer05Checked (Page 2233)

Objet Actions (liste)

Description



Représente une liste d'actions liées à un événement.

Désignation VBA de l'objet

HMIActions

Utilisation

Pour configurer une ou plusieurs actions liées à un événement, on utilise la méthode AddAction. Dans l'exemple qui suit, on insère un bouton et un cercle dans la vue active. Dans Runtime, le rayon du cercle augmente à chaque clic sur le bouton :

```
Sub CreateVBActionToClickedEvent()
'VBA203
Dim objButton As HMIButton
Dim objCircle As HMICircle
Dim objVBScript As HMIScriptInfo
Dim strVBCode As String
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_VB", "HMICircle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircle
.Top = 100
.Left = 100
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Text = "Increase Radius"
End With
'define event and assign sourcecode to it:
Set objVBScript = objButton.Events(1).Actions.AddAction(hmiActionCreationTypeVBScript)
strVBCode = "Dim myCircle" & vbCrLf & "Set myCircle = "
strVBCode = strVBCode & "HMIRuntime.ActiveScreen.ScreenItems(""Circle_VB"")"
strVBCode = strVBCode & vbCrLf & "myCircle.Radius = myCircle.Radius + 5"
With objVBScript
```

```
.SourceCode = strVBCode
End With
End Sub
```

Voir aussi

Méthode AddAction (Page 1773)

Configuration d'actions commandées par événement avec VBA (Page 1693)

Propriété Parent (Page 2303)

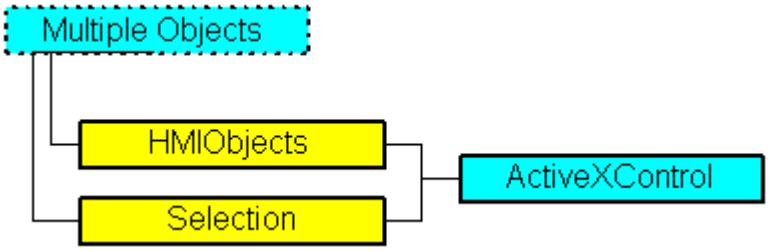
Propriété Item (Page 2213)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet ActiveXControl

Description



Représente l'objet ActiveXControl. L'objet ActiveXControl est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.

Désignation VBA de l'objet

HMIActiveXControl

Utilisation

Pour insérer une commande ActiveX dans une vue, on utilise p. ex. la méthode AddActiveXControl. Dans l'exemple qui suit, le contrôle ActiveX "WinCC Gauge Control" est insérée dans la vue active :

```
Sub AddActiveXControl()
'VBA204
Dim objActiveXControl As HMIActiveXControl
```

```
Set objActiveXControl = ActiveDocument.HMIObjects.AddActiveXControl("WinCC_Gauge",
"XGAUGE.XGaugeCtrl.1")
With ActiveDocument
.HMIObjects("WinCC_Gauge").Top = 40
.HMIObjects("WinCC_Gauge").Left = 40
End With
End Sub
```

Voir aussi

Propriété ServerName (Page 2350)

Méthode AddActiveXControl (Page 1774)

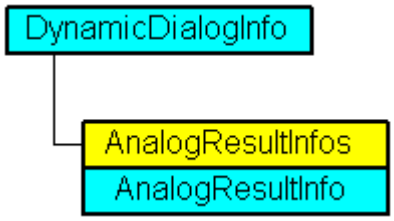
Référence VBA (Page 1724)

ActiveX Controls (Page 1663)

Propriété ProgID (Page 2324)

Objet AnalogResultInfo

Description



Représente une plage de valeurs analogiques et la valeur de propriété associée dans le dialogue Dynamique. L'objet AnalogResultInfo est un élément du listage AnalogResultInfos.

Désignation VBA de l'objet

HMIAnalogResultInfo

Utilisation

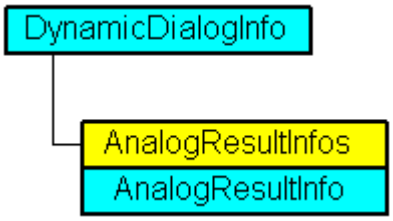
Pour renvoyer la plage de valeurs d'une valeur de propriété, on utilise l'objet AnalogResultInfo. Vous trouverez un exemple détaillé dans cette documentation à la rubrique "Objet AnalogResultInfos (listage)".

Voir aussi

Objet AnalogResultInfos (liste) (Page 1875)
Méthode Delete (Page 1807)
Propriété Value (Page 2395)
Propriété RangeTo (Page 2332)
Propriété Parent (Page 2303)
Propriété Application (Page 2067)

Objet AnalogResultInfos (liste)

Description



Listage des objets AnalogResultInfo qui renferment toutes les plages de valeurs analogiques et leur valeur de propriété associée dans le dialogue Dynamique.

Désignation VBA de l'objet

HMIAnalogResultInfos

Utilisation

Pour ajouter une nouvelle plage de valeurs par le dialogue Dynamique, on utilise la méthode Add. Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on crée trois plages de valeurs analogiques :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA206
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.Add 50, 40
.AnalogResultInfos.Add 100, 80
.AnalogResultInfos.ElseCase = 100
End With
```

```
End Sub
```

Pour renvoyer le listage AnalogResultInfos, on utilise AnalogResultInfos. Dans cet exemple, les plages de valeurs créées dans l'exemple précédent sont sorties :

```
Sub ShowAnalogResultInfosOfCircleRadius()
'VBA207
Dim colAResultInfos As HMIAnalogResultInfos
Dim objAResultInfo As HMIAnalogResultInfo
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Dim iAnswer As Integer
Dim varRange As Variant
Dim varValue As Variant
Set objCircle = ActiveDocument.HMIObjects("Circle_A")
Set objDynDialog = objCircle.Radius.Dynamic
Set colAResultInfos = objDynDialog.AnalogResultInfos
For Each objAResultInfo In colAResultInfos
varRange = objAResultInfo.RangeTo
varValue = objAResultInfo.value
iAnswer = MsgBox("Ranges of values from Circle_A-Radius:" & vbCrLf & "Range of value to: "
& varRange & vbCrLf & "Value of property: " & varValue, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objAResultInfo
End Sub
```

Voir aussi

Méthode Add (listage AnalogResultInfos) (Page 1766)

Propriété Parent (Page 2303)

Propriété Item (Page 2213)

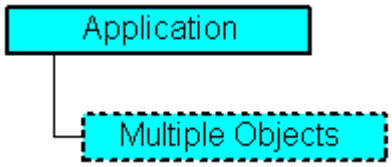
Propriété ElseCase (Page 2157)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet Application

Description



Représente l'Editeur Graphics Designer. L'objet Application renferme les propriétés et les méthodes renvoyées par les objets du niveau principal. Par exemple, ActiveDocument renvoie un objet Document.

Désignation VBA de l'objet

HMIApplication

Utilisation

On utilise Application pour renvoyer l'objet application. Dans l'exemple qui suit, la version de l'application est sortie :

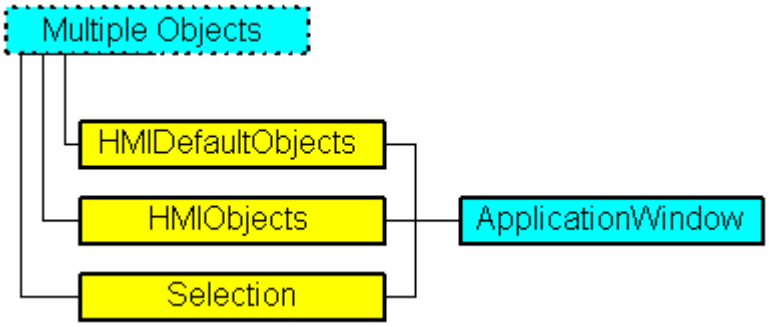
```
Sub ShowApplicationVersion()
'VBA208
MsgBox Application.Version
End Sub
```

Voir aussi

Méthode ShowTagDialog (Page 1864)
Propriété CurrentDesktopLanguage (Page 2144)
Méthode TileWindowsVertically (Page 1865)
Méthode TileWindowsHorizontally (Page 1864)
Méthode ShowSymbolLibraryDialog (Page 1863)
Méthode ShowPropertiesDialog (Page 1862)
Méthode SaveDefaultConfig (Page 1855)
Méthode LoadDefaultConfig (Page 1838)
Méthode CascadeWindows (Page 1793)
Méthode ArrangeMinimizedWindows (Page 1790)
Méthode Activate (Page 1765)
Référence VBA (Page 1724)
Propriété WindowState (Page 2476)
Propriété Visible (Page 2469)
Propriété Version (Page 2468)
Propriété VBE (Page 2468)
Propriété VBAVersion (Page 2468)
Propriété SymbolLibraries (Page 2361)
Propriété ProjectType (Page 2325)
Propriété ProjectName (Page 2325)
Propriété ProfileName (Page 2323)
Propriété Parent (Page 2303)
Propriété Name (Page 2289)
Propriété IsConnectedToProject (Page 2211)
Propriété Documents (Page 2155)
Propriété DefaultHMIObjects (Page 2149)
Propriété CustomToolbars (Page 2147)
Propriété CustomMenus (Page 2146)
Propriété CurrentDataLanguage (Page 2143)
Propriété ConfigurationFileName (Page 2141)
Propriété AvailableDataLanguages (Page 2072)
Propriété ApplicationDataPath (Page 2068)
Propriété Application (Page 2067)
Propriété ActiveDocument (Page 2055)

Objet ApplicationWindow

Description



Représente l'objet "Fenêtre d'application". L'objet ApplicationWindow est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIApplicationWindow

Utilisation

Pour créer un nouvel objet "Fenêtre d'application" dans une vue, on utilise la méthode Add :

```
Sub AddApplicationWindow()
'VBA209
Dim objApplicationWindow As HMIApplicationWindow
Set objApplicationWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow",
"HMIApplicationWindow")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditApplicationWindow()
'VBA210
Dim objApplicationWindow As HMIApplicationWindow
Set objApplicationWindow = ActiveDocument.HMIObjects("AppWindow")
objApplicationWindow.Sizeable = True
End Sub
Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :
```

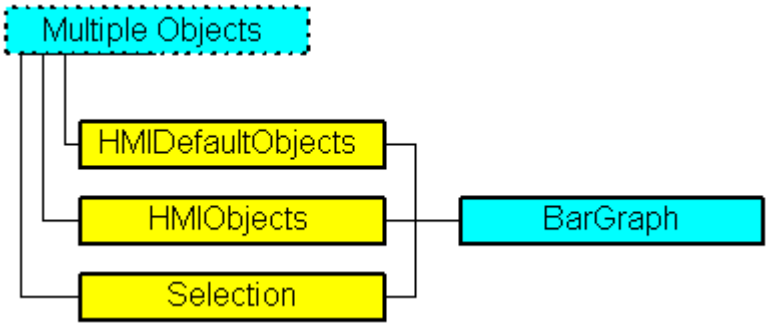
```
Sub ShowNameOfFirstSelectedObject()
'VBA211
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Caption (Page 2113)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété WindowBorder (Page 2473)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété Sizeable (Page 2354)
Propriété OnTop (Page 2297)
Propriété Name (Page 2289)
Propriété Moveable (Page 2289)
Propriété MaximizeButton (Page 2270)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété CloseButton (Page 2125)
Propriété Application (Page 2067)

Objet BarGraph

Description



Représente l'objet "Bargraphe". L'objet BarGraph est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA

HMIBarGraph

Utilisation

Pour créer un nouvel objet "Bargraphe" dans une vue, on utilise la méthode Add :

```
Sub AddBarGraph()
'VBA212
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditBarGraph()
'VBA213
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects("Bar1")
objBarGraph.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

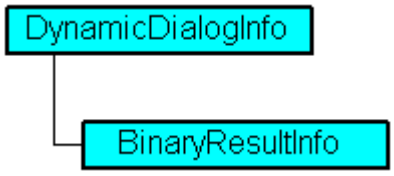
```
Sub ShowNameOfFirstSelectedObject()
'VBA214
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété TypeWarningHigh (Page 2385)
Propriété Max (Page 2269)
Propriété FillColor (Page 2164)
Propriété BorderStyle (Page 2103)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété ZeroPointValue (Page 2477)
Propriété ZeroPoint (Page 2476)
Propriété Width (Page 2472)
Propriété WarningLow (Page 2471)
Propriété WarningHigh (Page 2470)
Propriété Visible (Page 2469)
Propriété TypeWarningLow (Page 2386)
Propriété TypeToleranceLow (Page 2384)
Propriété TypeToleranceHigh (Page 2383)
Propriété TypeLimitLow5 (Page 2383)
Propriété TypeLimitLow4 (Page 2382)
Propriété TypeLimitHigh5 (Page 2381)
Propriété TypeLimitHigh4 (Page 2380)
Propriété TypeAlarmLow (Page 2379)
Propriété TypeAlarmHigh (Page 2379)
Propriété Trend (Page 2376)
Propriété TrendColor (Page 2376)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété ToleranceLow (Page 2371)
Propriété ToleranceHigh (Page 2370)
Propriété ScalingType (Page 2341)
Propriété Scaling (Page 2340)
Propriété ScaleTicks (Page 2340)
Propriété ScaleColor (Page 2339)
Propriété RightComma (Page 2335)
Propriété Process (Page 2323)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)

Objet BinaryResultInfo

Description



Représente les deux plages de valeurs binaires (booléennes) et leur valeur de propriété associée dans le dialogue Dynamique.

Désignation des objets VBA

HMIBinaryResultInfo

Utilisation

On utilise BinaryResultInfo pour renvoyer un objet BinaryResultInfo. Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on affecte une valeur de propriété à chaque plage de valeurs binaire :

```
Sub AddDynamicDialogToCircleRadiusTypeBinary()
'VBA215
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_C", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBool
.BinaryResultInfo.NegativeValue = 20
.BinaryResultInfo.PositiveValue = 40
End With
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Propriété PositiveValue (Page 2321)

Propriété Parent (Page 2303)

Propriété NegativeValue (Page 2291)

Propriété Application (Page 2067)

Objet BitResultInfo

Description

Représente les deux plages de valeurs applicables à Bit activé/non activé et leur valeur de propriété associée dans le dialogue Dynamique.

Désignation VBA de l'objet

HMIBitResultInfo

Utilisation

On utilise BitResultInfo pour renvoyer un objet BitResultInfo. Dans l'exemple qui suit, on dynamise le rayon d'un cercle à l'aide du dialogue Dynamique, on attribue un nom de variable, on définit le bit à configurer et on affecte la valeur de propriété respective aux états "activé"/"non activé" :

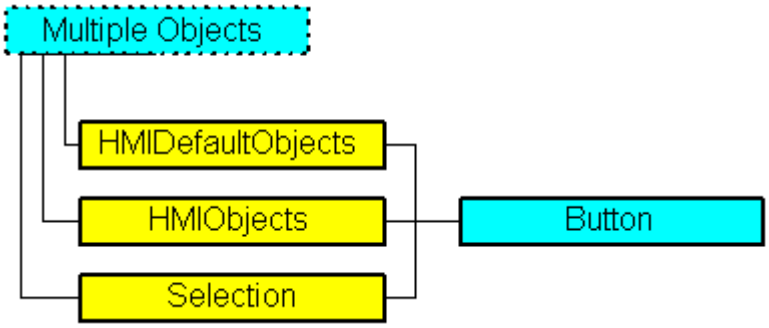
```
Sub AddDynamicDialogToCircleRadiusTypeBit()
'VBA216
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_B", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBit
.BitResultInfo.BitNumber = 1
.BitResultInfo.BitSetValue = 40
.BitResultInfo.BitNotSetValue = 80
End With
End Sub
```

Voir aussi

Méthode Delete (Page 1807)
Référence VBA (Page 1724)
Propriété BitSetValue (Page 2093)
Propriété BitNumber (Page 2092)
Propriété BitNotSetValue (Page 2091)
Propriété Application (Page 2067)

Objet Button

Description



Représente l'objet "Button". L'objet Button est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA

HMIButton

Utilisation

Pour créer un nouvel objet "Button" dans une vue, on utilise la méthode Add :

```
Sub AddButton()
'VBA217
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button", "HMIButton")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditButton()
'VBA218
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects("Button")
objButton.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

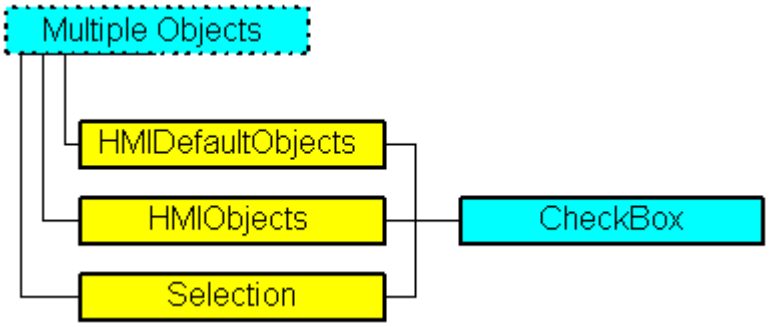
```
Sub ShowNameOfFirstSelectedObject()
'VBA219
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ForeFlashColorOn (Page 2195)
Propriété BorderColorBottom (Page 2098)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété Text (Page 2369)
Propriété PictureUp (Page 2315)
Propriété PictureDown (Page 2314)
Propriété PasswordLevel (Page 2305)
Propriété Orientation (Page 2301)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Hotkey (Page 2204)
Propriété Height (Page 2201)
Propriété ForeFlashColorOff (Page 2194)
Propriété ForeColor (Page 2194)
Propriété FontUnderline (Page 2193)
Propriété FontSize (Page 2192)
Propriété FontName (Page 2191)
Propriété FontItalic (Page 2191)
Propriété FontBold (Page 2188)
Propriété FlashRateForeColor (Page 2185)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashForeColor (Page 2176)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)

Objet CheckBox

Description



Représente l'objet "Case à cocher". L'objet CheckBox est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA

HMICheckBox

Utilisation

Pour créer un nouvel objet "Case à cocher" dans une vue, on utilise la méthode Add :

```
Sub AddCheckBox()
'VBA220
Dim objCheckBox As HMICheckBox
Set objCheckBox = ActiveDocument.HMIObjects.AddHMIObject("CheckBox", "HMICheckBox")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditCheckBox()
'VBA221
Dim objCheckBox As HMICheckBox
Set objCheckBox = ActiveDocument.HMIObjects("CheckBox")
objCheckBox.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

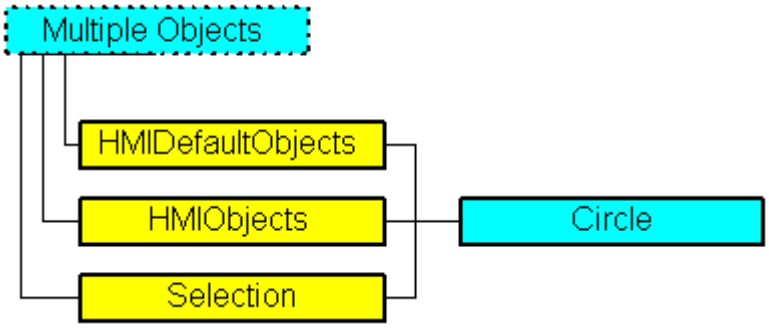
```
Sub ShowNameOfFirstSelectedObject()
'VBA222
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété BitSetValue (Page 2093)
Propriété BitNumber (Page 2092)
Propriété BitNotSetValue (Page 2091)
Propriété Application (Page 2067)

Objet Circle

Description



Représente l'objet "Cercle". L'objet Circle est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMICircle

Utilisation

Pour créer un nouvel objet "Cercle" dans une vue, on utilise la méthode Add :

```
Sub AddCircle()
'VBA223
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle", "HMICircle")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditCircle()
'VBA224
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects("Circle")
objCircle.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

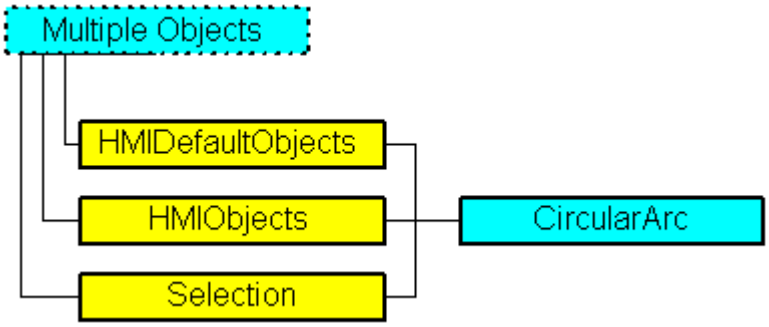
```
Sub ShowNameOfFirstSelectedObject()
'VBA225
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété FillColor (Page 2164)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété Radius (Page 2330)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackFlashColorOn (Page 2082)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)

Objet CircularArc

Description



Représente l'objet "Arc de cercle". L'objet CircularArc est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMICircularArc

Utilisation

Pour créer un nouvel objet "Arc de cercle" dans une vue, on utilise la méthode Add :

```
Sub AddCiruclarArc()
'VBA226
Dim objCiruclarArc As HMICircularArc
Set objCiruclarArc = ActiveDocument.HMIObjects.AddHMIObject("CircularArc",
"HMICircularArc")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditCiruclarArc()
'VBA227
Dim objCiruclarArc As HMICircularArc
Set objCiruclarArc = ActiveDocument.HMIObjects("CircularArc")
objCiruclarArc.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

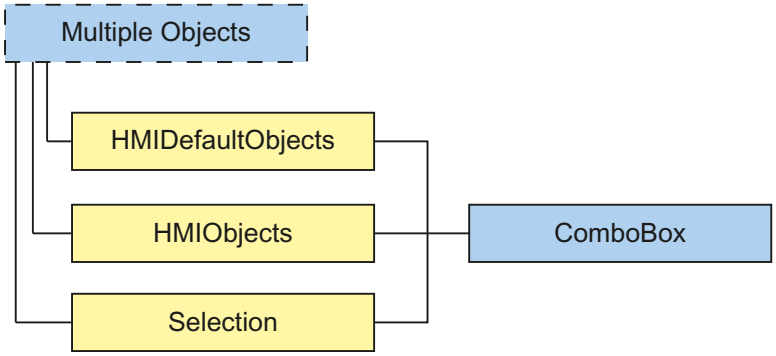
```
Sub ShowNameOfFirstSelectedObject()
'VBA228
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Objets HMIObjects (liste) (Page 1945)
Propriété BorderBackColor (Page 2095)
Objet Selection (liste) (Page 2010)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété StartAngle (Page 2359)
Propriété Radius (Page 2330)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashBorderColor (Page 2174)
Propriété EndAngle (Page 2158)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)

Objet ComboBox

Description



Représente l'objet "Champ de combinaison". L'objet ComboBox est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMIComboBox

Utilisation

Pour créer un nouvel objet "Champ de combinaison" dans la vue, on utilise la méthode Add :

```
Sub AddComboBox()
'VBA822
Dim objComboBox As HMIComboBox
Set objComboBox = ActiveDocument.HMIObjects.AddHMIObject("Champ de combinaison",
"HMIComboBox")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditComboBox()
'VBA850
Dim objComboBox As HMIComboBox
Set objComboBox = ActiveDocument.HMIObjects("Champ de combinaison")
objComboBox.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

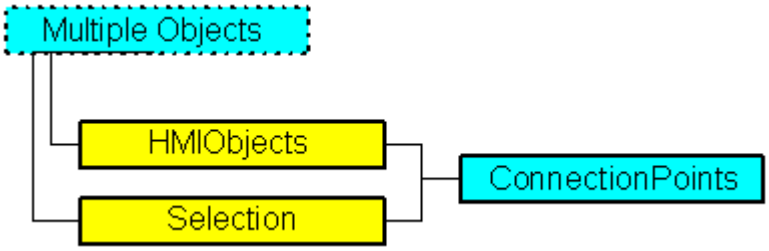
```
Sub ShowNameOfFirstSelectedObject()
'VBA824
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ObjectName (Page 2293)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Top (Page 2374)
Propriété Width (Page 2472)
Propriété Height (Page 2201)
Propriété NumberLines (Page 2292)
Propriété ForeColor (Page 2194)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackColor (Page 2076)
Propriété BorderStyle (Page 2103)
Propriété BorderWidth (Page 2105)
Propriété FillColor (Page 2164)
Propriété FillStyle (Page 2169)
Propriété FontName (Page 2191)
Propriété FontSize (Page 2192)
Propriété FontBold (Page 2188)
Propriété FontItalic (Page 2191)
Propriété FontUnderline (Page 2193)
Propriété AlignmentLeft (Page 2063)
Propriété GlobalShadow (Page 2197)
Propriété Index (Page 2207)
Propriété Text (Page 2369)
Propriété Operation (Page 2298)
Propriété PasswordLevel (Page 2305)
Propriété Visible (Page 2469)
Propriété TooltipText (Page 2373)
Propriété OperationMessage (Page 2299)
Propriété OperationReport (Page 2300)
Propriété SelText (Page 2349)
Propriété SelIndex (Page 2349)

Objet ConnectionPoints (liste)

Description



Ce listage renvoie le nombre de points de l'objet indiqué auxquels un connecteur peut s'accrocher.

Désignation VBA de l'objet

HMIConnectionPoints

Propriétés de l'objet

Un objet ConnectionPoints possède les propriétés suivantes :

- Application
- Count
- Item
- Parent

Exemple 1

Dans cet exemple, un rectangle est ajouté et le nombre de points de liaison du rectangle est indiqué :

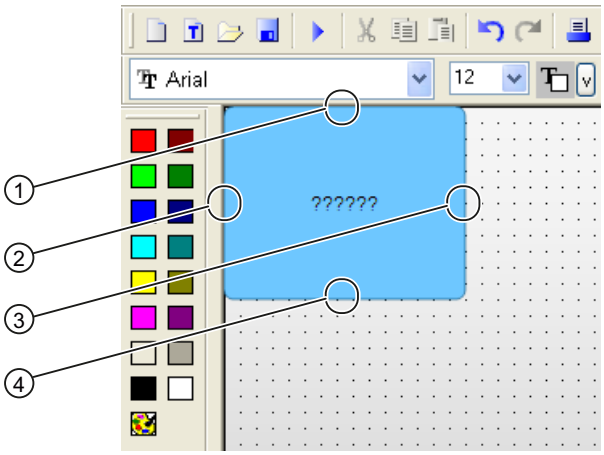
```
Sub CountConnectionPoints()
'VBA229
Dim objRectangle As HMIRectangle
Dim objConnPoints As HMIConnectionPoints
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
Set objConnPoints = ActiveDocument.HMIObjects("Rectangle1").ConnectionPoints
MsgBox "Rectangle1 has " & objConnPoints.Count & " connectionpoints."
End Sub
```

Exemple 2 :

Dans cet exemple, on ajoute une zone de texte et on accède aux points de liaison via "ConnectionPoints.Item". Les coordonnées des points de liaison sont affichées dans une fenêtre de sortie.

```
Sub GetConnectionPoints()
'VBA825
Dim xPos As Long
Dim yPos As Long
Dim objConnPoints As HMIConnectionPoints

Set objDoc = Application.ActiveDocument
Set objObject = objDoc.HMIObjects.AddHMIObject("Text","HMIStaticText")
Set objConnPoints = ActiveDocument.HMIObjects("Text").ConnectionPoints

For i = 1 To objConnPoints.Count
    xPos = objObject.ConnectionPoints.Item(i)(0)
    yPos = objObject.ConnectionPoints.Item(i)(1)
    MsgBox "Coordinates " & i & ". ConnectionPoint:" & Chr(13) & "x: " & xPos & Chr(13) &
"y: " & yPos
Next

End Sub
```

La vue suivante montre la position des 4 points de liaison de la zone de texte :



Remarque

Si vous accédez aux points de liaison d'un connecteur avec VBA, l'index de point de liaison commence par "1".

Si vous définissez les points de liaison dans la fenêtre des propriétés du connecteur dans l'interface graphique, l'index de point de liaison commence par "0".

Les numéros d'index comme par ex. celui du point de liaison inférieur sur la vue sont attribués comme suit :

- VBA : Indice = 3
- interface graphique : Indice = 2

Voir aussi

Propriété Parent (Page 2303)

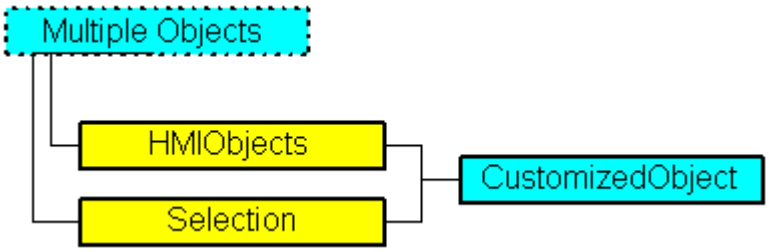
Propriété Item (Page 2213)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet CustomizedObject

Description



Représente l'objet "Objet utilisateur". L'objet CustomizedObject est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.

Pour un objet CustomizedObject, les seules propriétés d'objet disponibles sont celles que vous avez sélectionnées dans le "Dialogue de configuration" de l'objet utilisateur.

Remarque

VBA ne permet pas de configurer un objet CustomizedObject.

Vous trouverez des renseignements supplémentaires sur les objets utilisateur dans la documentation WinCC à la rubrique "Objet utilisateur".

Désignation VBA

HMICustomizedObject

Utilisation

Pour créer un nouvel "Objet utilisateur" dans une vue, on utilise la méthode CreateCustomizedObject et le listage Selection :

```
Sub CreateCustomizedObject()
'VBA230
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objCustomizedObject As HMICustomizedObject
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
With objCircle
.Left = 10
.Top = 10
.Selected = True
End With
With objRectangle
.Left = 50
.Top = 50
.Selected = True
End With
MsgBox "objects created and selected!"
Set objCustomizedObject = ActiveDocument.Selection.CreateCustomizedObject
objCustomizedObject.ObjectName = "Customer-Object"
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditCustomizedObject()
'VBA231
Dim objCustomizedObject As HMICustomizedObject
Set objCustomizedObject = ActiveDocument.HMIObjects("Customer-Object")
MsgBox objCustomizedObject.ObjectName
End Sub
```

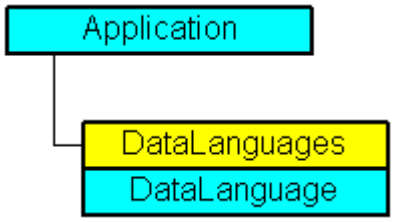
Voir aussi

Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Méthode Destroy (Page 1811)
Méthode Delete (Page 1807)
Méthode CreateCustomizedObject (Page 1802)
Pour éditer un objet utilisateur avec VBA (Page 1677)
Référence VBA (Page 1724)
Objets utilisateur (Page 1676)
Edition d'objets avec VBA (Page 1653)

D-I

Objet DataLanguage

Description



Représente la langue de travail configurée, identifiée par son nom et l'identifiant de la langue. L'objet DataLanguage appartient au listage DataLanguages.

Vous trouverez dans la documentation WinCC la liste des codes de langue (Index > Language Code). Vous devez convertir la valeur hexadécimale qui y est indiquée en sa valeur décimale.

Désignation VBA de l'objet

HMIDataLanguage

Utilisation

On utilise la propriété DataLanguages pour renvoyer un objet DataLanguage et un seul. Dans l'exemple qui suit, la langue de travail qui a été installée en premier est sortie :

```
Sub ShowFirstObjectOfCollection()
'VBA232
Dim strName As String
```

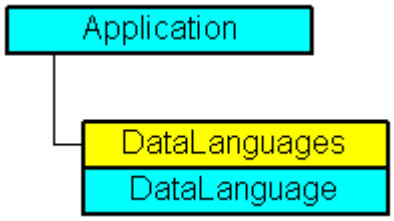
```
strName = ActiveDocument.Application.AvailableDataLanguages(1).LanguageName
MsgBox strName
End Sub
```

Voir aussi

Objet DataLanguages (liste) (Page 1903)
Référence VBA (Page 1724)
Configuration multilingue avec VBA (Page 1617)
Propriété Parent (Page 2303)
Propriété LanguageName (Page 2220)
Propriété LanguageID (Page 2219)
Propriété Application (Page 2067)

Objet DataLanguages (liste)

Description



Listage de tous les objets DataLanguage représentant les langues de travail configurées.

Désignation VBA de l'objet

HMIDataLanguages

Utilisation

Pour renvoyer le listage DataLanguages, on utilise la propriété AvailableDataLanguages. Dans l'exemple qui suit, les langues de travail installées sont sorties :

```
Sub ShowDataLanguage()
'VBA233
Dim colDataLanguages As HMIDataLanguages
Dim objDataLanguage As HMIDataLanguage
Dim strLanguages As String
Dim iCount As Integer
iCount = 0
```

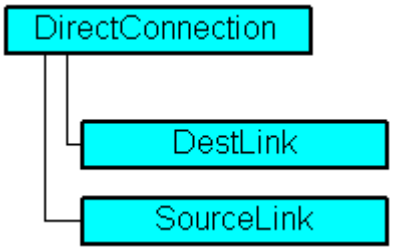
```
Set colDataLanguages = Application.AvailableDataLanguages
For Each objDataLanguage In colDataLanguages
If "" <> strLanguages Then strLanguages = strLanguages & "/"
strLanguages = strLanguages & objDataLanguage.LanguageName & " "
'Every 15 items of datalanguages output in a messagebox
If 0 = iCount Mod 15 And 0 <> iCount Then
MsgBox strLanguages
strLanguages = ""
End If
iCount = iCount + 1
Next objDataLanguage
MsgBox strLanguages
End Sub
```

Voir aussi

Configuration multilingue avec VBA (Page 1617)
Objet DataLanguage (Page 1902)
Méthode Item (Page 1834)
Référence VBA (Page 1724)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

Objet DestLink

Description



Représente la cible d'une liaison directe.

Désignation VBA de l'objet

HMIDestLink

Utilisation

On utilise la propriété DestinationLink pour renvoyer un objet DestLink. Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

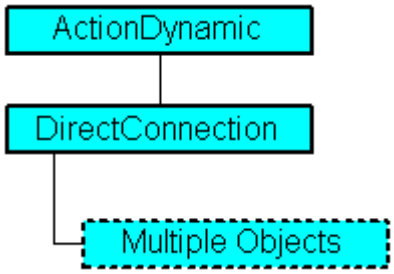
```
Sub SetZoomAndScrollPositionInActiveView()
'VBA234
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDirConnection As HMIDirectConnection
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 90
.Height = 50
.Text = "SetPosition"
End With
'
'Directconnection is initiated on mouseclick:
Set objDirConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDirConnection
'Sourceobject: Property "Top" of "Rectangle_A"
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Targetobject: Property "Left" of "Rectangle_B"
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
End Sub
```

Voir aussi

Objet DirectConnection (Page 1906)
Référence VBA (Page 1724)
Propriété Type (Page 2378)
Propriété ObjectName (Page 2293)
Propriété AutomationName (Page 2070)
Propriété DestinationLink (Page 2150)

Objet DirectConnection

Description



Représente une liaison directe.

Désignation VBA de l'objet

HMIDirectConnection

Utilisation

Pour configurer l'origine et la cible d'une liaison directe, on utilise les propriétés DestinationLink et SourceLink. Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub DirectConnection()
'VBA235
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDirConnection As HMIDirectConnection
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
```

```
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 90
.Height = 50
.Text = "SetPosition"
End With
'
'Directconnection is initiated on mouseclick:
Set objDirConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDirConnection
'Sourceobject: Property "Top" of "Rectangle_A"
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Targetobject: Property "Left" of "Rectangle_B"
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
End Sub
```

Voir aussi

Propriété DestinationLink (Page 2150)

Objet SourceLink (Page 2016)

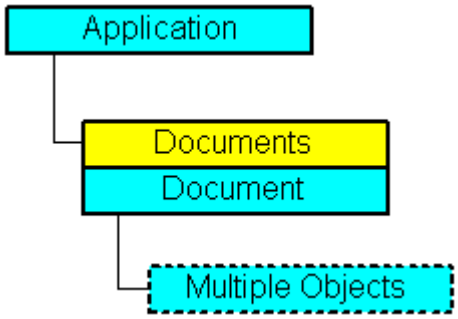
Objet DestLink (Page 1904)

Référence VBA (Page 1724)

Propriété SourceLink (Page 2356)

Objet Document

Description



Représente une vue dans Graphics Designer. L'objet Document appartient au listage Documents.

Désignation VBA de l'objet

HMIDocument

Utilisation

On utilise Documents(Index) pour renvoyer un objet Document individuel. Dans l'exemple qui suit, le nom de fichier de la première vue est affiché :

```
Sub ShowFirstObjectOfCollection()
'VBA236
Dim strName As String
strName = Application.Documents(3).Name
MsgBox strName
End Sub
```

Vous pouvez également utiliser l'objet "Me" si vous voulez accéder au document actuel :

```
Sub ShowDocumentName()
'VBA812
Dim obj As Document
set obj = Me
MsgBox obj.Name
End Sub
```

Vous pouvez p. ex. utiliser la méthode SaveAs pour enregistrer la vue sous un autre nom. Dans l'exemple qui suit, la première vue est enregistrée sous le nom "Copie de sauvegarde Vue1" :

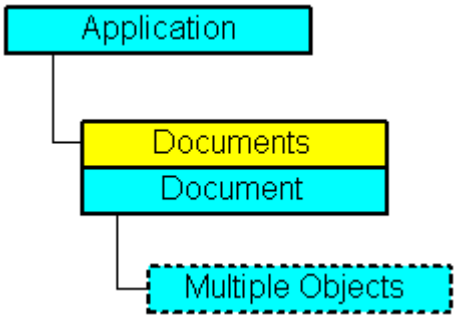
```
Sub SaveDocumentAs()
'VBA237
Application.Documents(3).SaveAs ("CopyOfPicture1")
End Sub
```

Voir aussi

Edition de vues avec VBA (Page 1647)
Propriété GridHeight (Page 2198)
Objet Documents (liste) (Page 1911)
Méthode SetRTLayerVisible (Page 1861)
Méthode SaveAs (Page 1854)
Méthode Save (Page 1853)
Méthode PrintProjectDocumentation (Page 1845)
Méthode PasteClipboard (Page 1843)
Méthode MoveSelection (Page 1841)
Méthode IsRTLayerVisible (Page 1833)
Méthode Export (Page 1815)
Méthode CopySelection (Page 1800)
Méthode Close (Page 1797)
Référence VBA (Page 1724)
Propriété ExtendedZoomingEnable (Page 2163)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Views (Page 2469)
Propriété UpdateCycle (Page 2390)
Propriété TabOrderOtherAction (Page 2365)
Propriété TabOrderMouse (Page 2365)
Propriété TabOrderKeyboard (Page 2364)
Propriété TabOrderAllHMIObjects (Page 2363)
Propriété SnapToGrid (Page 2355)
Objet HMIDefaultObjects (liste) (Page 1939)
Propriété Selection (Page 2348)
Propriété Properties (Page 2326)
Propriété Path (Page 2306)
Propriété PasswordLevel (Page 2305)
Propriété Parent (Page 2303)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété LockedByCreatorID (Page 2261)
Propriété LastChange (Page 2222)
Propriété HMIObjects (Page 2203)
Propriété Hide (Page 2202)
Propriété Height (Page 2201)
Propriété GridWidth (Page 2199)
Propriété GridColor (Page 2198)
Propriété Grid (Page 2197)

Objet Documents (liste)

Description



Désignation VBA de l'objet

HMIDocuments

Utilisation

Remarque

On utilise la propriété "ActiveDocument" pour se reporter à la vue active.

Pour renvoyer le listage Documents, on utilise la propriété Documents. Dans l'exemple qui suit, le nom de toutes les vues ouvertes est sorti :

```
Sub ShowDocuments()
'VBA238
Dim colDocuments As Documents
Dim objDocument As Document
Set colDocuments = Application.Documents
For Each objDocument In colDocuments
MsgBox objDocument.Name
Next objDocument
End Sub
```

Pour ajouter un nouvel objet Document au listage Documents, on utilise la propriété Add. Dans l'exemple qui suit, on crée une nouvelle vue :

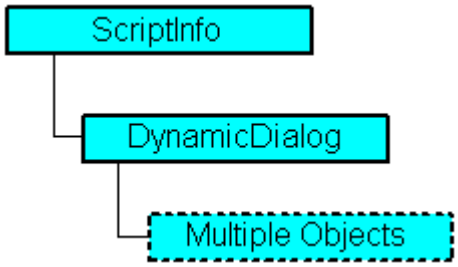
```
Sub AddNewDocument()
'VBA239
Dim objDocument As Document
Set objDocument = Application.Documents.Add
End Sub
```

Voir aussi

Méthode Add (Page 1765)
Objet Document (Page 1908)
Méthode SaveAll (Page 1854)
Méthode Open (Page 1842)
Méthode Item (Page 1834)
Méthode CloseAll (Page 1798)
Méthode Close (Page 1797)
Référence VBA (Page 1724)
Edition de vues avec VBA (Page 1647)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)
Propriété Application (Page 2067)
Propriété ActiveDocument (Page 2055)

Objet DynamicDialog

Description



Représente le dialogue Dynamique. Le dialogue Dynamique permet de dynamiser les propriétés des vues et des objets en fonction de diverses plages de valeurs.

La plage de valeurs est définie à l'aide de la propriété ResultType.

Désignation VBA de l'objet

HMIDynamicDialog

Utilisation

On utilise l'objet DynamicDialog pour dynamiser une propriété d'objet à l'aide du dialogue Dynamique. Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on crée trois plages de valeurs analogiques :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA240
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.Trigger.VariableTriggers.Add "NewDynamic2", hmiVariableCycleType_5s
.AnalogResultInfos.Add 50, 40
.AnalogResultInfos.Add 100, 80
.AnalogResultInfos.ElseCase = 100
End With
End Sub
```

Voir aussi

Méthode Delete (Page 1807)

Méthode ConvertToScript (Page 1799)

Méthode CheckSyntax (Page 1796)

Propriété VariableStateValues (Page 2465)

Propriété VariableStateChecked (Page 2464)

Propriété Trigger (Page 2377)

Propriété SourceCode (Page 2357)

Propriété ScriptType (Page 2342)

Propriété ResultType (Page 2335)

Propriété Parent (Page 2303)

Propriété ActualPointTop (Page 2140)

Propriété BitResultInfo (Page 2092)

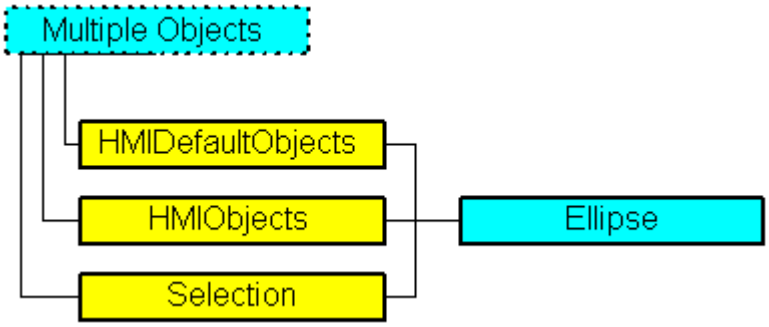
Propriété BinaryResultInfo (Page 2091)

Propriété Application (Page 2067)

Propriété AnalogResultInfos (Page 2065)

Objet Ellipse

Description



Représente l'objet "Ellipse". L'objet Ellipse est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIEllipse

Utilisation

Pour créer un nouvel objet "Ellipse" dans une vue, on utilise la méthode Add :

```
Sub AddEllipse()
'VBA241
Dim objEllipse As HMIEllipse
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("Ellipse", "HMIEllipse")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditEllipse()
'VBA242
Dim objEllipse As HMIEllipse
Set objEllipse = ActiveDocument.HMIObjects("Ellipse")
objEllipse.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

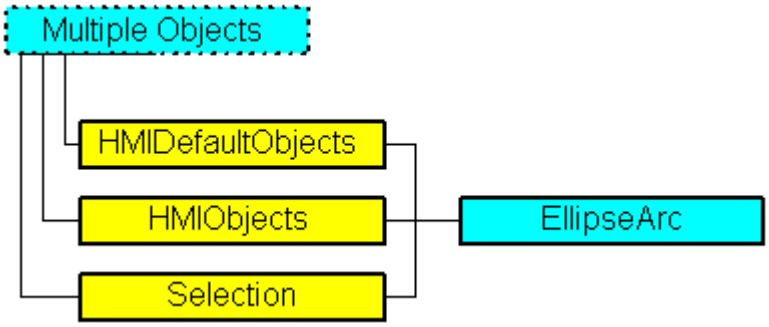
```
Sub ShowNameOfFirstSelectedObject()
'VBA243
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété FillingIndex (Page 2167)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété RadiusWidth (Page 2331)
Propriété RadiusHeight (Page 2331)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackFlashColorOn (Page 2082)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)

Objet EllipseArc

Description



Représente l'objet "Arc d'ellipse". L'objet Arc d'ellipse est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIEllipseArc

Utilisation

Pour créer un nouvel objet "Arc d'ellipse" dans une vue, on utilise la méthode Add :

```
Sub AddEllipseArc()
'VBA244
Dim objEllipseArc As HMIEllipseArc
Set objEllipseArc = ActiveDocument.HMIObjects.AddHMIObject("EllipseArc", "HMIEllipseArc")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditEllipseArc()
'VBA245
Dim objEllipseArc As HMIEllipseArc
Set objEllipseArc = ActiveDocument.HMIObjects("EllipseArc")
objEllipseArc.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

```
Sub ShowNameOfFirstSelectedObject()
'VBA246
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Pour renvoyer un objet du listage HMIDefaultObjects, on utilise "HMIDefaultObjects(Index)" :

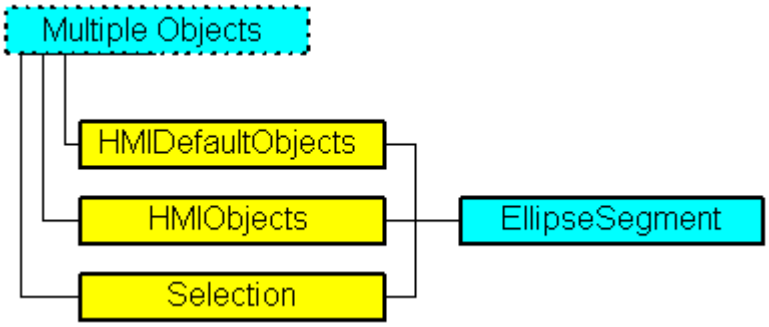
```
Sub EditDefaultPropertiesOfEllipseArc()
'VBA247
Dim objEllipseArc As HMIEllipseArc
Set objEllipseArc = Application.DefaultHMIObjects("HMIEllipseArc")
objEllipseArc.BorderColor = RGB(255, 255, 0)
'create new "EllipseArc"-object
Set objEllipseArc = ActiveDocument.HMIObjects.AddHMIObject("EllipseArc2", "HMIEllipseArc")
End Sub
```

Voir aussi

Propriété TooltipText (Page 2373)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété StartAngle (Page 2359)
Propriété RadiusWidth (Page 2331)
Propriété RadiusHeight (Page 2331)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashBorderColor (Page 2174)
Propriété EndAngle (Page 2158)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)

Objet EllipseSegment

Description



Représente l'objet "Segment d'ellipse". L'objet Segment d'ellipse est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIEllipseSegment

Utilisation

Pour créer un nouvel objet "Segment d'ellipse" dans une vue, on utilise la méthode Add :

```
Sub AddEllipseSegment()
'VBA248
Dim objEllipseSegment As HMIEllipseSegment
Set objEllipseSegment = ActiveDocument.HMIObjects.AddHMIObject("EllipseSegment",
"HMIEllipseSegment")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditEllipseSegment()
'VBA249
Dim objEllipseSegment As HMIEllipseSegment
Set objEllipseSegment = ActiveDocument.HMIObjects("EllipseSegment")
objEllipseSegment.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

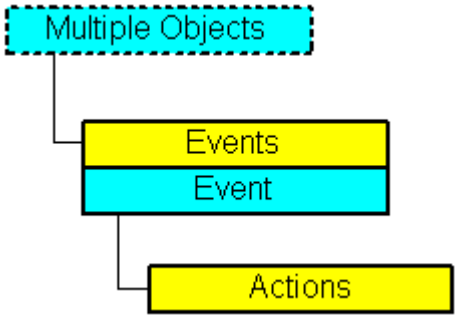
```
Sub ShowNameOfFirstSelectedObject()
'VBA250
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété TooltipText (Page 2373)
Propriété BackFlashColorOn (Page 2082)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété StartAngle (Page 2359)
Propriété RadiusWidth (Page 2331)
Propriété RadiusHeight (Page 2331)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété EndAngle (Page 2158)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)

Objet Event

Description



Représente un événement qui déclenche une ou plusieurs actions dans Runtime (p. ex. une liaison directe). Un événement peut être configuré sur un objet ou sur une propriété.

Désignation VBA de l'objet

HMIEvent

Utilisation

Pour configurer une action déclenchée par un événement, on utilise la méthode AddAction. Dans l'exemple qui suit, une modification de rayon doit déclencher une action en C dans Runtime :

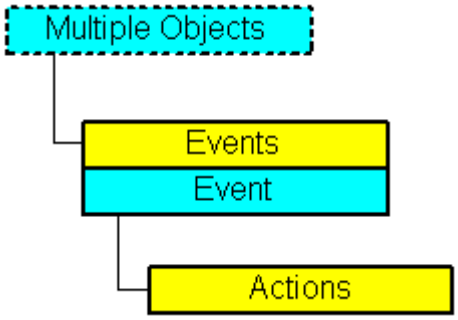
```
Sub AddActionToPropertyTypeCScript()
'VBA251
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
'Create circle in the picture. If property "Radius" is changed,
'a C-action is added:
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB", "HMICircle")
Set objEvent = objCircle.Radius.Events(1)
Set objCScript = objEvent.Actions.AddAction(hmiActionCreationTypeCScript)
End Sub
```

Voir aussi

Propriété Application (Page 2067)
Méthode Delete (Page 1807)
Méthode AddAction (Page 1773)
Référence VBA (Page 1724)
Propriété Parent (Page 2303)
Propriété EventType (Page 2161)
Propriété Actions (Page 2054)

Objet Events (liste)

Description



Listage de tous les objets Event représentant des événements configurés sur un objet. La méthode Item permet de définir le type d'événement à configurer :

- Pour programmer dans VBA une action liée à une propriété, utilisez la propriété "Events(1)", l'"Index 1" représentant l'événement "en cas de modification".
- Pour programmer dans VBA une action liée à un objet, utilisez la propriété "Events(Index)", l'"Index" représentant l'événement déclencheur (voir tableau) :

Indice	EventType (dépend de l'objet utilisé)
0	hmiEventTypeNotDefined
1	hmiEventTypeMouseClick
2	hmiEventTypeMouseLButtonDown
3	hmiEventTypeMouseLButtonUp
4	hmiEventTypeMouseRButtonDown
5	hmiEventTypeMouseRButtonUp
6	hmiEventTypeKeyboardDown
7	hmiEventTypeKeyboardUp
8	hmiEventTypeFocusEnter
9	hmiEventTypeObjectChange
10	hmiEventTypePictureOpen

Désignation VBA de l'objet

HMIEvents

Utilisation

Pour renvoyer un objet Event individuel, on utilise la méthode Item. Dans cet exemple, le nom et le type de tous les événements liés aux objets de la vue active sont sortis. pour que cet exemple s'applique, il faut que quelques objets soient présents dans la vue active et que divers événements soient configurés.

```
Sub ShowEventsOfAllObjectsInActiveDocument()
'VBA252
Dim colEvents As HMIEvents
Dim objEvent As HMIEvent
Dim iMax As Integer
Dim iIndex As Integer
Dim iAnswer As Integer
Dim strEventName As String
Dim strObjectName As String
Dim varEventType As Variant
iIndex = 1
iMax = ActiveDocument.HMIObjects.Count
For iIndex = 1 To iMax
Set colEvents = ActiveDocument.HMIObjects(iIndex).Events
strObjectName = ActiveDocument.HMIObjects(iIndex).ObjectName
For Each objEvent In colEvents
strEventName = objEvent.EventName
varEventType = objEvent.EventType
iAnswer = MsgBox("Objectname: " & strObjectName & vbCrLf & "Eventtype: " & varEventType &
vbCrLf & "Eventname: " & strEventName, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objEvent
If vbCancel = iAnswer Then Exit For
Next iIndex
End Sub
```

Voir aussi

Méthode Item (Page 1834)

Référence VBA (Page 1724)

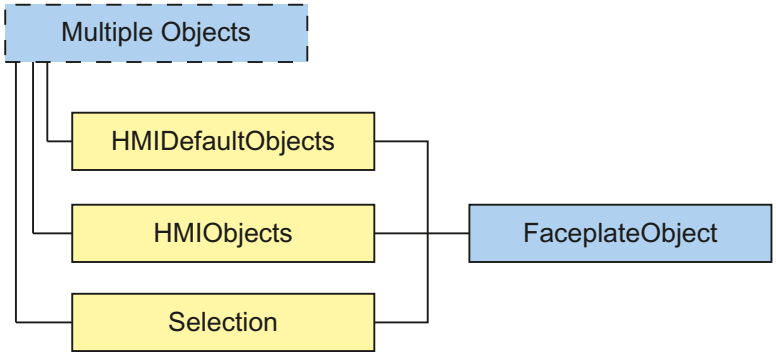
Propriété Parent (Page 2303)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet FaceplateObject

Description



Représente l'objet "Instance de bloc d'affichage". L'objet FaceplateObject est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMIFaceplateObject

Utilisation

Pour créer un nouvel objet "Instance de bloc d'affichage" dans la vue, on utilise la méthode Add :

```
Sub AddFaceplateInstance()
'VBA826
Dim objFaceplateInstance As HMIFaceplateObject
Set objFaceplateInstance = ActiveDocument.HMIObjects.AddHMIObject("Instance de bloc
d'affichage", "HMIFaceplateObject")
objFaceplateInstance.Properties.Item(3).value = "Faceplate1.fpt"
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditFaceplateInstance()
'VBA827
Dim objFaceplateInstance As HMIFaceplateObject
Set objFaceplateInstance = ActiveDocument.HMIObjects("Instance de bloc d'affichage")
```

```
objFaceplateInstance.visible = True
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

```
Sub ShowNameOfFirstSelectedObject()
'VBA828
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ObjectName (Page 2293)

Propriété Layer (Page 2222)

Propriété Left (Page 2253)

Propriété Top (Page 2374)

Propriété Width (Page 2472)

Propriété Height (Page 2201)

Propriété Operation (Page 2298)

Propriété PasswordLevel (Page 2305)

Propriété Visible (Page 2469)

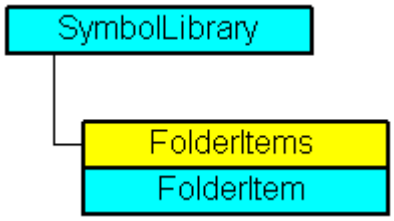
Propriété TooltipText (Page 2373)

Propriété ScalingMode (Page 2341)

Propriété FaceplateType (Page 2164)

Objet FolderItem

Description



Représente un dossier ou un objet de la bibliothèque des modules. Un objet FolderItem du type "Folder" est un élément du listage FolderItems. Un objet FolderItem du type "Item" est un élément du listage Folder.

Désignation VBA de l'objet

HMIFolderItem

Utilisation

Pour renvoyer le listage FolderItems, on utilise la propriété FolderItems. Dans l'exemple qui suit, le nom des dossiers de la "bibliothèque globale" est sorti :

```
Sub ShowFolderItemsOfGlobalLibrary()
'VBA253
Dim colFolderItems As HMIFolderItems
Dim objFolderItem As HMIFolderItem
Set colFolderItems = Application.SymbolLibraries(1).FolderItems
For Each objFolderItem In colFolderItems
MsgBox objFolderItem.Name
Next objFolderItem
End Sub
```

Pour copier un objet "ForlderItem" du type "Item" dans le presse-papiers, on utilise la méthode CopyToClipboard. Dans l'exemple qui suit, l'objet "PC" est copié dans le presse-papiers :

```
Sub CopyFolderItemToClipboard()
'VBA254
Dim objGlobalLib As HMISymbolLibrary
Set objGlobalLib = Application.SymbolLibraries(1)
objGlobalLib.FolderItems("Folder2").Folder("Folder2").Folder.Item("Object1").CopyToClipboa
rd
End Sub
```

Voir aussi

Propriété Type (Page 2378)

Objet FolderItems (liste) (Page 1929)

Méthode Delete (Page 1807)

Méthode CopyToClipboard (Page 1801)

Pour insérer avec VBA dans une vue un objet provenant de la bibliothèque des modules (Page 1645)

Pour utiliser la bibliothèque des modules avec VBA (Page 1643)

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Propriété Parent (Page 2303)

Propriété Name (Page 2289)

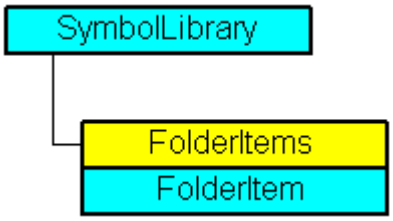
Propriété LDNames (Page 2249)

Propriété Folder (Page 2186)

Propriété Application (Page 2067)

Objet FolderItems (liste)

Description



Listage de tous les objets FolderItem représentant les dossiers et les objets de la bibliothèque des modules.

Désignation VBA de l'objet

HMIFolderItems

Utilisation

Pour renvoyer le listage FolderItems, on utilise la propriété FolderItems. Dans l'exemple qui suit, le nom des dossiers de la "bibliothèque globale" est sorti :

```
Sub ShowFolderItemsOfGlobalLibrary()
'VBA255
```

```
Dim colFolderItems As HMIFolderItems
Dim objFolderItem As HMIFolderItem
Set colFolderItems = Application.SymbolLibraries(1).FolderItems
For Each objFolderItem In colFolderItems
MsgBox objFolderItem.Name
Next objFolderItem
End Sub
```

Vous pouvez utiliser, p. ex. la méthode AddFolder pour créer un nouveau dossier dans la bibliothèque des modules. Dans l'exemple qui suit, le dossier "Dossier projet" est créé dans la "Bibliothèque du projet" :

```
Sub AddNewFolderToProjectLibrary()
'VBA256
Dim objProjectLib As HMISymbolLibrary
Set objProjectLib = Application.SymbolLibraries(2)
objProjectLib.FolderItems.AddFolder ("My Folder")
End Sub
```

Voir aussi

Méthode AddItem (Page 1782)

Objet SymbolLibrary (Page 2023)

Objet FolderItem (Page 1927)

Méthode Item (Page 1834)

Méthode AddFromClipboard (Page 1779)

Méthode AddFolder (Page 1778)

Pour insérer avec VBA dans une vue un objet provenant de la bibliothèque des modules (Page 1645)

Pour utiliser la bibliothèque des modules avec VBA (Page 1643)

Référence VBA (Page 1724)

Accès à la bibliothèque des modules avec VBA (Page 1640)

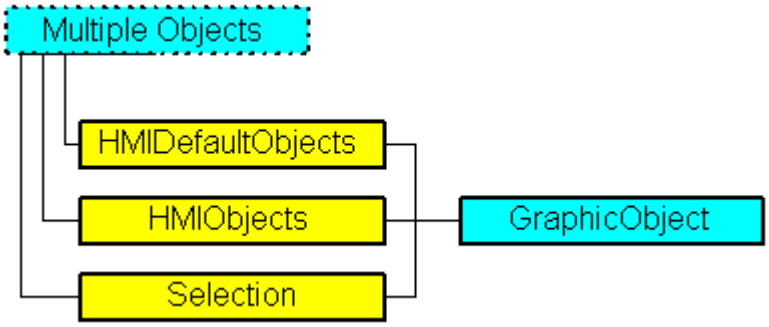
Propriété Parent (Page 2303)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet GraphicObject

Description



Représente l'objet "Objet graphique". L'objet GraphicObject est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIGraphicObject

Utilisation

Pour créer un nouvel objet "Objet graphique" dans une vue, on utilise la méthode Add :

```
Sub AddGraphicObject()
'VBA257
Dim objGraphicObject As HMIGraphicObject
Set objGraphicObject = ActiveDocument.HMIObjects.AddHMIObject("Graphic-Object",
"HMIGraphicObject")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditGraphicObject()
'VBA258
Dim objGraphicObject As HMIGraphicObject
Set objGraphicObject = ActiveDocument.HMIObjects("Graphic-Object")
objGraphicObject.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

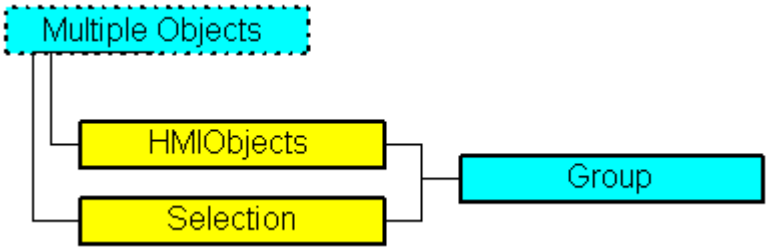
```
Sub ShowNameOfFirstSelectedObject()
'VBA259
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Left (Page 2253)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété PicUseTransColor (Page 2318)
Propriété PictureName (Page 2315)
Propriété PicTransColor (Page 2312)
Propriété PicReferenced (Page 2311)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackFlashColorOn (Page 2082)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)

Objet Group

Description



Représente l'objet "Objet groupe". L'objet Objet groupe est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.

Un objet-groupe est formé à partir des objets sélectionnés dans la vue. Les objets de l'Objet groupe sont en outre enregistrés dans le listage "GroupedHMIObjects", avec de nouveaux numéros d'indice.

Vous pouvez accéder sans restriction aux propriétés de tous les objets d'un Objet groupe.

Vous trouverez des renseignements supplémentaires sur les Objets groupe dans la documentation WinCC à la rubrique "Objet groupe".

Désignation VBA de l'objet

HMIGroup

Utilisation

Pour créer un nouvel "Objet groupe" dans une vue, on utilise la méthode "CreateGroup" et le listage Selection :

```
Sub DoCreateGroup()
'VBA260
Dim objGroup As HMIGroup
Set objGroup = ActiveDocument.Selection.CreateGroup
objGroup.ObjectName = "Group-Object"
End Sub
```

Les méthodes suivantes permettent d'éditer un objet-groupe existant :

- Méthode "Add(Index)" : Ajoute un nouvel objet à l'objet Groupe.
- Méthode "Remove(Index)" : Retire un objet de l'objet Groupe.
- Méthode "UnGroup()" : Dissocie l'objet groupe (annulation du groupage).
- Méthode "Delete()" : Supprime l'objet groupe et les objets qui le composent.

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditGroup()
'VBA261
Dim objGroup As HMIGroup
Set objGroup = ActiveDocument.HMIObjects("Group-Object")
MsgBox objGroup.ObjectName
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Objets HMIObjects (liste) (Page 1945)

Objet GroupedObjects (liste) (Page 1938)

Méthode Ungroup (Page 1866)

Méthode Remove (Page 1846)

Méthode Delete (Page 1807)

Méthode Add (listage GroupedObjects) (Page 1770)

Pour éditer les objets d'un objet Groupe avec VBA (Page 1673)

Pour éditer les objets Groupe avec VBA (Page 1670)

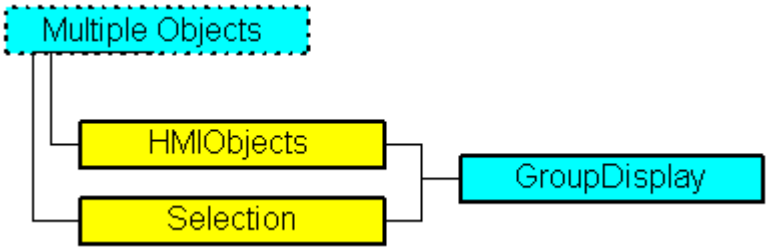
Référence VBA (Page 1724)

Objets Groupe (Page 1668)

Edition d'objets avec VBA (Page 1653)

Objet GroupDisplay

Description



Représente l'objet "Signalisation groupée". L'objet GroupDisplay est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.

Désignation VBA de l'objet

HMIGroupDisplay

Utilisation

Pour créer un nouvel objet "Signalisation groupée" dans une vue, on utilise la méthode Add :

```
Sub AddGroupDisplay()
'VBA262
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("Groupdisplay",
"HMIGroupDisplay")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditGroupDisplay()
'VBA263
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects("Groupdisplay")
objGroupDisplay.BackColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

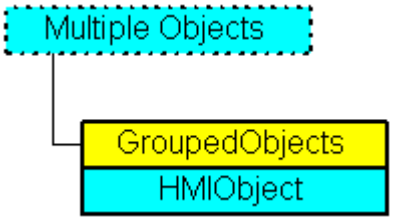
```
Sub ShowNameOfFirstSelectedObject()
'VBA264
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété MCText (Page 2283)
Propriété Height (Page 2201)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété UserValue4 (Page 2394)
Propriété UserValue3 (Page 2393)
Propriété UserValue2 (Page 2392)
Propriété UserValue1 (Page 2391)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété SignificantMask (Page 2353)
Propriété SameSize (Page 2338)
Propriété Relevant (Page 2334)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété MessageClass (Page 2286)
Propriété MCKQTextFlash (Page 2282)
Propriété MCKQTextColorOn (Page 2282)
Propriété MCKQTextColorOff (Page 2281)
Propriété MCKQBackFlash (Page 2280)
Propriété MCKQBackColorOn (Page 2280)
Propriété MCKQBackColorOff (Page 2279)
Propriété MCKOTextFlash (Page 2278)
Propriété MCKOTextColorOn (Page 2278)
Propriété MCKOTextColorOff (Page 2277)
Propriété MCKOBackFlash (Page 2276)
Propriété MCKOBackColorOn (Page 2276)
Propriété MCKOBackColorOff (Page 2275)
Propriété MCGUTextFlash (Page 2274)
Propriété MCGUTextColorOn (Page 2274)
Propriété MCGUTextColorOff (Page 2273)
Propriété MCGUBackFlash (Page 2272)
Propriété MCGUBackColorOn (Page 2272)
Propriété MCGUBackColorOff (Page 2271)

Objet GroupedObjects (liste)

Description



Listage de tous les objets HMIObject représentant les objets d'un objet-groupe.

Désignation VBA de l'objet

HMIGroupedObjects

Utilisation

Pour renvoyer le listage GroupedObjects, on utilise la propriété GroupedHMIObjects. Dans l'exemple qui suit, tous les objets du premier objet-groupe de la vue actuelle sont sortis. Il faut que l' Objet groupe nommé "Groupe1" ait été créé :

```
Sub ShowGroupedObjectsOfFirstGroup()
'VBA265
Dim colGroupedObjects As HMIGroupedObjects
Dim objObject As HMIObject
Set colGroupedObjects = ActiveDocument.HMIObjects("Group1").GroupedHMIObjects
For Each objObject In colGroupedObjects
MsgBox objObject.ObjectName
Next objObject
End Sub
```

Vous pouvez utiliser, p. ex. la méthode Remove pour séparer un objet de l'Objet groupe. Dans l'exemple qui suit, on sépare le premier objet de l'Objet groupe "Groupe1" :

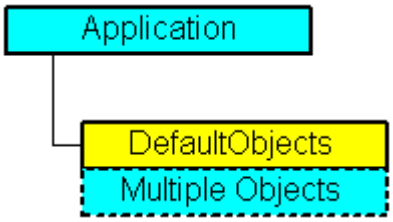
```
Sub RemoveObjectFromGroup()
'VBA266
Dim objGroup As HMIGroup
Set objGroup = ActiveDocument.HMIObjects("Group1")
objGroup.GroupedHMIObjects.Remove (1)
End Sub
```

Voir aussi

Méthode Item (Page 1834)
Objet Group (Page 1934)
Méthode Remove (Page 1846)
Méthode Add (listage GroupedObjects) (Page 1770)
Pour éditer les objets Groupe avec VBA (Page 1670)
Référence VBA (Page 1724)
Objets Groupe (Page 1668)
Propriété Parent (Page 2303)
Propriété GroupedHMIObjects (Page 2200)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

Objet HMIDefaultObjects (liste)

Description



Listage des objets HMIObject suivants :

Objet	Désignation VBA de l'objet
Trait	HMILine
Polygone	HMIPolygon
Trait polygonal	HMIPolyLine
Ellipse	HMIEllipse
Cercle	HMICircle
Segment d'ellipse	HMIEllipseSegment
Segment de cercle	HMIPieSegment
Arc d'ellipse	HMIEllipseArc
Arc de cercle	HMICircularArc
Rectangle	HMIRectangle
Rectangle arrondi	HMIRoundRectangle
Fenêtre d'application	HMIApplicationWindow
Fenêtre de vue	HMIPictureWindow
Texte statique	HMIStaticText

Objet	Désignation VBA de l'objet
Champ d'E/S	HMIIOField
Bouton	HMIButton
Case à cocher	HMICheckBox
Case d'option	HMIOptionGroup
Bouton rond	HMIRoundButton
Bargraphe	HMIBarGraph
Défileur	HMISlider
Objet graphique	HMIGraphicObject
Indicateur d'état	HMIStatusDisplay
Liste de textes	HMITextList
Connecteur	HMIObjConnection
Texte multiligne	HMIMultiLineEdit
Champ de combinaison	HMIComboBox
Champ de liste	HMIListBox
Tuyau polygonal	HMITubePolyline
Raccord en T	HMITubeTeeObject
Double raccord en T	HMITubeDoubleTeeObject
Coude de tuyau	HMITubeArcObject
Bargraphes tridimensionnels	HMI3DBarGraph
Signalisation groupée	HMIGroupDisplay
Instance de bloc d'affichage	HMIFaceplateObject

Désignation VBA de l'objet

HMIDefaultObjects

Utilisation

On utilise la propriété DefaultHMIObjects pour modifier le paramétrage par défaut des valeurs de propriété des objets inclus. Dans cet exemple, tous les objets du listage sont renvoyés :

```
Sub ShowDefaultObjects()
'VBA267
Dim strType As String
Dim strName As String
Dim strMessage As String
Dim iMax As Integer
Dim iIndex As Integer
iMax = Application.DefaultHMIObjects.Count
iIndex = 1
For iIndex = 1 To iMax
With Application.DefaultHMIObjects(iIndex)
strType = .Type
strName = .ObjectName
```

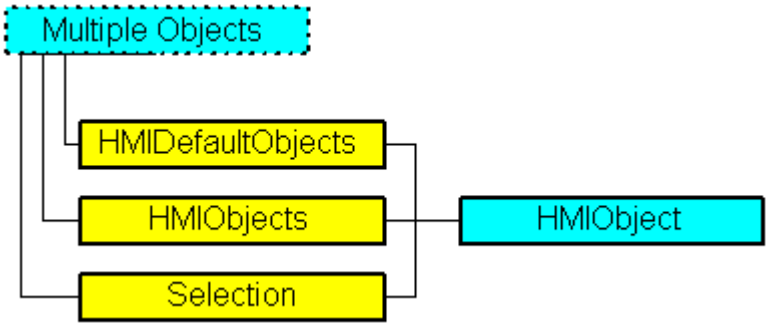
```
strMessage = strMessage & "Element: " & iIndex & " / Objecttype: " & strType & " /
Objectname: " & strName
End With
If 0 = iIndex Mod 10 Then
MsgBox strMessage
strMessage = ""
Else
strMessage = strMessage & vbCrLf & vbCrLf
    End If
Next iIndex
MsgBox "Element: " & iIndex & vbCrLf & "Objecttype: " & strType & vbCrLf & "Objectname: "
& strName
End Sub
```

Voir aussi

Objet Button (Page 1886)
Objet TextList (Page 2025)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet PictureWindow (Page 1980)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet IOField (Page 1947)
Objet EllipseSegment (Page 1920)
Objet EllipseArc (Page 1917)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet BarGraph (Page 1881)
Objet ApplicationWindow (Page 1879)
Méthode Item (Page 1834)
Référence VBA (Page 1724)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)
Propriété DefaultHMIObjects (Page 2149)
Propriété Application (Page 2067)

Objet HMIObject

Description



Représente un objet de la palette des objets de Graphics Designer. L'objet HMIObject est un élément des listages suivants :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Cet objet renferme les propriétés d'objet valables pour tous les objets simples, complexes et Windows (dont Width, Height, Top et Left).

Désignation VBA de l'objet

HMIObject

Utilisation

On peut p. ex. utiliser HMIObjects(Index) pour renvoyer un objet HMIObject individuel. Comme "Index", on peut utiliser le numéro d'index ou le nom de l'objet. Dans l'exemple suivant, le nom du premier objet de la vue active est sorti :

```
Sub ShowFirstObjectOfCollection()
'VBA268
Dim strName As String
strName = ActiveDocument.HMIObjects(1).ObjectName
MsgBox strName
End Sub
```

Pour supprimer un objet du listage HMIObjects, on utilise la méthode Delete. Dans l'exemple qui suit, on supprime le premier objet de la vue active :

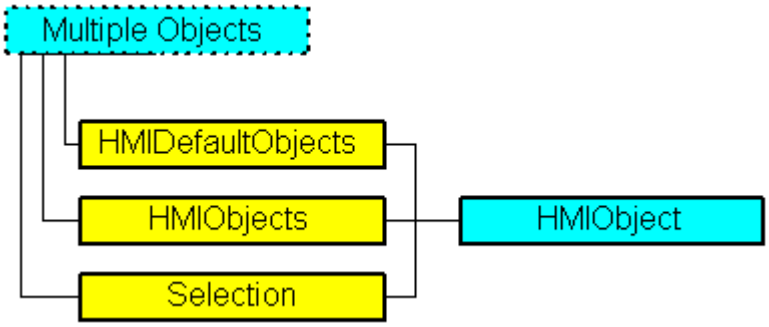
```
Sub DeleteObject()
'VBA269
ActiveDocument.HMIObjects(1).Delete
End Sub
```

Voir aussi

Propriété Name (Page 2289)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode Delete (Page 1807)
Référence VBA (Page 1724)
Objets standard, complexes, Windows et tuyaux (Page 1655)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Type (Page 2378)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété TabOrderAlpha (Page 2363)
Propriété TabOrderSwitch (Page 2362)
Propriété Selected (Page 2347)
Propriété Properties (Page 2326)
Propriété PasswordLevel (Page 2305)
Propriété Parent (Page 2303)
Propriété Operation (Page 2298)
Propriété Left (Page 2253)
Propriété LDTooltipTexts (Page 2252)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété GroupParent (Page 2200)
Propriété Events (Page 2159)
Propriété Application (Page 2067)

Objets HMIObjects (liste)

Description



Listage de tous les objets HMIObject représentant les objets d'une vue.

Désignation VBA de l'objet

HMIObjects

Remarque

L'ordre des objets HMI dans la liste HMIObjects peut changer lorsque des objets HMI sont ajoutés et/ou supprimés.

L'ordre de listage peut aussi changer quand les objets HMI sont édités dans le listage actuel. Le comportement peut se produire quand la propriété "Layers" est modifiée et/ou quand les méthodes "SendToBack" et "BringToFront" sont utilisés.

Utilisation

Pour renvoyer le listage HMIObjects, on utilise la propriété HMIObjects. Dans l'exemple qui suit, le nom de tous les objets de la vue active est sorti :

```
Sub ShowObjectsOfDocument()
'VBA270
Dim colObjects As HMIObjects
Dim objObject As HMIObject
Set colObjects = ActiveDocument.HMIObjects
For Each objObject In colObjects
MsgBox objObject.ObjectName
Next objObject
End Sub
```

Pour créer un nouvel objet dans une vue, vous pouvez p. ex. utiliser la méthode "AddHMIObject". Dans l'exemple qui suit, on insère un cercle dans la vue active :

```
Sub AddCircle()
'VBA271
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_1", "HMICircle")
End Sub
```

Vous pouvez p. ex. utiliser la méthode Find pour rechercher un ou plusieurs objets dans la vue. Dans l'exemple qui suit, on recherche dans la vue les objets du type "HMICircle" :

```
Sub FindObjectsByType()
'VBA272
Dim colSearchResults As HMICollection
Dim objMember As HMIObject
Dim iResult As Integer
Dim strName As String
Set colSearchResults = ActiveDocument.HMIObjects.Find(ObjectType:="HMICircle")
For Each objMember In colSearchResults
iResult = colSearchResults.Count
strName = objMember.ObjectName
MsgBox "Found: " & CStr(iResult) & vbCrLf & "Objectname: " & strName
Next objMember
End Sub
```

Voir aussi

Propriété Count (Page 2141)

Objet HMIDefaultObjects (liste) (Page 1939)

Objet Selection (liste) (Page 2010)

Méthode Item (Page 1834)

Méthode Find (Page 1816)

Méthode AddOLEObject (Page 1783)

Méthode AddHMIObject (Page 1780)

Méthode AddActiveXControl (Page 1774)

Vous éditez de la sorte les objets standard, complexes, Windows et tuyaux (Page 1657)

Référence VBA (Page 1724)

Objets standard, complexes, Windows et tuyaux (Page 1655)

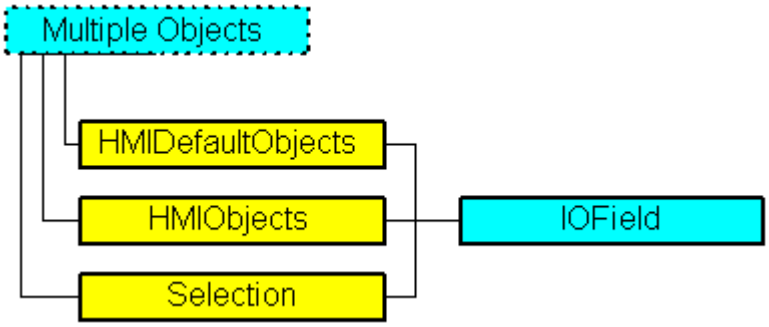
Edition d'objets avec VBA (Page 1653)

Propriété Parent (Page 2303)

Propriété Application (Page 2067)

Objet IOField

Description



Représente l'objet "Champ E/S". L'objet IOField est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIIOField

Utilisation

Pour créer un nouvel objet "Champ E/S" dans une vue, on utilise la méthode Add :

```
Sub AddIOField()
'VBA273
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IO-Field", "HMIIOField")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditIOField()
'VBA274
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects("IO-Field")
objIOField.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

```
Sub ShowNameOfFirstSelectedObject()
'VBA275
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Pour renvoyer un objet du listage HMIDefaultObjects, on utilise "HMIDefaultObjects(Index)" :

```
Sub EditDefaultPropertiesOfIOField()
'VBA276
Dim objIOField As HMIIOField
Set objIOField = Application.DefaultHMIObjects("HMIIOField")
objIOField.BorderColor = RGB(255, 255, 0)
End Sub
```

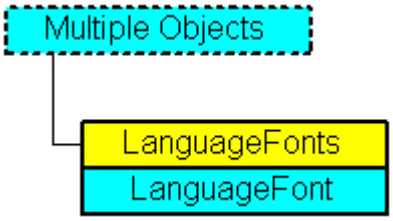
Voir aussi

Propriété LimitMin (Page 2259)
Propriété ClearOnNew (Page 2124)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété PasswordLevel (Page 2305)
Propriété OutputValue (Page 2302)
Propriété OutputFormat (Page 2301)
Propriété Orientation (Page 2301)
Propriété OperationReport (Page 2300)
Propriété OperationMessage (Page 2299)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété LimitMax (Page 2258)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété HiddenInput (Page 2203)
Propriété Height (Page 2201)
Propriété ForeFlashColorOn (Page 2195)
Propriété ForeFlashColorOff (Page 2194)
Propriété ForeColor (Page 2194)
Propriété FontUnderline (Page 2193)
Propriété FontSize (Page 2192)
Propriété FontName (Page 2191)
Propriété FontItalic (Page 2191)
Propriété FontBold (Page 2188)
Propriété FlashRateForeColor (Page 2185)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBackForeColor (Page 2176)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)

L-Q

Objet LanguageFont

Description



Renferme le paramétrage des polices de la langue de travail. L'objet LanguageFont appartient au listage LanguageFont.

Désignation VBA de l'objet

HMILanguageFont

Utilisation

Pour renvoyer un objet LanguageFontindividuel, on utilise LDFonts(Index). Dans l'exemple qui suit, on crée un objet Button et le nom de la première police configurée est sorti :

```
Sub ShowFirstObjectOfCollection()
'VBA277
Dim strName As String
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button", "HMIButton")
strName = objButton.LDFonts(1).Family
MsgBox strName
End Sub
```

Propriétés de l'objet

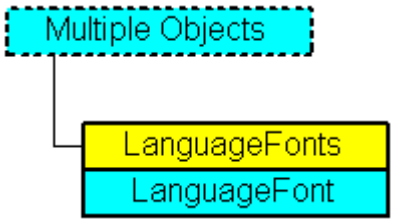
Un objet LanguageFont possède les propriétés suivantes :

Voir aussi

Objet LanguageFonts (liste) (Page 1951)
Référence VBA (Page 1724)
Propriété Underlined (Page 2388)
Propriété Size (Page 2353)
Propriété Parent (Page 2303)
Propriété LanguageID (Page 2219)
Propriété Italic (Page 2212)
Propriété FontFamily (Page 2189)
Propriété Bold (Page 2094)
Propriété Application (Page 2067)

Objet LanguageFonts (liste)

Description



Listage de tous les objets LanguageFont représentant les polices linguistiques d'un objet.

Désignation VBA de l'objet

HMILanguageFonts

Utilisation

Pour renvoyer le listage LanguageFonts, on utilise la propriété LDFonts. Dans l'exemple qui suit, les codes de langue des polices configurées sont sortis :

```
Sub ShowLanguageFont()
'VBA278
Dim colLanguageFonts As HMILanguageFonts
Dim objLanguageFont As HMILanguageFont
Dim objButton As HMIButton
Dim iMax As Integer
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
Set colLanguageFonts = objButton.LDFonts
iMax = colLanguageFonts.Count
```

```
For Each objLanguageFont In colLanguageFonts
MsgBox "Planned fonts: " & iMax & vbCrLf & "Language-ID: " & objLanguageFont.LanguageID
Next objLanguageFont
End Sub
```

On utilise la méthode ItemByLcid pour définir la langue à laquelle le paramétrage de police doit s'appliquer. Dans l'exemple qui suit, on paramètre les attributs de police d'un bouton pour le français et l'anglais.

Remarque

Pour que cet exemple s'applique, les langues concernées doivent avoir été configurées.

```
Sub ExampleForLanguageFonts()
'VBA279
Dim colLangFonts As HMILanguageFonts
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
objButton.Text = "DefText"
Set colLangFonts = objButton.LDFonts

'Adjust fontsettings for french:
With colLangFonts.ItemByLCID(1036)
.Family = "Courier New"
.Bold = True
.Italic = False
.Underlined = True
.Size = 12
End With
'Adjust fontsettings for english:
With colLangFonts.ItemByLCID(1033)
.Family = "Times New Roman"
.Bold = False
.Italic = True
.Underlined = False
.Size = 14
End With
End Sub
```

Voir aussi

Objet LanguageFont (Page 1950)

Méthode ItemByLcid (Page 1836)

Méthode Item (Page 1834)

Référence VBA (Page 1724)

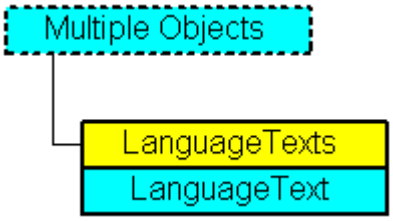
Propriété Parent (Page 2303)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet LanguageText

Description



Renferme les textes multilingues d'un objet. L'objet LanguageText appartient au listage LanguageTexts.

Désignation VBA de l'objet

HMILanguageText

Utilisation

Dans l'exemple qui suit, on affecte un texte en allemand et en anglais au bouton "myButton" :

```
Sub AddLanguagesToButton()
'VBA280
Dim objLabelText As HMILanguageText
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
'
'Add text in actual datalanguage:
objButton.Text = "Actual-Language Text"
'
'Add english text:
Set objLabelText = ActiveDocument.HMIObjects("myButton").LDTexts.Add(1033, "English Text")
End Sub
```

Voir aussi

Objet LanguageTexts (liste) (Page 1954)

Méthode Delete (Page 1807)

Référence VBA (Page 1724)

Propriété Parent (Page 2303)

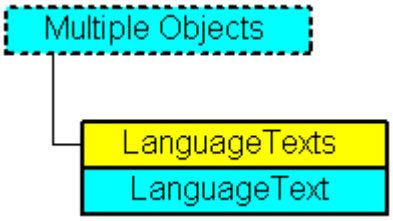
Propriété LanguageID (Page 2219)

Propriété DisplayText (Page 2154)

Propriété Application (Page 2067)

Objet LanguageTexts (liste)

Description



Listage de tous les objets LanguageText représentant les textes multilingues d'un objet.

Désignation VBA de l'objet

HMILanguageTexts

Utilisation

Les propriétés suivantes permettent de renvoyer le listage LanguageTexts :

- Propriété LDLabelTexts
- Propriété LDNames
- Propriété LDStatusTexts
- Propriété LDTexts
- Propriété LDTooltipTexts

Vous trouverez un exemple d'utilisation du listage LanguageTexts dans cette documentation à la rubrique Propriété "LDStatusTexts".

Pour ajouter des textes multilingues à un objet, on utilise la méthode Add. Dans l'exemple qui suit, on affecte un texte en allemand et en anglais au bouton "myButton" :

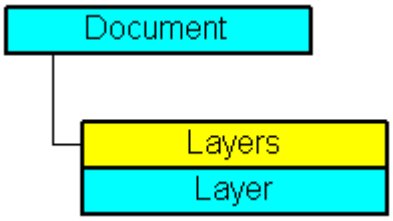
```
Sub AddLanguagesToButton()
'VBA281
Dim objLabelText As HMILanguageText
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
'
'Add text in actual datalanguage:
objButton.Text = "Actual-Language Text"
'
'Add english text:
Set objLabelText = ActiveDocument.HMIObjects("myButton").LDTexts.Add(1033, "English Text")
End Sub
```

Voir aussi

Objet LanguageText (Page 1953)
Méthode ItemByLcid (Page 1836)
Méthode Item (Page 1834)
Référence VBA (Page 1724)
Propriété Parent (Page 2303)
Propriété LDTooltipTexts (Page 2252)
Propriété LDTexts (Page 2251)
Propriété LDStatusTexts (Page 2250)
Propriété LDNames (Page 2249)
Propriété LDLabelTexts (Page 2248)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

Objet Layer

Description



Représente un des 32 niveaux disponibles dans une vue.

Désignation VBA de l'objet

HMILayer

Utilisation

On utilise l'objet Layer pour attribuer un nom à un niveau et pour défini ses facteurs de zoom minimum et maximum. La visibilité du niveau est configurée à part selon qu'il s'agit d'un niveau CS ou d'un niveau RT :

- Document-Objekt : commande la visibilité des niveaux RT.
- Objet View : commande la visibilité des niveaux CS.

On utilise le listage Layers pour renvoyer un objet Layer. Dans l'exemple qui suit, on définit les paramètres du niveau le plus bas de la vue active :

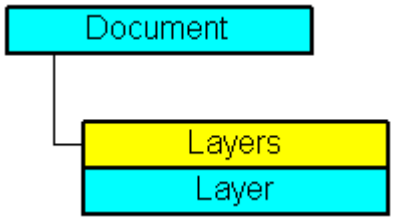
```
Sub ConfigureSettingsOfLayer
'VBA282
Dim objLayer As HMILayer
Set objLayer = ActiveDocument.Layers(1)
With objLayer
'configure "Layer 0"
.MinZoom = 10
.MaxZoom = 100
.Name = "Configured with VBA"
End With
End Sub
```

Voir aussi

Propriété Layers (Page 2246)
Référence VBA (Page 1724)
Edition de niveaux avec VBA (Page 1650)
Propriété Visible (Page 2469)
Propriété Number (Page 2291)
Propriété Name (Page 2289)
Propriété MinZoom (Page 2287)
Propriété MaxZoom (Page 2270)
Propriété LDNames (Page 2249)
Propriété ActiveLayer (Page 2056)

Objet Layers (Liste)

Description



Listage des objets Layer qui représentent les 32 niveaux d'un vue.

Désignation VBA de l'objet

HMILayer

Utilisation

Pour renvoyer le listage Layers, on utilise la propriété LayersCS ou la propriété LayersRT. Dans l'exemple qui suit, le nom des niveaux d'une copie de la vue active est sorti :

```
Sub ShowLayer()
'VBA283
Dim colLayers As HMILayers
Dim objLayer As HMILayer
Dim strLayerList As String
Dim iCounter As Integer
iCounter = 1
Set colLayers = ActiveDocument.Layers
For Each objLayer In colLayers
If 1 = iCounter Mod 2 And 32 > iCounter Then
strLayerList = strLayerList & vbCrLf
ElseIf 11 > iCounter Then
strLayerList = strLayerList & "       "
Else
strLayerList = strLayerList & "       "
End If
strLayerList = strLayerList & objLayer.Name
iCounter = iCounter + 1
Next objLayer
MsgBox strLayerList
End Sub
```

Voir aussi

Objet Layer (Page 1955)

Méthode Item (Page 1834)

Référence VBA (Page 1724)

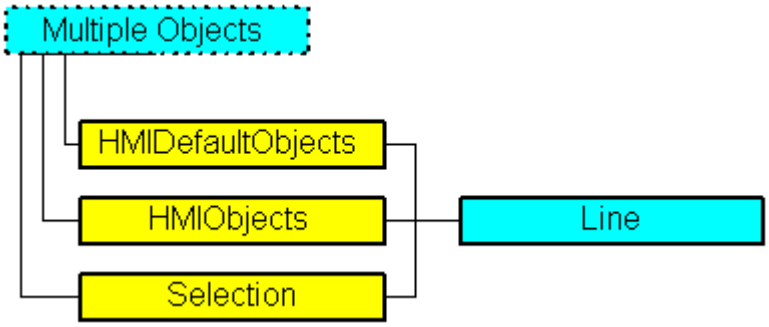
Propriété Parent (Page 2303)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet Line

Description



Représente l'objet "Ligne". L'objet Line est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMILine

Utilisation

Pour créer un nouvel objet "Ligne" dans une vue, on utilise la méthode Add :

```
Sub AddLine()
'VBA285
Dim objLine As HMILine
Set objLine = ActiveDocument.HMIObjects.AddHMIObject("Line1", "HMILine")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditLine()
'VBA286
Dim objLine As HMILine
Set objLine = ActiveDocument.HMIObjects("Line1")
objLine.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

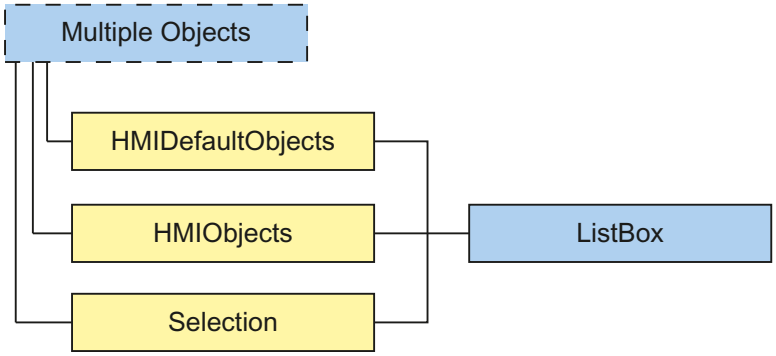
```
Sub ShowNameOfFirstSelectedObject()
'VBA287
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Méthode AddHMIObject (Page 1780)
Propriété BorderBackColor (Page 2095)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété RotationAngle (Page 2336)
Propriété ReferenceRotationTop (Page 2333)
Propriété ReferenceRotationLeft (Page 2332)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Index (Page 2207)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashBorderColor (Page 2174)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderEndStyle (Page 2099)
Propriété BorderColor (Page 2096)
Propriété ActualPointTop (Page 2057)
Propriété ActualPointLeft (Page 2056)

Objet ListBox

Description



Représente l'objet "Champ de liste". L'objet ListBox est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMIListBox

Utilisation

Pour créer un nouvel objet "Champ de liste" dans la vue, on utilise la méthode Add :

```
Sub AddListBox()
'VBA829
Dim objListBox As HMIListBox
Set objListBox = ActiveDocument.HMIObjects.AddHMIObject("Champ de liste", "HMIListBox")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditListBox()
'VBA830
Dim objListBox As HMIListBox
Set objListBox = ActiveDocument.HMIObjects("Champ de liste")
objListBox.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

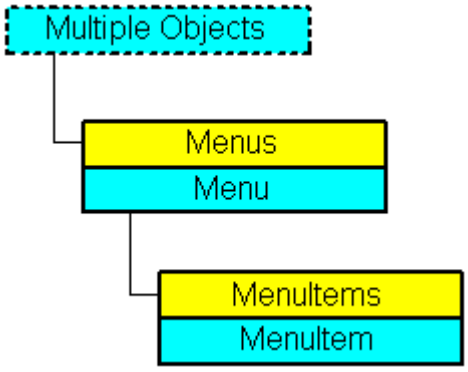
```
Sub ShowNameOfFirstSelectedObject()
'VBA831
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ObjectName (Page 2293)
Propriété Layer (Page 2222)
Propriété Left (Page 2253)
Propriété Top (Page 2374)
Propriété Width (Page 2472)
Propriété Height (Page 2201)
Propriété NumberLines (Page 2292)
Propriété ForeColor (Page 2194)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackColor (Page 2076)
Propriété FillColor (Page 2164)
Propriété BorderStyle (Page 2103)
Propriété BorderWidth (Page 2105)
Propriété FillStyle (Page 2169)
Propriété GlobalShadow (Page 2197)
Propriété FontName (Page 2191)
Propriété FontSize (Page 2192)
Propriété FontBold (Page 2188)
Propriété FontItalic (Page 2191)
Propriété FontUnderline (Page 2193)
Propriété AlignmentLeft (Page 2063)
Propriété Index (Page 2207)
Propriété Text (Page 2369)
Propriété Operation (Page 2298)
Propriété PasswordLevel (Page 2305)
Propriété Visible (Page 2469)
Propriété TooltipText (Page 2373)
Propriété OperationMessage (Page 2299)
Propriété OperationReport (Page 2300)
Propriété SelIndex (Page 2349)
Propriété SelText (Page 2349)

Objet Menu

Description



Représente l'objet "Menu personnalisé". L'objet Menu appartient au listage CustomMenus.

Désignation VBA de l'objet

HMIMenu

Utilisation

On utilise CustomMenus pour renvoyer un objet Menu individuel. Comme "Index", on peut utiliser le numéro d'index ou le nom de l'objet. Pour que l'exemple qui suit s'applique, vous devez créer un menu personnalisé. Vous trouverez un exemple de cette création dans cette documentation à la rubrique "Création d'un nouveau menu personnalisé". Dans l'exemple qui suit, la désignation du premier menu personnalisé de la vue active est sorti :

```
Sub ShowFirstMenuOfMenucollection()
'VBA288
Dim strName As String
strName = ActiveDocument.CustomMenus(1).Label
MsgBox strName
End Sub
```

Pour supprimer un objet "Menu" du listage "CustomMenus", on utilise la méthode Delete. Dans l'exemple qui suit, on supprime le premier menu personnalisé de la vue active :

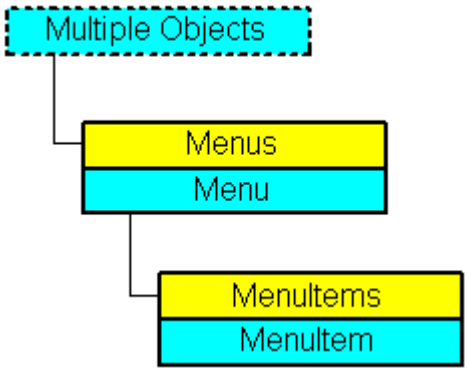
```
Sub DeleteMenu()
'VBA289
Dim objMenu As HMIMenu
Set objMenu = ActiveDocument.CustomMenus(1)
objMenu.Delete
End Sub
```

Voir aussi

Objet Menus (liste) (Page 1965)
Méthode Delete (Page 1807)
Pour créer des menus et des barres d'outils spécifiques à une vue (Page 1648)
Pour créer un nouveau menu spécifique à une application (Page 1622)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)
Propriété Visible (Page 2469)
Propriété StatusText (Page 2359)
Propriété Position (Page 2320)
Propriété Parent (Page 2303)
Propriété MenuItems (Page 2284)
Propriété LDStatusTexts (Page 2250)
Propriété LDLabelTexts (Page 2248)
Propriété Label (Page 2218)
Propriété Key (Page 2217)
Propriété Enabled (Page 2158)
Propriété Application (Page 2067)

Objet Menus (liste)

Description



Listage de tous les objets Menu représentant les menus personnalisés de Graphics Designer.

Désignation VBA de l'objet

HMIMenus

Utilisation

Pour renvoyer le listage Menus, on utilise la propriété CustomMenus. Dans l'exemple qui suit, tous les menus personnalisés de la vue active sont sortis.

Remarque

Le listage Menus restauré ne distingue pas entre les menus spécifiques d'une application et ceux spécifiques d'une vue.

```
Sub ShowCustomMenusOfDocument()
'VBA290
Dim colMenus As HMIMenus
Dim objMenu As HMIMenu
Dim strMenuList As String
Set colMenus = ActiveDocument.CustomMenus
For Each objMenu In colMenus
strMenuList = strMenuList & objMenu.Label & vbCrLf
Next objMenu
MsgBox strMenuList
End Sub
```

Pour créer un menu spécifique d'une application, on utilise la propriété Application et la méthode InsertMenu. Le code VBA est créé soit dans le document "Project Template Document", soit dans "Global Template Document". Dans l'exemple qui suit, on crée le menu spécifique d'une application "myApplicationMenu" :

```
Sub InsertApplicationSpecificMenu()
'VBA291
Dim objMenu As HMIMenu
Set objMenu = Application.CustomMenus.InsertMenu(1, "a_Menu1", "myApplicationMenu")
End Sub
```

Pour créer un menu spécifique d'une vue, on utilise la propriété ActiveDocument et la méthode InsertMenu. Le code VBA est à créer dans le document "ThisDocument". Dans l'exemple qui suit, on crée le menu spécifique d'une vue "myDocumentMenu" :

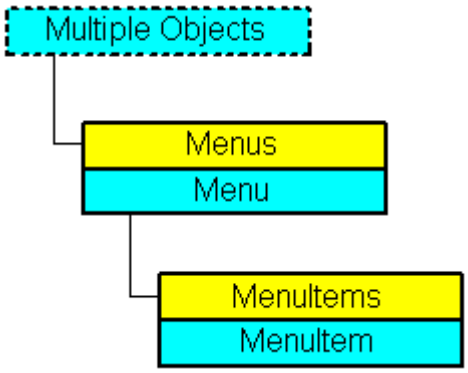
```
Sub InsertDocumentSpecificMenu()
'VBA292
Dim objMenu As HMIMenu
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "d_Menu1", "myDocumentMenu")
End Sub
```

Voir aussi

Objet Menu (Page 1964)
Méthode Item (Page 1834)
Méthode InsertMenu (Page 1825)
Pour créer des menus et des barres d'outils spécifiques à une vue (Page 1648)
Pour créer un nouveau menu spécifique à une application (Page 1622)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

Objet MenuItem

Description



Représente une entrée d'un menu personnalisé de GraphicsDesigner. L'objet MenuItem appartient au listage MenuItems.

Désignation VBA de l'objet

HMIMenuItem

Utilisation

Remarque

Pour que les exemples s'appliquent, vous devez d'abord créer un menu personnalisé. Vous trouverez un exemple de cette création, p. ex. à la rubrique "Insertion d'une commande dans le menu".

On utilise MenuItems(Index) pour renvoyer un objet MenuItem individuel. Comme "Index", on peut utiliser le numéro d'index ou le nom de l'objet. Dans l'exemple qui suit, la première entrée du premier menu personnalisé de la vue active est sortie :

```
Sub ShowFirstObjectOfCollection()
'VBA293
Dim strName As String
strName = ActiveDocument.CustomMenus(1).MenuItems(1).Label
MsgBox strName
End Sub
```

Pour supprimer un objet du listage "MenuItems", on utilise la méthode Delete. Dans l'exemple qui suit, on supprime la première entrée du premier menu personnalisé de la vue active :

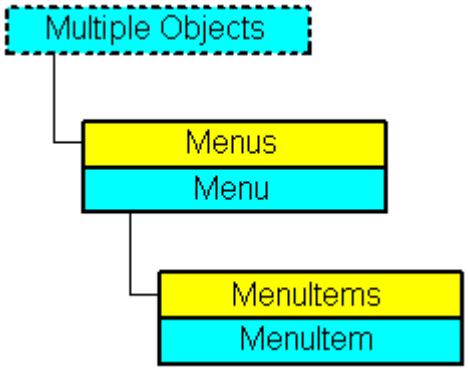
```
Sub DeleteMenuItem()
'VBA294
ActiveDocument.CustomMenus(1).MenuItems(1).Delete
End Sub
```

Voir aussi

Propriété Parent (Page 2303)
Objet MenuItems (liste) (Page 1970)
Méthode Delete (Page 1807)
Configuration des menus et des barres d'outils (Page 1619)
Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)
Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)
Pour insérer une nouvelle option de menu (Page 1623)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)
Propriété Visible (Page 2469)
Propriété Tag (Page 2366)
Propriété SubMenu (Page 2360)
Propriété StatusText (Page 2359)
Propriété ShortCut (Page 2351)
Propriété Position (Page 2320)
Propriété MenuItemType (Page 2285)
Propriété Macro (Page 2267)
Propriété LDStatusTexts (Page 2250)
Propriété LDLabelTexts (Page 2248)
Propriété Label (Page 2218)
Propriété Key (Page 2217)
Propriété Icon (Page 2206)
Propriété Enabled (Page 2158)
Propriété Checked (Page 2116)
Propriété Application (Page 2067)

Objet MenuItems (liste)

Description



Listage de tous les objets MenuItem représentant les entrées d'un menu personnalisé.

Utilisation

Remarque

Pour que les exemples s'appliquent, vous devez d'abord créer un menu personnalisé. Vous trouverez un exemple de cette création, p. ex. à la rubrique "Insertion d'une commande dans le menu".

Pour renvoyer le listage MenuItems, on utilise la propriété MenuItems. Dans l'exemple qui suit, toutes les entrées du premier menu personnalisé de la vue active sont sorties :

Remarque

Le listage Menus restauré ne distingue pas entre les menus spécifiques d'une application et ceux spécifiques d'une vue.

```
Sub ShowMenuItems()
'VBA295
Dim colMenuItems As HMIMenuItems
Dim objMenuItem As HMIMenuItem
Dim strItemList As String
Set colMenuItems = ActiveDocument.CustomMenus(1).MenuItems
For Each objMenuItem In colMenuItems
strItemList = strItemList & objMenuItem.Label & vbCrLf
Next objMenuItem
MsgBox strItemList
End Sub
```

Utilisez p. ex. la méthode InsertMenuItem pour insérer une commande de menu dans un menu personnalisé existant. Dans l'exemple qui suit, on crée le menu spécifique d'une vue "DocMenu2" dans la vue active et on y insère l'entrée de menu "MenuItem 1" :

```
Sub InsertMenuItem()
'VBA296
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(2, "d_Menu2", "DocMenu2")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "m_Item2_1", "MenuItem 1")
End Sub
```

Voir aussi

Méthode InsertSubmenu (Page 1829)

Objet MenuItem (Page 1967)

Méthode InsertSeparator (Page 1828)

Méthode InsertMenuItem (Page 1826)

Pour insérer une nouvelle option de menu (Page 1623)

Référence VBA (Page 1724)

Création de menus et de barres d'outils personnalisés (Page 1620)

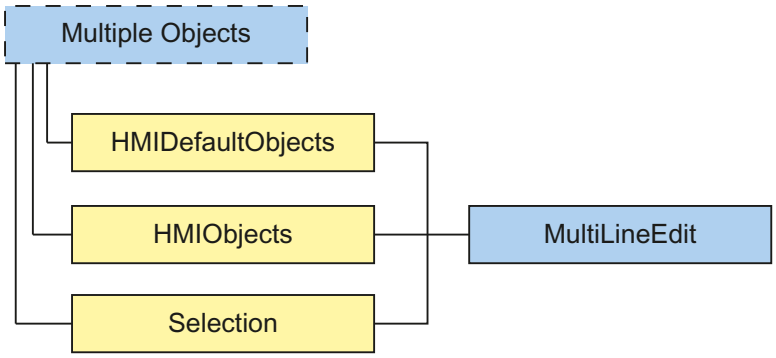
Propriété Parent (Page 2303)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet MultiLineEdit

Description



Représente l'objet "Texte multiligne". L'objet MultiLineEdit est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMIMultiLineEdit

Utilisation

Pour créer un nouvel objet "Texte multiligne" dans la vue, on utilise la méthode Add :

```
Sub AddMultiLineEdit()
'VBA832
Dim objMultiLineEdit As HMIMultiLineEdit
Set objMultiLineEdit = ActiveDocument.HMIObjects.AddHMIObject("Texte multiligne",
"HMIMultiLineEdit")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditMultiLineEdit()
'VBA833
Dim objMultiLineEdit As HMIMultiLineEdit
Set objMultiLineEdit = ActiveDocument.HMIObjects("Texte multiligne")
objMultiLineEdit.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

```
Sub ShowNameOfFirstSelectedObject()
'VBA834
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Layer (Page 2222)
Propriété Left (Page 2253)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BorderStyle (Page 2103)
Propriété BorderWidth (Page 2105)
Propriété BackColor (Page 2076)
Propriété FontName (Page 2191)
Propriété FontSize (Page 2192)
Propriété FontBold (Page 2188)
Propriété FontItalic (Page 2191)
Propriété FontUnderline (Page 2193)
Propriété ForeColor (Page 2194)
Propriété AlignmentLeft (Page 2063)
Propriété Top (Page 2374)
Propriété Width (Page 2472)
Propriété Height (Page 2201)
Propriété Text (Page 2369)
Propriété Operation (Page 2298)
Propriété PasswordLevel (Page 2305)
Propriété Visible (Page 2469)
Propriété TooltipText (Page 2373)
Propriété ObjectName (Page 2293)
Propriété GlobalShadow (Page 2197)

Objet objConnection

Description

Représente l'objet "Connecteur". L'objet objConnection est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.

Remarque

Les propriétés d'un objet objConnection ne sont accessibles qu'en lecture seule.

Désignation VBA de l'objet

HMIobjConnection

Utilisation

Les propriétés d'un objet objConnection permettent de savoir quels sont les objets liés.

Exemple

Pour que l'exemple suivant s'applique, vous devez avoir lié deux objets de la vue qui est active dans Graphics Designer à l'aide du connecteur. L'objet Connecteur se trouve dans la palette d'objets de Graphics Designer, sous "Objets simples". Pour que l'exemple s'applique, nommez le connecteur "Connector1".

L'entrée "Info connecteurs" du menu "Connector Info" vous permet d'afficher les objets liés :

```
Sub ShowConnectorInfo_Menu()
'VBA297
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim strDocName As String
strDocName = Application.ApplicationDataPath & ActiveDocument.Name
Set objMenu = Documents(strDocName).CustomMenus.InsertMenu(1, "ConnectorMenu",
"Connector_Info")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "ShowConnectInfo", "Info Connector")
End Sub

Sub ShowConnectorInfo()
Dim objConnector As HMIObjConnection
Dim iStart As Integer
Dim iEnd As Integer
Dim strStart As String
Dim strEnd As String
Dim strObjStart As String
Dim strObjEnd As String
Set objConnector = ActiveDocument.HMIObjects("Connector1")
iStart = objConnector.BottomConnectedConnectionPointIndex
iEnd = objConnector.TopConnectedConnectionPointIndex
strObjStart = objConnector.BottomConnectedObjectName
strObjEnd = objConnector.TopConnectedObjectName
Select Case iStart
Case 0
strStart = "top"
Case 1
strStart = "right"
Case 2
strStart = "bottom"
Case 3
strStart = "left"
End Select
Select Case iEnd
Case 0
strEnd = "top"
```

```
Case 1
strEnd = "right"
Case 2
strEnd = "bottom"
Case 3
strEnd = "left"
End Select
MsgBox "The selected connector links the objects " & vbCrLf & "'" & strObjStart & "' and '"
& strObjEnd & "'" & vbCrLf & "Connected points: " & vbCrLf & strObjStart & ": " & strStart
& vbCrLf & strObjEnd & ": " & strEnd
End Sub

Private Sub Document_MenuItemClicked(ByVal MenuItem As IHMIMenuItem)
Select Case MenuItem.Key
Case "ShowConnectInfo"
Call ShowConnectorInfo
End Select
End Sub
```

Voir aussi

Propriété TopConnectedConnectionPointIndex (Page 2375)

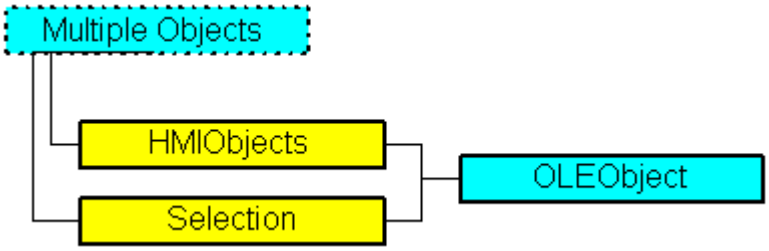
Propriété TopConnectedObjectName (Page 2374)

Propriété BottomConnectedConnectionPointIndex (Page 2107)

Propriété BottomConnectedObjectName (Page 2106)

Objet OLEObject

Description



Représente l'objet "Objet OLE". L'objet OLEObject est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.

Désignation VBA de l'objet

HMIOLEObject

Utilisation

Pour créer un nouvel objet "Objet OLE" dans une vue, on utilise la méthode Add : Dans l'exemple qui suit, on insère un objet OLE dans la vue active qui renferme un document WordPad :

```
Sub AddOLEObjectToActiveDocument()
'VBA298
Dim objOleObject As HMIOLEObject
Set objOleObject = ActiveDocument.HMIObjects.AddOLEObject("Wordpad Document",
"Wordpad.Document.1")
End Sub
```

Pour renvoyer un objet du listage HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet : Dans cet exemple, on attribue la valeur 140 à la coordonnée x de l'objet OLE "Wordpad Document" :

```
Sub EditOLEObject()
'VBA299
Dim objOleObject As HMIOLEObject
Set objOleObject = ActiveDocument.HMIObjects("Wordpad Document")
objOleObject.Left = 140
End Sub
```

Pour renvoyer un objet du listage Selection, on utilise la méthode "Selection(Index)" : Comme "Index", on peut utiliser le numéro d'index ou le nom de l'objet. Dans cet exemple, on affiche le nom du premier objet sélectionné :

```
Sub ShowNameOfFirstSelectedObject()
'VBA300
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Pour créer des menus et des barres d'outils spécifiques à une vue (Page 1648)

Objet Selection (liste) (Page 2010)

Objets HMIObjects (liste) (Page 1945)

Méthode Delete (Page 1807)

Méthode AddOLEObject (Page 1783)

Pour créer une nouvelle barre d'outils spécifique à une application (Page 1629)

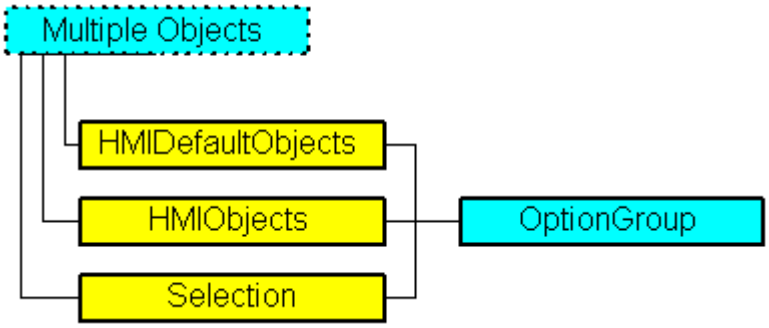
Référence VBA (Page 1724)

Objets OLE (Page 1661)

Propriété Application (Page 2067)

Objet OptionGroup

Description



Représente l'objet "Radio-Box" (case d'option). L'objet OptionGroup est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIOptionGroup

Utilisation

Pour créer un nouvel objet "Radio-Box" dans une vue, on utilise la méthode Add :

```
Sub AddOptionGroup()
```

```
'VBA301
Dim objOptionGroup As HMIOptionGroup
Set objOptionGroup = ActiveDocument.HMIObjects.AddHMIObject("Radio-Box", "HMIOptionGroup")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditOptionGroup()
'VBA302
Dim objOptionGroup As HMIOptionGroup
Set objOptionGroup = ActiveDocument.HMIObjects("Radio-Box")
objOptionGroup.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

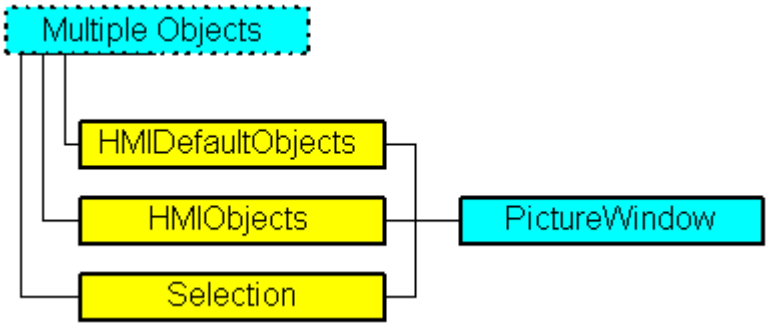
```
Sub ShowNameOfFirstSelectedObject()
'VBA303
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Left (Page 2253)
Propriété BorderStyle (Page 2103)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété Text (Page 2369)
Propriété Process (Page 2323)
Propriété PasswordLevel (Page 2305)
Propriété Orientation (Page 2301)
Propriété OperationMessage (Page 2299)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Layer (Page 2222)
Propriété Index (Page 2207)
Propriété Height (Page 2201)
Propriété ForeFlashColorOn (Page 2195)
Propriété ForeFlashColorOff (Page 2194)
Propriété ForeColor (Page 2194)
Propriété FontUnderline (Page 2193)
Propriété FontSize (Page 2192)
Propriété FontName (Page 2191)
Propriété FontItalic (Page 2191)
Propriété FontBold (Page 2188)
Propriété FlashRateForeColor (Page 2185)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashForeColor (Page 2176)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)

Objet PictureWindow

Description



Représente l'objet "Fenêtre de vue". L'objet "PictureWindow" est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIPictureWindow

Utilisation

Pour créer un nouvel objet "Fenêtre de vue" dans une vue, on utilise la méthode Add :

```
Sub AddPictureWindow()
'VBA304
Dim objPictureWindow As HMIPictureWindow
Set objPictureWindow = ActiveDocument.HMIObjects.AddHMIObject("PictureWindow1",
"HMIPictureWindow")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditPictureWindow()
'VBA305
Dim objPictureWindow As HMIPictureWindow
Set objPictureWindow = ActiveDocument.HMIObjects("PictureWindow1")
objPictureWindow.Sizeable = True
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

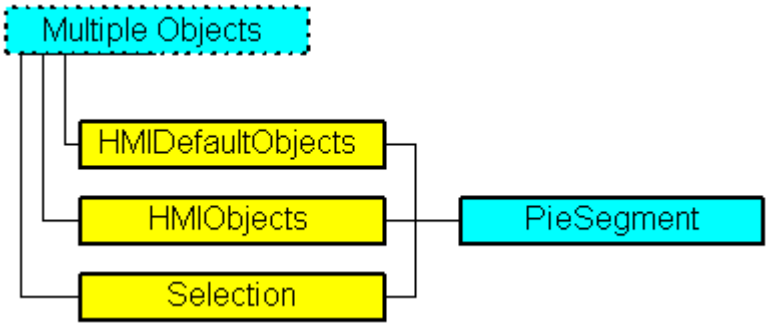
```
Sub ShowNameOfFirstSelectedObject()
'VBA306
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété MaximizeButton (Page 2270)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Zoom (Page 2478)
Propriété WindowBorder (Page 2473)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété UpdateCycle (Page 2390)
Propriété Top (Page 2374)
Propriété TagPrefix (Page 2367)
Propriété Sizeable (Page 2354)
Propriété ServerPrefix (Page 2350)
Propriété ScrollPositionY (Page 2345)
Propriété ScrollPositionX (Page 2344)
Propriété ScrollBars (Page 2343)
Propriété PictureName (Page 2315)
Propriété OnTop (Page 2297)
Propriété OffsetTop (Page 2296)
Propriété OffsetLeft (Page 2296)
Propriété Name (Page 2289)
Propriété Moveable (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété CloseButton (Page 2125)
Propriété CaptionText (Page 2114)
Propriété Caption (Page 2113)
Propriété AdaptSize (Page 2060)
Propriété AdaptPicture (Page 2059)

Objet PieSegment

Description



Représente l'objet Segment de cercle. L'objet "PieSegment" est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIPieSegment

Utilisation

Pour créer un nouvel objet "Segment de cercle" dans une vue, on utilise la méthode Add :

```
Sub AddPieSegment()
'VBA307
Dim objPieSegment As HMIPieSegment
Set objPieSegment = ActiveDocument.HMIObjects.AddHMIObject("PieSegment1", "HMIPieSegment")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditPieSegment()
'VBA308
Dim objPieSegment As HMIPieSegment
Set objPieSegment = ActiveDocument.HMIObjects("PieSegment1")
objPieSegment.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

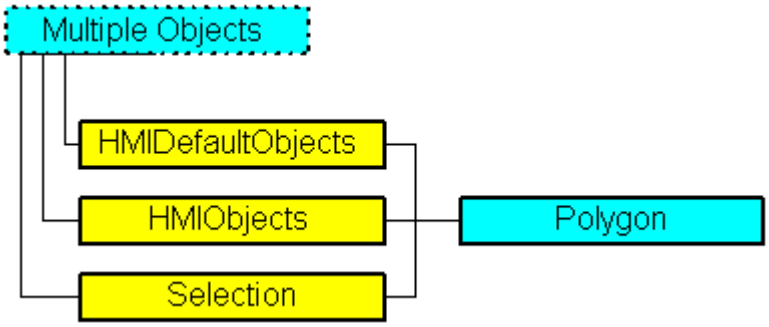
```
Sub ShowNameOfFirstSelectedObject()
'VBA309
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Filling (Page 2166)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété StartAngle (Page 2359)
Propriété Radius (Page 2330)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété FillColor (Page 2164)
Propriété EndAngle (Page 2158)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackFlashColorOn (Page 2082)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)

Objet Polygon

Description



Représente l'objet "Polygone". L'objet Polygone est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIPolygon

Utilisation

Pour créer un nouvel objet "Polygone" dans une vue, on utilise la méthode Add :

```
Sub AddPolygon()
'VBA310
Dim objPolygon As HMIPolygon
Set objPolygon = ActiveDocument.HMIObjects.AddHMIObject("Polygon", "HMIPolygon")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditPolygon()
'VBA311
Dim objPolygon As HMIPolygon
Set objPolygon = ActiveDocument.HMIObjects("Polygon")
objPolygon.BorderColor = RGB (255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

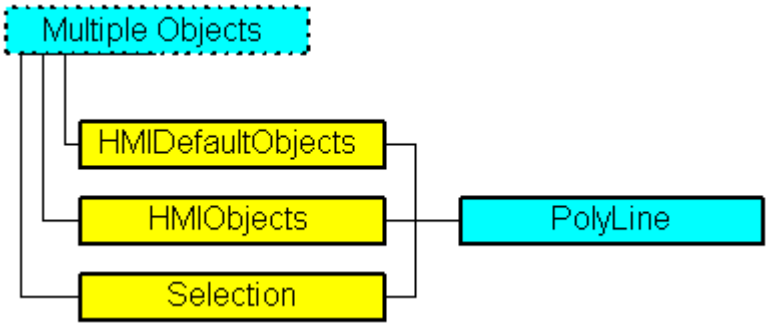
```
Sub ShowNameOfFirstSelectedObject()
'VBA312
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété TooltipText (Page 2373)
Propriété BorderBackColor (Page 2095)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété RotationAngle (Page 2336)
Propriété ReferenceRotationTop (Page 2333)
Propriété ReferenceRotationLeft (Page 2332)
Propriété PointCount (Page 2319)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Index (Page 2207)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BackFlashColorOn (Page 2082)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)
Propriété ActualPointTop (Page 2057)
Propriété ActualPointLeft (Page 2056)

Objet PolyLine

Description



Représente l'objet "Trait polygonal". L'objet "PolyLine" est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIPolyLine

Utilisation

Pour créer un nouvel objet "Trait polygonal" dans une vue, on utilise la méthode Add :

```
Sub AddPolyLine()
'VBA313
Dim objPolyLine As HMIPolyLine
Set objPolyLine = ActiveDocument.HMIObjects.AddHMIObject("PolyLine1", "HMIPolyLine")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditPolyLine()
'VBA314
Dim objPolyLine As HMIPolyLine
Set objPolyLine = ActiveDocument.HMIObjects("PolyLine1")
objPolyLine.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

```
Sub ShowNameOfFirstSelectedObject()
'VBA315
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Pour renvoyer un objet du listage HMIDefaultObjects, on utilise "HMIDefaultObjects(Index)" :

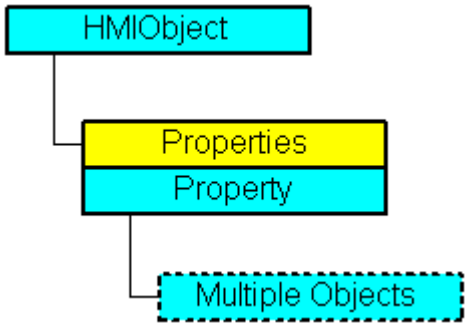
```
Sub EditDefaultPropertiesOfPolyLine()
'VBA316
Dim objPolyLine As HMIPolyLine
Set objPolyLine = Application.DefaultHMIObjects("HMIPolyLine")
objPolyLine.BorderColor = RGB(255, 255, 0)
End Sub
```

Voir aussi

Objet HMIDefaultObjects (liste) (Page 1939)
Propriété BorderEndStyle (Page 2099)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété RotationAngle (Page 2336)
Propriété ReferenceRotationTop (Page 2333)
Propriété ReferenceRotationLeft (Page 2332)
Propriété PointCount (Page 2319)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Index (Page 2207)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashBorderColor (Page 2174)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété ActualPointTop (Page 2057)
Propriété ActualPointLeft (Page 2056)

Objet Properties (liste)

Description



Listage de tous les objets Property représentant les propriétés d'un objet.

Désignation VBA de l'objet

HMIProperties

Utilisation

Lorsqu'il est impossible d'accéder directement à une propriété d'objet, on utilise la propriété Properties(Index) pour restaurer un objet Property. Comme "Index", on peut utiliser le numéro d'index ou la désignation VBA de l'objet. Dans l'exemple qui suit, on est obligé d'utiliser la propriété Properties pour pouvoir accéder aux diverses propriétés d'un cercle. Le cercle est inséré dans la vue en tant qu'objet HMIObject :

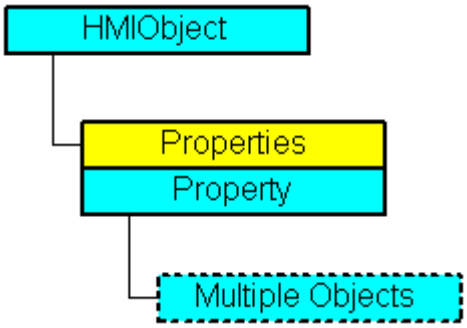
```
Sub AddObject()
'VBA319
Dim objObject As HMIObject
Set objObject = ActiveDocument.HMIObjects.AddHMIObject("CircleAsHMIObject", "HMICircle")
'
'Standard properties (e.g. "Position") are available every time:
objObject.Top = 40
objObject.Left = 40
'
'Individual properties have to be called using
'property "Properties":
objObject.Properties("FlashBackColor") = True
End Sub
```

Voir aussi

Méthode Item (Page 1834)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

Objet Property

Description



Représente la propriété d'un objet Dans un objet Property, la propriété Value est utilisée par défaut. De ce fait, si vous souhaitez p. ex. affecter une nouvelle valeur à une propriété d'objet, vous pouvez utiliser la formulation suivante :

```
<Object>.<Property> = <Value>
```

La propriété "Dynamic" permet de dynamiser une propriété d'objet avec VBA. Pour configurer des actions avec VBA, on utilise le listage "Events".

L'objet Property appartient au listage Properties.

Désignation VBA de l'objet

HMIProperty

Utilisation

On utilise Properties(Index) pour restaurer un objet Property individuel. Comme "Index", on peut utiliser le numéro d'index ou le nom de la propriété d'objet. Dans l'exemple qui suit, on obtient le nom de la première propriété de l'objet Cercle :

```
Sub ShowFirstObjectOfCollection()
'VBA317
```

```
Dim objCircle As HMICircle
Dim strName As String
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle", "HMICircle")
strName = objCircle.Properties(1).Name
MsgBox strName
End Sub
```

Pour dynamiser une propriété d'objet, on peut p. ex. utiliser la méthode CreateDynamic. Dans l'exemple qui suit, on dynamise la propriété d'objet "Rayon" d'un cercle par la variable "Otto" actualisée toutes les deux secondes :

```
Sub DynamicToRadiusOfNewCircle()
'VBA318
Dim objVariableTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects("Circle")
Set objVariableTrigger =
objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVariableDirect, "NewDynamic1")
objVariableTrigger.CycleType = hmiCycleType_2s
End Sub
```

Voir aussi

Propriété DisplayName (Page 2152)

Objet Properties (liste) (Page 1992)

Méthode DeleteDynamic (Page 1809)

Méthode CreateDynamic (Page 1803)

Référence VBA (Page 1724)

Création de dynamisations avec VBA (Page 1680)

Edition d'objets avec VBA (Page 1653)

Propriété Value (Page 2395)

Propriété Type (Page 2378)

Propriété Parent (Page 2303)

Propriété Name (Page 2289)

Propriété IsDynamicable (Page 2211)

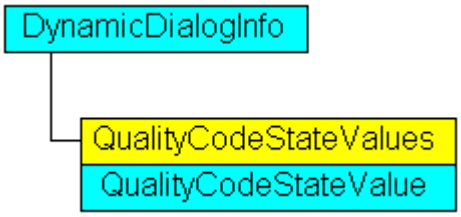
Propriété Events (Page 2159)

Propriété Dynamic (Page 2155)

Propriété Application (Page 2067)

Objets QualityCodeStateValue

Description



Représente le code qualité d'une variable à laquelle est affectée dans le dialogue dynamique une valeur utilisée pour la dynamisation.

Désignation VBA de l'objet

HMIQualityCodeStateValue

Propriétés de l'objet

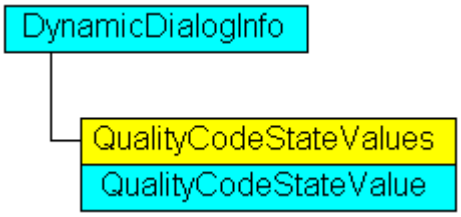
Un objet QualityCodeStateValue possède les propriétés suivantes :

Voir aussi

Propriété VALUE_BAD_CONFERROR (Page 2402)
Référence VBA (Page 1724)
Propriété VarName (Page 2467)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Propriété Parent (Page 2303)
Propriété Application (Page 2067)

Objet QualityCodeStateValues (listage)

Description



Un listage d'objets QualityCodeStateValue contenant tous les codes qualité dans le dialogue dynamique utilisés pour la dynamisation.

Désignation VBA de l'objet

HMIQualityCodeStateValues

Utilisation

Utilisez par exemple la propriété Item pour définir dans le dialogue dynamique les valeurs utilisées pour la dynamisation si la variable indiquée renvoie le code qualité configuré. Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA813
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'Activate qualitycode-statecheck
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
```

```
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Propriétés de l'objet

Un objet QualityCodeStateValues possède les propriétés suivantes :

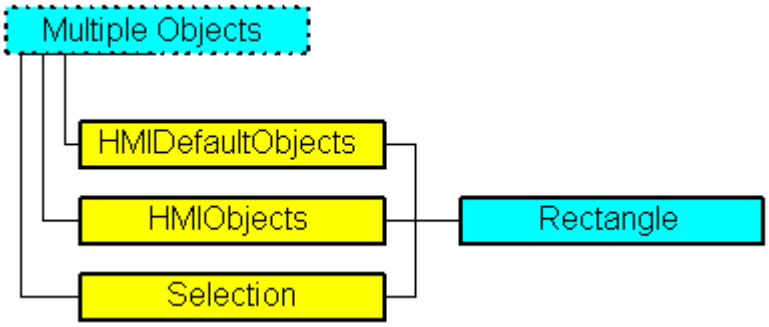
Voir aussi

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Référence VBA (Page 1724)
Propriété VarName (Page 2467)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Propriété Parent (Page 2303)
Propriété Item (Page 2213)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

R-Z

Objet Rectangle

Description



Représente l'objet "Rectangle". L'objet Rectangle est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIRectangle

Utilisation

Pour créer un nouvel objet "Rectangle" dans une vue, on utilise la méthode Add :

```
Sub AddRectangle()
'VBA320
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditRectangle()
'VBA321
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects("Rectangle1")
objRectangle.BorderColor = RGB(255, 0, 0)
```

```
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

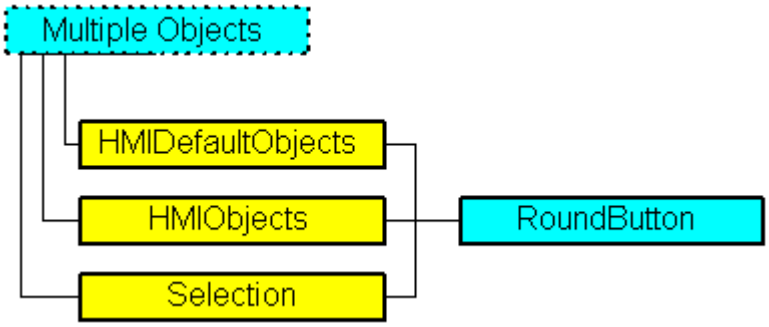
```
Sub ShowNameOfFirstSelectedObject()
'VBA322
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Name (Page 2289)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BackFlashColorOn (Page 2082)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)

Objet RoundButton

Description



Représente l'objet "Bouton rond". L'objet RoundButton est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIRoundButton

Utilisation

Pour créer un nouvel objet "Bouton rond" dans une vue, on utilise la méthode Add :

```
Sub AddRoundButton()
'VBA323
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("Roundbutton1",
"HMIRoundButton")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditRoundButton()
'VBA324
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects("Roundbutton1")
objRoundButton.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

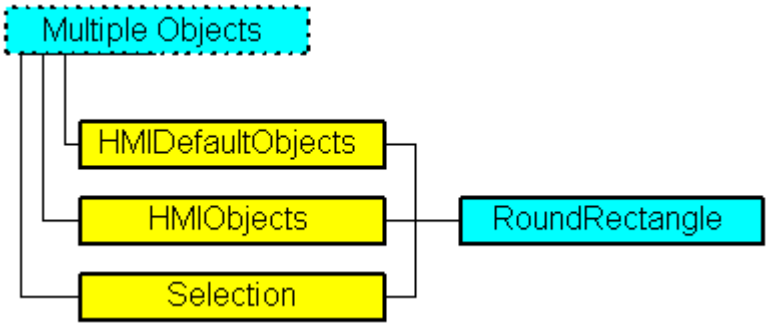
```
Sub ShowNameOfFirstSelectedObject()
'VBA325
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété TooltipText (Page 2373)
Propriété FlashBackColor (Page 2173)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété Toggle (Page 2370)
Propriété Radius (Page 2330)
Propriété Pressed (Page 2322)
Propriété PicUpUseTransColor (Page 2317)
Propriété PicUpTransparent (Page 2317)
Propriété PicUpReferenced (Page 2316)
Propriété PictureUp (Page 2315)
Propriété PictureDown (Page 2314)
Propriété PictureDeactivated (Page 2313)
Propriété PicDownUseTransColor (Page 2311)
Propriété PicDownTransparent (Page 2310)
Propriété PicDownReferenced (Page 2309)
Propriété PicDeactUseTransColor (Page 2309)
Propriété PicDeactTransparent (Page 2308)
Propriété PicDeactReferenced (Page 2307)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété BorderWidth (Page 2105)

Objet RoundRectangle

Description



Représente l'objet "Rectangle arrondi". L'objet RoundRectangle est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIRoundRectangle

Utilisation

Pour créer un nouvel objet "Rectangle arrondi" dans une vue, on utilise la méthode Add :

```
Sub AddRoundRectangle()
'VBA326
Dim objRoundRectangle As HMIRoundRectangle
Set objRoundRectangle = ActiveDocument.HMIObjects.AddHMIObject("Roundrectangle1",
"HMIRoundRectangle")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditRoundRectangle()
'VBA327
Dim objRoundRectangle As HMIRoundRectangle
Set objRoundRectangle = ActiveDocument.HMIObjects("Roundrectangle1")
objRoundRectangle.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

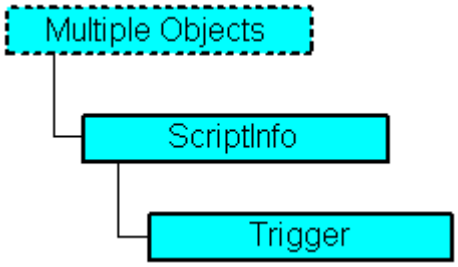
```
Sub ShowNameOfFirstSelectedObject()
'VBA328
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Width (Page 2472)
Propriété BorderBackColor (Page 2095)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété RoundCornerWidth (Page 2338)
Propriété RoundCornerHeight (Page 2337)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BackFlashColorOn (Page 2082)
Propriété BackFlashColorOff (Page 2081)
Propriété BackColor (Page 2076)

Objet ScriptInfo

Description



Représente un script (C ou VB) configuré pour dynamiser une propriété ou une action en fonction d'un événement.

Désignation VBA de l'objet

HMIScriptInfo

Utilisation

Pour dynamiser une propriété par un script, on peut p. ex. utiliser la méthode CreateDynamic. Exemple :...

```
Sub AddDynamicAsCSkriptToProperty()
'VBA329
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1", "HMICircle")
Set objCScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeCScript)
'
'Define triggertype and cycletime:
With objCScript
.SourceCode = ""
.Trigger.Type = hmiTriggerTypeStandardCycle
.Trigger.CycleType = hmiCycleType_2s
.Trigger.Name = "Trigger1"
End With
End Sub
```

Pour configurer une action déclenchée par un événement, on peut p. ex. utiliser la méthode AddAction. Exemple :...

```
Sub AddActionToPropertyTypeCScript()
'VBA330
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
```

```
'Add circle to picture. By changing of property "Radius"
'a C-Action is initiated:
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB", "HMICircle")
Set objEvent = objCircle.Radius.Events(1)
Set objCScript = objEvent.Actions.AddAction(hmiActionCreationTypeCScript)
End Sub
```

Voir aussi

Propriété Prototype (Page 2327)

Méthode Delete (Page 1807)

Référence VBA (Page 1724)

Création de dynamisations avec VBA (Page 1680)

Propriété Trigger (Page 2377)

Propriété SourceCode (Page 2357)

Propriété ScriptType (Page 2342)

Propriété Parent (Page 2303)

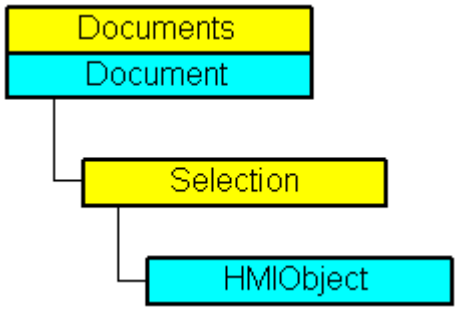
Propriété ActualPointTop (Page 2140)

Propriété Application (Page 2067)

Propriété UsedLanguage (Page 2387)

Objet Selection (liste)

Description



Listage de tous les objets HMIObject représentant les objets sélectionnés dans une vue.

Désignation VBA de l'objet

HMISelectedObjects

Utilisation

Pour restaurer le listage Selection, on utilise la propriété Selection. Dans l'exemple qui suit, on obtient le nom de tous les objets sélectionnés dans la vue active :

```
Sub ShowSelectionOfDocument()
'VBA331
Dim colSelection As HMISelectedObjects
Dim objObject As HMIObject
Dim strObjectList As String
Set colSelection = ActiveDocument.Selection
If colSelection.Count <> 0 Then
strObjectList = "List of selected objects:"
For Each objObject In colSelection
strObjectList = strObjectList & vbCrLf & objObject.ObjectName
Next objObject
    Else
strObjectList = "No objects selected"
    End If
MsgBox strObjectList
End Sub
```

Pour sélectionner tous les objets de la vue, on peut p. ex. utiliser la méthode SelectAll-Methode. Dans l'exemple qui suit, on sélectionne tous les objets de la vue active.

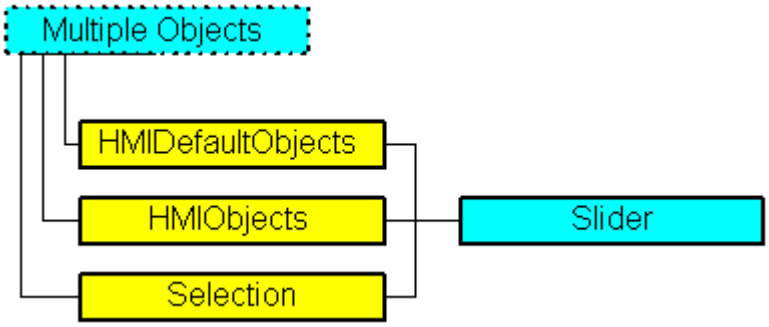
```
Sub SelectAllObjects()
'VBA332
ActiveDocument.Selection.SelectAll
End Sub
```

Voir aussi

Objets HMIObjects (liste) (Page 1945)
Méthode AlignTop (Page 1789)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode BringToFront (Page 1792)
Méthode SendToBack (Page 1857)
Méthode SelectAll (Page 1856)
Méthode SameWidthAndHeight (Page 1852)
Méthode SameWidth (Page 1850)
Méthode SameHeight (Page 1849)
Méthode Rotate (Page 1848)
Méthode Remove (Page 1846)
Méthode ForwardOneLevel (Page 1821)
Méthode BackwardOneLevel (Page 1791)
Méthode MoveSelection (Page 1841)
Méthode Item (Page 1834)
Méthode FlipVertically (Page 1820)
Méthode FlipHorizontally (Page 1819)
Méthode EvenlySpaceVertically (Page 1814)
Méthode EvenlySpaceHorizontally (Page 1813)
Méthode DuplicateSelection (Page 1812)
Méthode DeselectAll (Page 1810)
Méthode DeleteAll (Page 1808)
Méthode CreateGroup (Page 1805)
Méthode CreateCustomizedObject (Page 1802)
Méthode CopySelection (Page 1800)
Méthode CenterVertically (Page 1795)
Méthode CenterHorizontally (Page 1794)
Méthode AlignRight (Page 1788)
Méthode AlignLeft (Page 1787)
Méthode AlignBottom (Page 1786)
Pour éditer un objet utilisateur avec VBA (Page 1677)
Pour éditer les objets Groupe avec VBA (Page 1670)
Vous éditez de la sorte les objets standard, complexes, Windows et tuyaux (Page 1657)
Référence VBA (Page 1724)
Objets utilisateur (Page 1676)
Objets Groupe (Page 1668)
Objets standard, complexes, Windows et tuyaux (Page 1855)
Edition d'objets avec VBA (Page 1653)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)

Objet Slider

Description



Représente l'objet "Défileur". L'Défileur est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMISlider

Utilisation

Pour créer un nouvel "Défileur" dans une vue, on utilise la méthode Add :

```
Sub AddSlider()
'VBA333
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("Slider1", "HMISlider")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditSlider()
'VBA334
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects("Slider1")
objSlider.ButtonColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

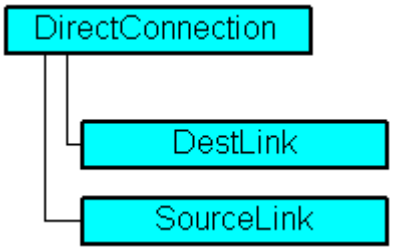
```
Sub ShowNameOfFirstSelectedObject()
'VBA335
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété OperationReport (Page 2300)
Propriété BorderFlashColorOff (Page 2100)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété SmallChange (Page 2355)
Propriété Process (Page 2323)
Propriété PasswordLevel (Page 2305)
Propriété OperationMessage (Page 2299)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Min (Page 2286)
Propriété Max (Page 2269)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété ExtendedOperation (Page 2163)
Propriété Direction (Page 2151)
Propriété ColorTop (Page 2135)
Propriété ColorBottom (Page 2128)
Propriété ButtonColor (Page 2112)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderColor (Page 2096)

Objet SourceLink

Description



Représente la source d'une liaison directe.

Désignation VBA de l'objet

HMISourceLink

Utilisation

On utilise la propriété SourceLink pour restaurer un objet SourceLink. Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub SetZoomAndScrollPositionInActiveView()
'VBA336
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDirConnection As HMIDirectConnection
'
'Add objects to active document:
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Text = "SetPosition"
End With
```

```
'
'Initiation of directconnection by mouseclick:
Set objDirConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDirConnection
'Sourceobject: Top-property of Rectangle_A
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Targetobject: Left-property of Rectangle_B
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
End Sub
```

Voir aussi

Objet DirectConnection (Page 1906)

Référence VBA (Page 1724)

Propriété Type (Page 2378)

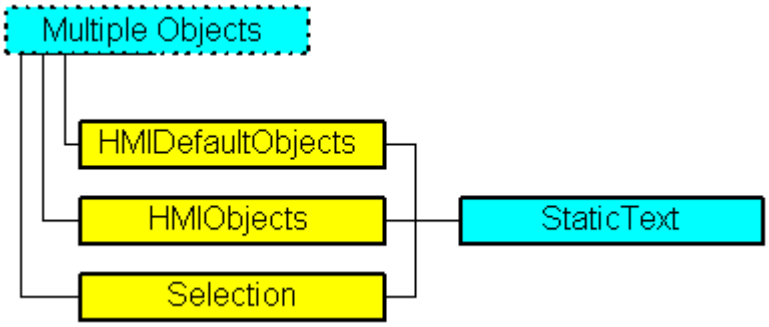
Propriété SourceLink (Page 2356)

Propriété ObjectName (Page 2293)

Propriété AutomationName (Page 2070)

Objet StaticText

Description



Représente l'objet "Texte statique". L'objet StaticText est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIStaticText

Utilisation

Pour créer un nouvel objet "Texte statique" dans une vue, on utilise la méthode Add :

```
Sub AddStaticText()
'VBA337
Dim objStaticText As HMIStaticText
Set objStaticText = ActiveDocument.HMIObjects.AddHMIObject("Static_Text1", "HMIStaticText")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditStaticText()
'VBA338
Dim objStaticText As HMIStaticText
Set objStaticText = ActiveDocument.HMIObjects("Static_Text1")
objStaticText.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

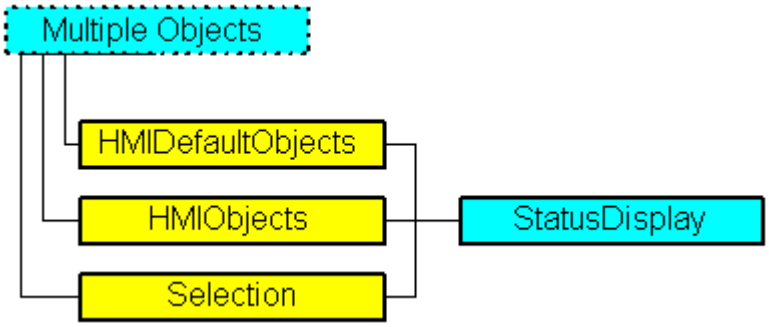
```
Sub ShowNameOfFirstSelectedObject()
'VBA339
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)
Propriété FontBold (Page 2188)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété Text (Page 2369)
Propriété PasswordLevel (Page 2305)
Propriété Orientation (Page 2301)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Height (Page 2201)
Propriété ForeFlashColorOn (Page 2195)
Propriété ForeFlashColorOff (Page 2194)
Propriété ForeColor (Page 2194)
Propriété FontUnderline (Page 2193)
Propriété FontSize (Page 2192)
Propriété FontName (Page 2191)
Propriété FontItalic (Page 2191)
Propriété FlashRateForeColor (Page 2185)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashRateBackColor (Page 2181)
Propriété FlashForeColor (Page 2176)
Propriété FlashBorderColor (Page 2174)
Propriété FlashBackColor (Page 2173)
Propriété FillStyle (Page 2169)
Propriété FillingIndex (Page 2167)
Propriété Filling (Page 2166)
Propriété FillColor (Page 2164)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)

Objet StatusDisplay

Description



Représente l'objet "Indicateur d'état". L'objet "StatusDisplay" est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMIStatusDisplay

Utilisation

Pour créer un nouvel objet "Indicateur d'état" dans une vue, on utilise la méthode Add :

```
Sub AddStatusDisplay()
'VBA340
Dim objStatusDisplay As HMIStatusDisplay
Set objStatusDisplay = ActiveDocument.HMIObjects.AddHMIObject("Statusdisplay1",
"HMIStatusDisplay")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditStatusDisplay()
'VBA341
Dim objStatusDisplay As HMIStatusDisplay
Set objStatusDisplay = ActiveDocument.HMIObjects("Statusdisplay1")
objStatusDisplay.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

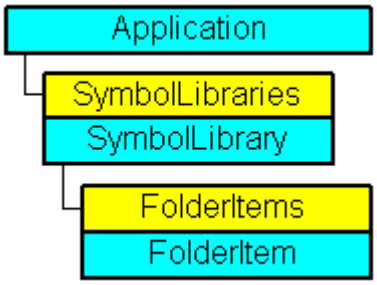
```
Sub ShowNameOfFirstSelectedObject()
'VBA342
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété TooltipText (Page 2373)
Propriété BasePicReferenced (Page 2087)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Width (Page 2472)
Propriété Visible (Page 2469)
Propriété Top (Page 2374)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété Name (Page 2289)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Index (Page 2207)
Propriété Height (Page 2201)
Propriété FlashRateFlashPic (Page 2184)
Propriété FlashRateBorderColor (Page 2182)
Propriété FlashPicUseTransColor (Page 2179)
Propriété FlashPicture (Page 2178)
Propriété FlashPicTransColor (Page 2177)
Propriété FlashPicReferenced (Page 2177)
Propriété FlashFlashPicture (Page 2175)
Propriété FlashBorderColor (Page 2174)
Propriété BorderWidth (Page 2105)
Propriété BorderStyle (Page 2103)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété BorderColor (Page 2096)
Propriété BorderBackColor (Page 2095)
Propriété BasePicUseTransColor (Page 2089)
Propriété BasePicture (Page 2088)
Propriété BasePicTransColor (Page 2087)

Objet SymbolLibrary

Description



Représente la "bibliothèque globale" ou la "bibliothèque du projet". L'objet SymbolLibrary appartient au listage SymbolLibraries.

Désignation VBA de l'objet

HMISymbolLibrary

Utilisation

On utilise SymbolLibraries(Index) pour renvoyer un objet SymbolLibrary individuel. Comme "Index", on peut utiliser le numéro d'index ou le nom de l'objet. Dans l'exemple qui suit, on obtient le nom de la "bibliothèque globale" :

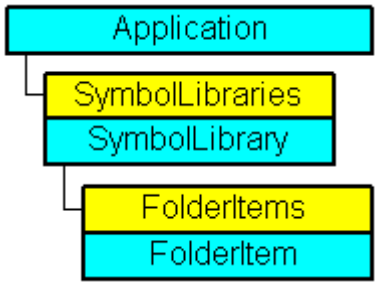
```
Sub ShowFirstObjectOfCollection()
'VBA343
Dim strName As String
strName = Application.SymbolLibraries(1).Name
MsgBox strName
End Sub
```

Voir aussi

Objet SymbolLibraries (liste) (Page 2024)
Méthode GetItemByPath (Page 1822)
Méthode FindByDisplayName (Page 1818)
Référence VBA (Page 1724)
Accès à la bibliothèque des modules avec VBA (Page 1640)
Propriété Parent (Page 2303)
Propriété Name (Page 2289)
Propriété FolderItems (Page 2187)
Propriété Application (Page 2067)

Objet SymbolLibraries (liste)

Description



Listage des objets SymbolLibrary représentant la bibliothèque des modules. Cette liste renferme deux objets : le premier est la "bibliothèque globale" et le second, la "bibliothèque du projet".

Désignation VBA de l'objet

HMISymbolLibraries

Utilisation

Pour renvoyer le listage SymbolLibraries, on utilise la propriété SymbolLibraries. Dans l'exemple qui suit, on obtient le nom des bibliothèques :

```
Sub ShowSymbolLibraries()
'VBA344
Dim colSymbolLibraries As HMISymbolLibraries
Dim objSymbolLibrary As HMISymbolLibrary
Dim strLibraryList As String
Set colSymbolLibraries = Application.SymbolLibraries
```

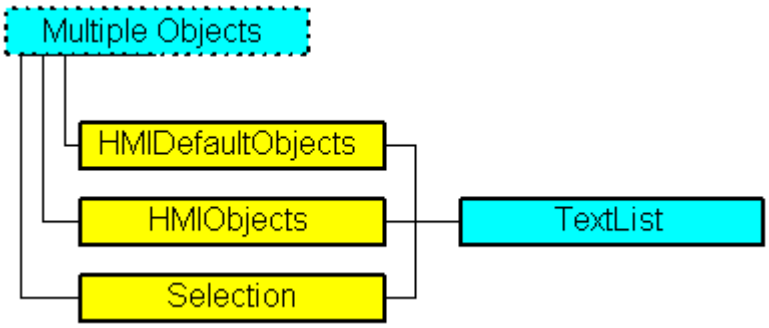
```
For Each objSymbolLibrary In colSymbolLibraries
strLibraryList = strLibraryList & objSymbolLibrary.Name & vbCrLf
Next objSymbolLibrary
MsgBox strLibraryList
End Sub
```

Voir aussi

Objet SymbolLibrary (Page 2023)
Méthode Item (Page 1834)
Référence VBA (Page 1724)
Accès à la bibliothèque des modules avec VBA (Page 1640)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

Objet TextList

Description



Représente l'objet "Liste d'objets". L'objet TextList est un élément des listages suivants :

- Objects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme le paramétrage par défaut des valeurs de propriété de tous les objets simples, complexes et Windows.

Désignation VBA de l'objet

HMITextList

Utilisation

Pour créer un nouvel objet "Liste d'objets" dans une vue, on utilise la méthode Add :

```
Sub AddTextList()
'VBA345
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("Textlist1", "HMITextList")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" étant ici le nom de l'objet :

```
Sub EditTextList()
'VBA346
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects("Textlist1")
objTextList.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

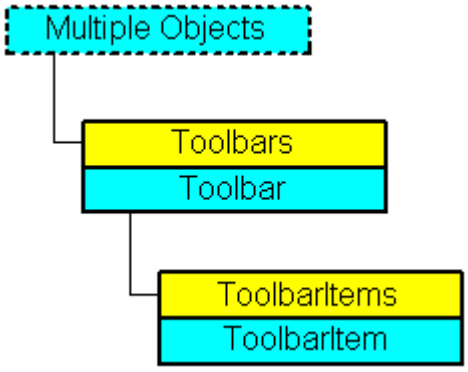
```
Sub ShowNameOfFirstSelectedObject()
'VBA347
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name of the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété Width (Page 2472)
Propriété ForeFlashColorOn (Page 2195)
Propriété BitNumber (Page 2092)
Objet Selection (liste) (Page 2010)
Objets HMIObjects (liste) (Page 1945)
Objet HMIDefaultObjects (liste) (Page 1939)
Méthode AddHMIObject (Page 1780)
Référence VBA (Page 1724)
Edition d'objets avec VBA (Page 1653)
Propriété Visible (Page 2469)
Propriété UnselTextColor (Page 2390)
Propriété UnselBGColor (Page 2389)
Propriété Top (Page 2374)
Propriété TooltipText (Page 2373)
Propriété SelTextColor (Page 2349)
Propriété SelBGColor (Page 2347)
Propriété PasswordLevel (Page 2305)
Propriété OutputValue (Page 2302)
Propriété Orientation (Page 2301)
Propriété OperationReport (Page 2300)
Propriété OperationMessage (Page 2299)
Propriété Operation (Page 2298)
Propriété NumberLines (Page 2292)
Propriété Name (Page 2289)
Propriété ListType (Page 2260)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété LanguageSwitch (Page 2221)
Propriété ItemBorderWidth (Page 2216)
Propriété ItemBorderStyle (Page 2216)
Propriété ItemBorderColor (Page 2215)
Propriété ItemBorderBackColor (Page 2214)
Propriété Height (Page 2201)
Propriété ForeFlashColorOff (Page 2194)
Propriété ForeColor (Page 2194)
Propriété FontUnderline (Page 2193)
Propriété FontSize (Page 2192)
Propriété FontName (Page 2191)
Propriété FontItalic (Page 2191)
Propriété FontBold (Page 2188)

Objet Toolbar

Description



Représente l'objet "Barre d'outils personnalisée". L'objet Toolbar appartient au listage CustomToolbars.

Désignation VBA de l'objet

HMIToolbar

Utilisation

On utilise CustomToolbars(Index) pour renvoyer un objet Toolbar individuel. Comme "Index", on peut utiliser le numéro d'index ou le nom de l'objet. Dans l'exemple qui suit, on affiche le paramètre "Key" de la première barre d'outils personnalisée de la vue active :

```
Sub ShowFirstObjectOfCollection()
'VBA348
Dim strName As String
strName = ActiveDocument.CustomToolbars(1).Key
MsgBox strName
End Sub
```

Pour supprimer un objet "Toolbar" du listage "CustomToolbars", on utilise la méthode Delete. Dans l'exemple qui suit, on supprime la première barre d'outils personnalisée de la vue active :

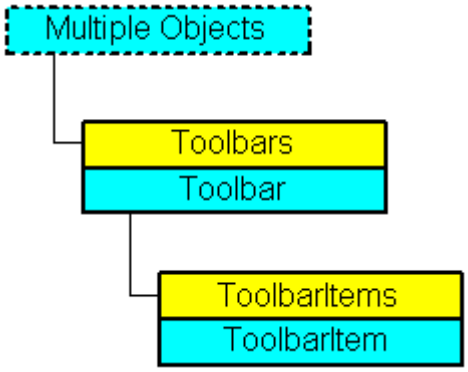
```
Sub DeleteToolbar()
'VBA349
Dim objToolbar As HMIToolbar
Set objToolbar = ActiveDocument.CustomToolbars(1)
objToolbar.Delete
End Sub
```

Voir aussi

Propriété Key (Page 2217)
Objet Toolbars (liste) (Page 2029)
Méthode Delete (Page 1807)
Pour créer des menus et des barres d'outils spécifiques à une vue (Page 1648)
Pour créer une nouvelle barre d'outils spécifique à une application (Page 1629)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)
Propriété Visible (Page 2469)
Propriété ToolbarItems (Page 2372)
Propriété Parent (Page 2303)
Propriété Application (Page 2067)

Objet Toolbars (liste)

Description



Listage de tous les objets Toolbar représentant des barres d'outils personnalisées dans le Graphics Designer.

Désignation VBA de l'objet

HMICustomToolbars

Utilisation

Remarque

Pour que les exemples s'appliquent, vous devez d'abord créer une barre d'outils personnalisée. Un exemple d'une telle création est donné p. ex. à la rubrique "Création d'une nouvelle barre d'outils personnalisée".

Pour renvoyer le listage Toolbars, on utilise la propriété CustomToolbars. Dans l'exemple qui suit, on obtient la valeur de la propriété Key de toutes les barres d'outils personnalisées de la vue active.

Remarque

Le listage Toolbars obtenu ne distingue pas entre les barres d'outils spécifiques d'une application et celles spécifiques d'une vue.

```
Sub ShowCustomToolbarsOfDocument()
'VBA350
Dim colToolbars As HMIToolbars
Dim objToolbar As HMIToolbar
Dim strToolbarList As String
Set colToolbars = ActiveDocument.CustomToolbars
If 0 <> colToolbars.Count Then
For Each objToolbar In colToolbars
strToolbarList = strToolbarList & objToolbar.Key & vbCrLf
Next objToolbar
Else
strToolbarList = "No toolbars existing"
End If
MsgBox strToolbarList
End Sub
```

Pour créer une barre d'outils spécifique d'une application, on utilise la propriété Application et la méthode Add. Le code VBA est créé soit dans le document "Project Template Document", soit dans "Global Template Document".

```
Sub InsertApplicationSpecificToolbar()
'VBA351
Dim objToolbar As HMIToolbar
Set objToolbar = Application.CustomToolbars.Add("a_Toolbar1")
End Sub
```

Pour créer une barre d'outils spécifique d'une vue, on utilise la propriété ActiveDocument et la méthode Add. Le code VBA est à créer dans le document "ThisDocument".

```
Sub InsertDocumentSpecificToolbar()
'VBA352
```

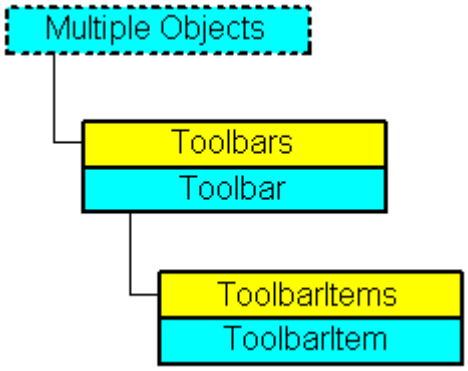
```
Dim objToolbar As HMIToolbar
Set objToolbar = ActiveDocument.CustomToolbars.Add("d_Toolbar1")
End Sub
```

Voir aussi

Objet Toolbar (Page 2028)
Méthode Item (Page 1834)
Méthode Add (Page 1765)
Pour créer des menus et des barres d'outils spécifiques à une vue (Page 1648)
Pour créer une nouvelle barre d'outils spécifique à une application (Page 1629)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)
Propriété Parent (Page 2303)
Propriété Count (Page 2141)
Propriété Application (Page 2067)

Objet ToolbarItem

Description



Représente un objet (icône ou ligne de séparation) d'une barre d'outils personnalisée dans Graphics Designer. L'objet ToolbarItem appartient au listage ToolbarItems.

Désignation VBA de l'objet

HMIToolbarItem

Utilisation

Remarque

Pour que les exemples s'appliquent, vous devez d'abord créer une barre d'outils personnalisée. Un exemple d'une telle création est donné p. ex. à la rubrique "Création d'une nouvelle barre d'outils personnalisée".

On utilise ToolbarItems(Index) pour renvoyer un objet ToolbarItem individuel. Comme "Index", on peut utiliser le numéro d'index ou le nom de l'objet. Dans l'exemple qui suit, on obtient le type du premier objet de la première barre d'outils personnalisée de la vue active :

```
Sub ShowFirstObjectOfCollection()
'VBA353
Dim strType As String
strType = ActiveDocument.CustomToolbars(1).ToolbarItems(1).ToolbarItemType
MsgBox strType
End Sub
```

Pour supprimer un objet du listage "ToolbarItems", on utilise la méthode Delete. Dans l'exemple qui suit, on supprime la première entrée de la première barre d'outils personnalisée de la vue active :

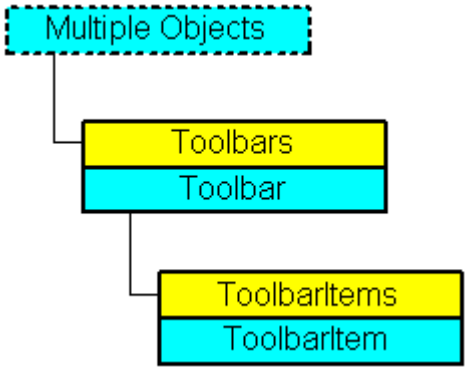
```
Sub DeleteToolbarItem()
'VBA354
ActiveDocument.CustomToolbars(1).ToolbarItems(1).Delete
End Sub
```

Voir aussi

Propriété Macro (Page 2267)
Objet ToolbarItems (liste) (Page 2034)
Méthode Delete (Page 1807)
Configuration des menus et des barres d'outils (Page 1619)
Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)
Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)
Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)
Référence VBA (Page 1724)
Création de menus et de barres d'outils personnalisés (Page 1620)
Propriété Visible (Page 2469)
Propriété Type (Page 2378)
Propriété TooltipText (Page 2373)
Propriété Tag (Page 2366)
Propriété StatusText (Page 2359)
Propriété ShortCut (Page 2351)
Propriété Position (Page 2320)
Propriété Parent (Page 2303)
Propriété LDTooltipTexts (Page 2252)
Propriété LDStatusTexts (Page 2250)
Propriété Key (Page 2217)
Propriété Icon (Page 2206)
Propriété Enabled (Page 2158)
Propriété Application (Page 2067)

Objet ToolbarItems (liste)

Description



Un listage d'objets ToolbarItem représentant tous les objets d'une barre d'outils définie par l'utilisateur.

Désignation VBA de l'objet

HMIToolbarItems

Utilisation

Utilisez la propriété ToolbarItems pour restaurer le listage ToolbarItems. Dans l'exemple suivant, tous les types d'objet de la première barre d'outils définie par l'utilisateur s'affichent dans la vue active.

Remarque

Le listage ToolbarIterms obtenu ne distingue pas entre les barres d'outils spécifiques d'une application et celles spécifiques d'une vue.

```
Sub ShowToolbarItems()
'VBA355
Dim colToolbarItems As HMIToolbarItems
Dim objToolbarItem As HMIToolbarItem
Dim strTypeList As String
Set colToolbarItems = ActiveDocument.CustomToolbars(1).ToolbarItems
If 0 <> colToolbarItems.Count Then
For Each objToolbarItem In colToolbarItems
strTypeList = strTypeList & objToolbarItem.ToolbarItemType & vbCrLf
Next objToolbarItem
    Else
strTypeList = "No Toolbaritems existing"
    End If
MsgBox strTypeList
```

```
End Sub
```

Utilisez par exemple la méthode InsertToolbarItem pour ajouter une icône dans une barre d'outils personnalisée. Dans l'exemple suivant, une barre d'outils spécifique à une vue est créée dans la vue active et une icône est ajoutée :

```
Sub InsertToolbarItem()
'VBA356
Dim objToolbar As HMIToolbar
Dim objToolbarItem As HMIToolbarItem
Set objToolbar = ActiveDocument.CustomToolbars.Add("d_Toolbar2")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(1, "t_Item2_1",
"ToolbarItem 1")
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Méthode InsertToolbarItem (Page 1831)

Méthode InsertSeparator (Page 1828)

Méthode InsertFromMenuItem (Page 1823)

Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)

Référence VBA (Page 1724)

Création de menus et de barres d'outils personnalisés (Page 1620)

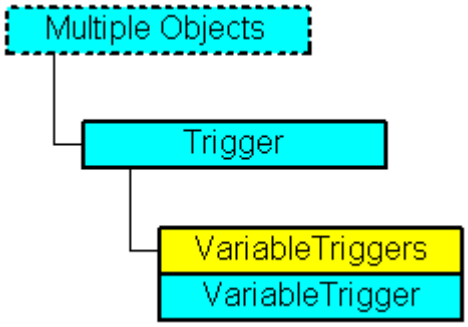
Propriété Parent (Page 2303)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet Trigger

Description



Représente le Trigger (exemple : cycle de vue) nécessaire pour dynamiser des propriétés avec des scripts. Un Trigger peut posséder plusieurs variables.

Désignation VBA de l'objet

HMITrigger

Utilisation

Utilisez la propriété Trigger pour restaurer l'objet Trigger. Dans cet exemple, la propriété "Radius" d'un cercle est dynamisée avec un script VB (la valeur de retour active le rayon) :

```
Sub AddDynamicAsVBSkriptToProperty()
'VBA357
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1", "HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
'
'Define cycletime and sourcecode
With objVBScript
.SourceCode = ""
.Trigger.Type = hmiTriggerTypeStandardCycle
.Trigger.CycleType = hmiCycleType_2s
.Trigger.Name = "Trigger1"
End With
End Sub
```

Voir aussi

Méthode Delete (Page 1807)

Référence VBA (Page 1724)

Propriété VariableTriggers (Page 2466)

Propriété Type (Page 2378)

Propriété Trigger (Page 2377)

Propriété Parent (Page 2303)

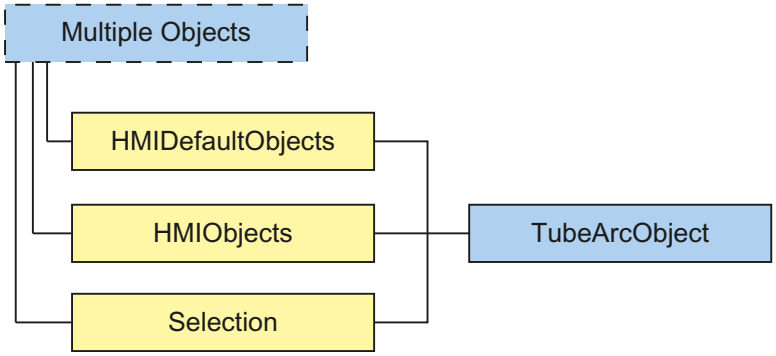
Propriété Name (Page 2289)

Propriété CycleType (Page 2148)

Propriété Application (Page 2067)

Objet TubeArcObject

Description



Représente l'objet "Coude de tuyau". L'objet TubeArcObject est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMITubeArcObject

Utilisation

Pour créer un nouvel objet "Coude de tuyau" dans la vue, on utilise la méthode Add :

```
Sub AddTubeArcObject()
'VBA835
Dim objTubeArcObject As HMITubeArcObject
Set objTubeArcObject = ActiveDocument.HMIObjects.AddHMIObject("Coude de tuyau",
"HMITubeArcObject")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditTubeArcObject()
'VBA836
Dim objTubeArcObject As HMITubeArcObject
Set objTubeArcObject = ActiveDocument.HMIObjects("Coude de tuyau")
objTubeArcObject.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

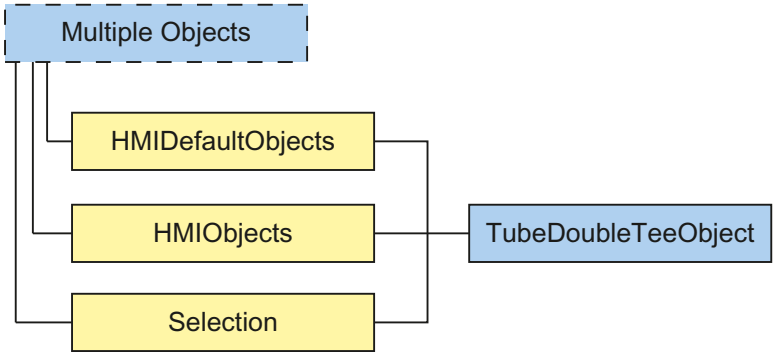
```
Sub ShowNameOfFirstSelectedObject()
'VBA837
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ObjectName (Page 2293)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Top (Page 2374)
Propriété Width (Page 2472)
Propriété Height (Page 2201)
Propriété BorderColor (Page 2096)
Propriété BorderWidth (Page 2105)
Propriété TooltipText (Page 2373)
Propriété Visible (Page 2469)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété FlashRateBorderColor (Page 2182)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété FlashBorderColor (Page 2174)
Propriété Transparency (Page 2375)
Propriété GlobalShadow (Page 2197)
Propriété GlobalColorScheme (Page 2196)
Propriété StartAngle (Page 2359)
Propriété EndAngle (Page 2158)
Propriété RadiusHeight (Page 2331)
Propriété RadiusWidth (Page 2331)

Objet TubeDoubleTeeObject

Description



Représente l'objet "Double raccord en T". L'objet TubeDoubleTeeObject est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMITubeDoubleTeeObject

Utilisation

Pour créer un nouvel objet "Double raccord en T" dans la vue, on utilise la méthode Add :

```
Sub AddTubeDoubleTeeObject()
'VBA838
Dim objTubeDoubleTeeObject As HMITubeDoubleTeeObject
Set objTubeDoubleTeeObject = ActiveDocument.HMIObjects.AddHMIObject("Double raccord en T",
"HMITubeDoubleTeeObject")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditTubeDoubleTeeObject()
'VBA839
Dim objTubeDoubleTeeObject As HMITubeDoubleTeeObject
Set objTubeDoubleTeeObject = ActiveDocument.HMIObjects("Double raccord en T")
objTubeDoubleTeeObject.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

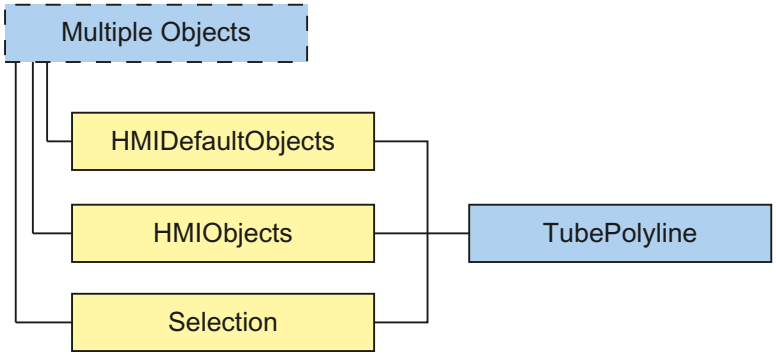
```
Sub ShowNameOfFirstSelectedObject()
'VBA840
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ObjectName (Page 2293)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Top (Page 2374)
Propriété Width (Page 2472)
Propriété Height (Page 2201)
Propriété BorderColor (Page 2096)
Propriété BorderWidth (Page 2105)
Propriété TooltipText (Page 2373)
Propriété Visible (Page 2469)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété FlashRateBorderColor (Page 2182)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété FlashBorderColor (Page 2174)
Propriété Transparency (Page 2375)
Propriété GlobalShadow (Page 2197)
Propriété GlobalColorScheme (Page 2196)

Objet TubePolyline

Description



Représente l'objet "Tuyau polygonal". L'objet TubePolyline est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMITubePolyline

Utilisation

Pour créer un nouvel objet "Tuyau polygonal" dans la vue, on utilise la méthode Add :

```
Sub AddTubePolyline()
'VBA841
Dim objTubePolyline As HMITubePolyline
Set objTubePolyline = ActiveDocument.HMIObjects.AddHMIObject("Tuyau polygonal",
"HMITubePolyline")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditTubePolyline()
'VBA842
Dim objTubePolyline As HMITubePolyline
Set objTubePolyline = ActiveDocument.HMIObjects("Tuyau polygonal")
objTubePolyline.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

```
Sub ShowNameOfFirstSelectedObject()
'VBA843
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ObjectName (Page 2293)

Propriété Left (Page 2253)

Propriété Layer (Page 2222)

Propriété Top (Page 2374)

Propriété Width (Page 2472)

Propriété Height (Page 2201)

Propriété BorderColor (Page 2096)

Propriété BorderWidth (Page 2105)

Propriété TooltipText (Page 2373)

Propriété Visible (Page 2469)

Propriété PasswordLevel (Page 2305)

Propriété Operation (Page 2298)

Propriété FlashRateBorderColor (Page 2182)

Propriété BorderFlashColorOn (Page 2102)

Propriété BorderFlashColorOff (Page 2100)

Propriété FlashBorderColor (Page 2174)

Propriété Transparency (Page 2375)

Propriété GlobalShadow (Page 2197)

Propriété GlobalColorScheme (Page 2196)

Propriété PointCount (Page 2319)

Propriété ActualPointLeft (Page 2056)

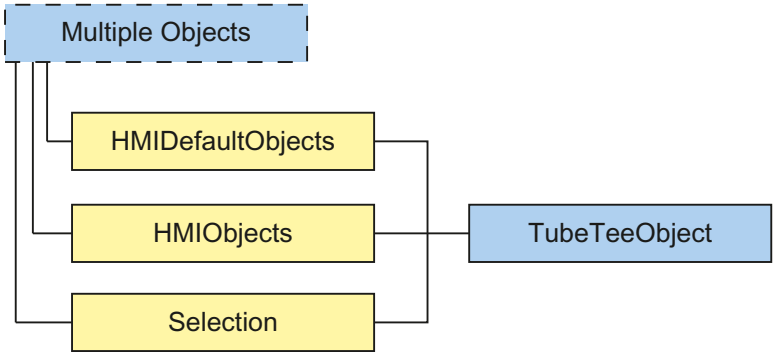
Propriété ActualPointTop (Page 2057)

Propriété Index (Page 2207)

Propriété LineJoinStyle (Page 2260)

Objet TubeTeeObject

Description



Représente l'objet "Raccord en T". L'objet TubeTeeObject est un élément des listes suivantes :

- HMIObjects : renferme tous les objets d'une vue.
- Selection : renferme tous les objets d'une vue qui sont sélectionnés.
- HMIDefaultObjects : renferme les paramétrages par défaut des valeurs de propriété de tous les objets standard, complexes, Windows et tuyaux.

Désignation VBA de l'objet

HMITubeTeeObject

Utilisation

Pour créer un nouvel objet "Raccord en T" dans la vue, on utilise la méthode Add :

```
Sub AddTubeTeeObject()
'VBA844
Dim objTubeTeeObject As HMITubeTeeObject
Set objTubeTeeObject = ActiveDocument.HMIObjects.AddHMIObject("Raccord en T",
"HMITubeTeeObject")
End Sub
```

Pour renvoyer un objet de la liste HMIObjects, on utilise la méthode "HMIObjects(Index)", "index" identifiant ici l'objet avec son nom :

```
Sub EditTubeTeeObject()
'VBA845
Dim objTubeTeeObject As HMITubeTeeObject
Set objTubeTeeObject = ActiveDocument.HMIObjects("Raccord en T")
objTubeTeeObject.BorderColor = RGB(255, 0, 0)
End Sub
```

Pour renvoyer un objet de la liste Selection, on utilise la méthode "Selection(Index)" :

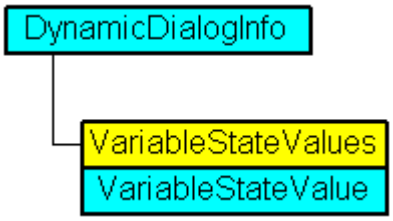
```
Sub ShowNameOfFirstSelectedObject()
'VBA846
'Select all objects in the picture:
ActiveDocument.Selection.SelectAll
'Get the name from the first object of the selection:
MsgBox ActiveDocument.Selection(1).ObjectName
End Sub
```

Voir aussi

Propriété ObjectName (Page 2293)
Propriété Left (Page 2253)
Propriété Layer (Page 2222)
Propriété Top (Page 2374)
Propriété Width (Page 2472)
Propriété Height (Page 2201)
Propriété BorderColor (Page 2096)
Propriété BorderWidth (Page 2105)
Propriété TooltipText (Page 2373)
Propriété Visible (Page 2469)
Propriété PasswordLevel (Page 2305)
Propriété Operation (Page 2298)
Propriété FlashRateBorderColor (Page 2182)
Propriété BorderFlashColorOn (Page 2102)
Propriété BorderFlashColorOff (Page 2100)
Propriété FlashBorderColor (Page 2174)
Propriété Transparency (Page 2375)
Propriété GlobalShadow (Page 2197)
Propriété GlobalColorScheme (Page 2196)
Propriété RotationAngle (Page 2336)

Objet VariableStateValue

Description



Représente l'état d'une variable à laquelle est affectée dans le dialogue dynamique une valeur utilisée pour la dynamisation.

Désignation VBA de l'objet

HMIVariableStateValue

Voir aussi

Propriété VALUE_SERVERDOWN (Page 2437)

Référence VBA (Page 1724)

Propriété VarName (Page 2467)

Propriété VALUE_TIMEOUT (Page 2440)

Propriété VALUE_STARTUP_VALUE (Page 2439)

Propriété VALUE_NOT_ESTABLISHED (Page 2436)

Propriété VALUE_MIN_RANGE (Page 2434)

Propriété VALUE_MIN_LIMIT (Page 2433)

Propriété VALUE_MAX_RANGE (Page 2431)

Propriété VALUE_MAX_LIMIT (Page 2429)

Propriété VALUE_INVALID_KEY (Page 2425)

Propriété VALUE_HARDWARE_ERROR (Page 2422)

Propriété VALUE_HANDSHAKE_ERROR (Page 2420)

Propriété VALUE_CONVERSION_ERROR (Page 2418)

Propriété VALUE_ADDRESS_ERROR (Page 2397)

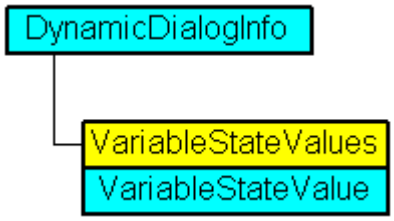
Propriété VALUE_ACCESS_FAULT (Page 2395)

Propriété Parent (Page 2303)

Propriété Application (Page 2067)

Objet VariableStateValues (liste)

Description



Un listage d'objets VariableStateValue contenant tous les états de variables dans le dialogue dynamique utilisés pour la dynamisation.

Désignation VBA de l'objet

HMIVariableStateValues

Utilisation

Utilisez par exemple la propriété Item pour définir dans le dialogue dynamique les valeurs utilisées pour la dynamisation si la variable indiquée renvoie l'état configuré. Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA358
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'Activate variable-statecheck
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
```

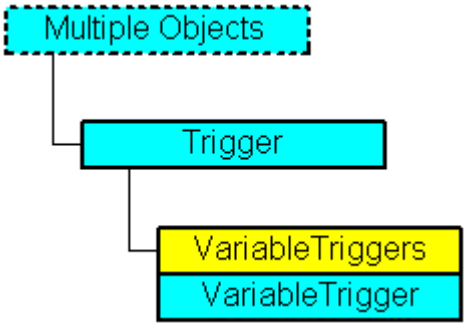
```
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VALUE_MAX_RANGE (Page 2431)
Référence VBA (Page 1724)
Propriété VarName (Page 2467)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Propriété Parent (Page 2303)
Propriété Item (Page 2213)
Propriété Application (Page 2067)

Objet VariableTrigger

Description



Représente un déclenchement en fonction d'une variable.

Désignation VBA de l'objet

HMIVariableTrigger

Utilisation

Utilisez l'objet VariableTrigger pour éditer ou supprimer un déclenchement en fonction d'une variable. Dans cet exemple, on dynamise la propriété "Top" d'un cercle par la variable "Otto" :

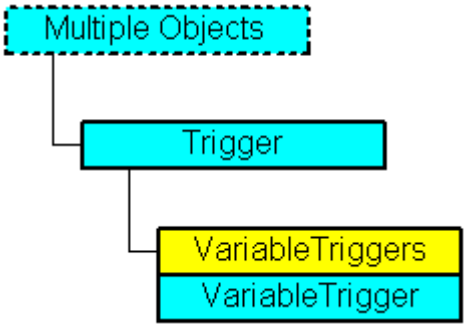
```
Sub AddDynamicAsVariableDirectToProperty()
'VBA359
Dim objVariableTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1", "HMICircle")
Set objVariableTrigger = objCircle.Top.CreateDynamic(hmiDynamicCreationTypeVariableDirect,
"'NewDynamic1'")
'
'Define cycletime
With objVariableTrigger
.CycleType = hmiCycleType_2s
End With
End Sub
```

Voir aussi

Méthode Delete (Page 1807)
Objet VariableTriggers (liste) (Page 2049)
Référence VBA (Page 1724)
Propriété VariableTriggers (Page 2466)
Propriété Type (Page 2378)
Propriété Parent (Page 2303)
Propriété Name (Page 2289)
Propriété CycleType (Page 2148)
Propriété Application (Page 2067)

Objet VariableTriggers (liste)

Description



Un listage d'objets VariableTrigger représentant tous les déclenchements en fonction de variables utilisés.

Désignation VBA de l'objet

HMIVariableTriggers

Utilisation

Utilisez la méthode Add par exemple pour créer un nouveau déclenchement en fonction d'une variable. Dans l'exemple suivant, le rayon du cercle est dynamisé avec un script VB. Le déclenchement utilisé est du type en fonction d'une variable :

```
Sub DynamicWithVariableTriggerCycle()
'VBA360
Dim objVBScript As HMIScriptInfo
Dim objVarTrigger As HMIVariableTrigger
```

```
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_VariableTrigger",
"HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
With objVBScript
'Definition of triggername and cycletime is to do with the Add-methode
Set objVarTrigger = .Trigger.VariableTriggers.Add("VarTrigger", hmiVariableCycleType_10s)
.SourceCode = ""
End With
End Sub
```

Voir aussi

Méthode Add (listage VariableTriggers) (Page 1772)

Référence VBA (Page 1724)

Propriété Parent (Page 2303)

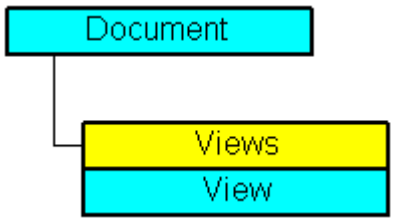
Propriété Item (Page 2213)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

Objet View

Description



Représente la copie d'une vue. L'objet View est un élément du listage des vues.

Avec les propriétés de l'objet View, vous pouvez entre autre piloter la visibilité des niveaux CS et définir le zoom.

Désignation VBA de l'objet

HMIView

Utilisation

Utilisez les Views(Index) pour renvoyer un objet View. Dans l'exemple suivant, le nombre des copies de la vue active est affiché :

```
Sub ShowNumberOfExistingViews()
'VBA361
Dim iMaxViews As Integer
iMaxViews = ActiveDocument.Views.Count
MsgBox "Number of copies from active document: " & iMaxViews
End Sub
```

Utilisez la méthode Add pour ajouter un nouvel objet View au listage "Views". Dans l'exemple qui suit, la vue active est copiée et la copie est ensuite activée :

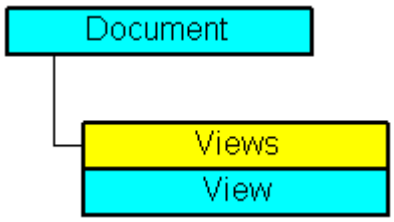
```
Sub AddView()
'VBA362
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
End Sub
```

Voir aussi

Propriété Height (Page 2201)
Objet Views (liste) (Page 2052)
Méthode SetCSLayerVisible (Page 1859)
Méthode PrintDocument (Page 1844)
Méthode IsCSLayerVisible (Page 1832)
Méthode Delete (Page 1807)
Méthode Add (listage Views) (Page 1773)
Méthode Activate (Page 1765)
Référence VBA (Page 1724)
Edition d'une copie de vue avec VBA (Page 1651)
Edition de niveaux avec VBA (Page 1650)
Propriété ExtendedZoomingEnable (Page 2163)
Propriété Zoom (Page 2478)
Propriété WindowState (Page 2476)
Propriété Width (Page 2472)
Propriété Top (Page 2374)
Propriété ScrollPosY (Page 2346)
Propriété ScrollPosX (Page 2345)
Propriété Parent (Page 2303)
Propriété Left (Page 2253)
Propriété IsActive (Page 2210)
Propriété Application (Page 2067)
Propriété ActiveLayer (Page 2056)

Objet Views (liste)

Description



Un listage d'objets View représentant la copie d'une vue.

Désignation VBA de l'objet

HMIViews

Utilisation

Utilisez le listage Views pour renvoyer un objet View. Dans l'exemple suivant, le nombre des copies existantes de la vue active est affiché :

```
Sub ShowNumberOfExistingViews()
'VBA363
Dim iMaxViews As Integer
iMaxViews = ActiveDocument.Views.Count
MsgBox "Number of copies from active document: " & iMaxViews
End Sub
```

Utilisez la méthode Add pour générer la copie d'une vue. Dans l'exemple qui suit, la vue active est copiée et la copie est ensuite activée :

```
Sub AddViewToActiveDocument()
'VBA364
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
End Sub
```

Voir aussi

Méthode Item (Page 1834)

Objet View (Page 2050)

Méthode Add (Page 1765)

Référence VBA (Page 1724)

Propriété Parent (Page 2303)

Propriété Count (Page 2141)

Propriété Application (Page 2067)

3.5.1.8 Propriétés

A

Propriété Actions

Description

Renvoie le listage Actions. Utilisez la propriété Actions pour configurer une action pilotée par des événements.

Exemple

Dans l'exemple qui suit, on insère un bouton et un cercle dans la vue active. Dans Runtime, le rayon du cercle augmente à chaque clic sur le bouton :

```
Sub CreateVBActionToClickedEvent()
'VBA365
Dim objButton As HMIButton
Dim objCircle As HMICircle
Dim objEvent As HMIEvent
Dim objVBScript As HMIScriptInfo
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_VB", "HMICircle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircle
.Top = 100
.Left = 100
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 120
.Text = "Increase Radius"
End With
'Define event and assign sourcecode:
Set objVBScript = objButton.Events(1).Actions.AddAction(hmiActionCreationTypeVBScript)
With objVBScript
.SourceCode = "Dim myCircle" & vbCrLf & _
              "Set myCircle = HMIRuntime.ActiveScreen.ScreenItems(""Circle_VB"")" & _
              vbCrLf & "myCircle.Radius = myCircle.Radius + 5"
End With
End Sub
```

Voir aussi

Objet Actions (liste) (Page 1872)

Méthode AddAction (Page 1773)

Configuration d'actions commandées par événement avec VBA (Page 1693)

Propriété ActiveDocument

Description

Renvoie un objet de type "Document" qui représente la vue active dans Graphics Designer. S'il n'y a pas de vue active ou ouverte dans Graphics Designer, un message d'erreur apparaît.

Remarque

La propriété "ActiveDocument" fait référence à la fenêtre possédant le focus de saisie. Si d'autres éditeurs (par ex. CrossReference) accèdent à une vue, le focus de saisie peut changer. Pour éviter l'erreur qui se produit alors, attribuez un nom univoque à la vue à l'aide du listage Documents.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Delete Objects" et d'ajouter deux options ("Delete Rectangles" et "Delete Circles") :

```
Sub CreateMenuItem()
'VBA366
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
'
'Create new menu "Delete Objects":
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete Objects")
'
'Add two menuitems to the menu "Delete Objects
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
Rectangles")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete Circles")
End Sub
```

Voir aussi

Objet Documents (liste) (Page 1911)

Propriété ActiveLayer

Description

Définit ou renvoie le plan actif pour l'objet View. Plage de valeurs de 0 à 31, "0" représentant le plan supérieur et "31", le plan inférieur.

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans l'exemple qui suit, un nouvel objet View est créé ou le plan 1 reçoit l'état "actif" :

```
Sub ActiveDocumentConfiguration()
'VBA367
Application.ActiveDocument.Views.Add
Application.ActiveDocument.Views(1).ActiveLayer = 2
End Sub
```

Voir aussi

Objet View (Page 2050)

Propriété ActualPointLeft

Description

Définit ou renvoie pour les objets "Polygone" et "Trait polygonal" la coordonnée x de l'angle actuel par rapport à l'origine de la vue (en haut à gauche). Chaque point d'angle est identifié par un indice dérivé du nombre ("PointCount") de points d'angle existants.

Une modification de la valeur peut avoir des répercussions sur les propriétés "Width" (largeur de l'objet) et "Left" (coordonnée x de la position de l'objet).

Exemple

La procédure "PolygonCoordinatesOutput()" permet de sortie les coordonnées de tous les angles du premier trait polygonal de la vue en cours :

```
Sub PolygonCoordinatesOutput()
'VBA368
Dim objPolyline As HMIPolyLine
Dim iPosX As Integer
Dim iPosY As Integer
Dim iCounter As Integer
Dim strResult As String
iCounter = 1
```

```
Set objPolyline = ActiveDocument.HMIObjects.AddHMIObject("Polyline1", "HMIPolyLine")
For iCounter = 1 To objPolyline.PointCount
With objPolyline
.index = iCounter
iPosX = .ActualPointLeft
iPosY = .ActualPointTop
End With
strResult = strResult & vbCrLf & "Corner " & iCounter & ": x=" & iPosX & " y=" & iPosY
Next iCounter
MsgBox strResult
End Sub
```

Voir aussi

Propriété PointCount (Page 2319)

Propriété Index (Page 2207)

Propriété ActualPointTop (Page 2057)

Objet PolyLine (Page 1989)

Objet Polygon (Page 1986)

Objet Line (Page 1958)

Propriété ActualPointTop

Description

Définit ou renvoie pour les objets "Polygone" et "Trait polygonal" la coordonnée y de l'angle actuel par rapport à l'origine de la vue (en haut à gauche). Chaque point d'angle est identifié par un indice dérivé du nombre ("PointCount") de points d'angle existants.

Une modification de la valeur peut avoir des répercussions sur les propriétés "Height" (hauteur de l'objet) et "Top" (coordonnée y de la position).

Exemple

La procédure "Polygon()" permet de sortie les coordonnées de tous les angles du premier trait polygonal de la vue en cours :

```
Sub PolygonCoordinatesOutput()
'VBA369
Dim objPolyline As HMIPolyLine
Dim iPosX As Integer
Dim iPosY As Integer
Dim iCounter As Integer
Dim strResult As String
iCounter = 1
Set objPolyline = ActiveDocument.HMIObjects.AddHMIObject("Polyline1", "HMIPolyLine")
```

```
For iCounter = 1 To objPolyline.PointCount
With objPolyline
.index = iCounter
iPosX = .ActualPointLeft
iPosY = .ActualPointTop
End With
strResult = strResult & vbCrLf & "Corner " & iCounter & ": x=" & iPosX & " y=" & iPosY
Next iCounter
MsgBox strResult
End Sub
```

Voir aussi

Propriété PointCount (Page 2319)

Propriété Index (Page 2207)

Propriété ActualPointLeft (Page 2056)

Objet PolyLine (Page 1989)

Objet Polygon (Page 1986)

Objet Line (Page 1958)

Propriété AdaptBorder

Description

TRUE si le cadre de champ doit être adapté dynamiquement à la taille du texte. BOOLEAN Accès en écriture et en lecture.

Remarque

Lors de la modification dynamique du contenu du champ, le pompage du champ peut se produire.

"AdaptBorder = False" permet d'obtenir de meilleures performances dans Runtime.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, la taille du texte est adaptée dynamiquement à la taille du champ.

```
Sub IOFieldConfiguration()
'VBA372
Dim objIOField As HMIIOField
'
'Add new IO-Field to active document:
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
```

```
With objIOField
.AdaptBorder = True
End With
End Sub
```

Voir aussi

Objet OptionGroup (Page 1977)
Objet TextList (Page 2025)
Objet StaticText (Page 2017)
Objet IOField (Page 1947)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)

Propriété AdaptPicture

Description

TRUE si la taille de la vue est adaptée à la taille de la fenêtre de vue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA373
Dim objPicWindow As HMIPictureWindow
'
'Add new picturewindow into active document:
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = ""
.TagPrefix = "Struct."
```

```
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété AdaptSize

Description

TRUE si la taille de la fenêtre de vue est adaptée à la taille de la vue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA374
Dim objPicWindow As HMIPictureWindow
'
'Add new picturewindow into active document:
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété AlarmHigh

Description

Détermine le seuil supérieur auquel l'alarme se déclenche ou le restitue.

Déterminez avec la propriété TypeAlarmHigh" le type d'exploitation (en pour cent ou absolu).

La propriété "CheckAlarmHigh" détermine si la surveillance de ce seuil est activée.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "50".

```
Sub BarGraphLimitConfiguration()
'VBA375
Dim objBarGraph As HMIBarGraph
'
'Add new BarGraph to active document:
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolut
.TypeAlarmHigh = False
'Activate monitoring
.CheckAlarmHigh = True
'Set barcolor to "yellow"
.ColorAlarmHigh = RGB(255, 255, 0)
'set upper limit to "50"
.AlarmHigh = 50
End With
End Sub
```

Voir aussi

Propriété TypeAlarmHigh (Page 2379)

Propriété ColorAlarmHigh (Page 2126)

Propriété CheckAlarmHigh (Page 2115)

Objet BarGraph (Page 1881)

Propriété AlarmLow

Description

Détermine le seuil inférieur auquel l'alarme se déclenche ou le restitue.

Déterminez avec la propriété "TypeAlarmLow" le type d'exploitation (en pour cent ou absolu).

La propriété "CheckAlarmLow" détermine si la surveillance de ce seuil est activée.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit être déclenchée à la valeur "10".

```
Sub BarGraphLimitConfiguration()
'VBA376
Dim objBarGraph As HMIBarGraph
'
'Add new BarGraph to active document:
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolut
.TypeAlarmLow = False
'Activate monitoring
.CheckAlarmLow = True
'Set Barcolor to "yellow"
.ColorAlarmLow = RGB(255, 255, 0)
'set lower limit to "10"
.AlarmLow = 10
End With
End Sub
```

Voir aussi

Propriété TypeAlarmLow (Page 2379)

Propriété ColorAlarmLow (Page 2127)

Propriété CheckAlarmLow (Page 2115)

Objet BarGraph (Page 1881)

Propriété Alignment

Description

Définit ou renvoie l'échelle (à gauche/à droite ou en haut/en bas) selon la position de l'objet BarGraph. La propriété Scaling doit être définie sur TRUE pour afficher l'échelle.

Effets	Valeur affectée
A droite ou en bas	True
A gauche ou en haut	False

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans ce exemple, l'échelle doit se trouver à droite du bargraphe :

```
Sub BarGraphConfiguration()
'VBA377
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Alignment = True
.Scaling = True
End With
End Sub
```

Voir aussi

Propriété Scaling (Page 2340)

Propriété Direction (Page 2151)

Objet BarGraph (Page 1881)

Propriété AlignmentLeft

Description

Détermine l'alignement horizontal du texte ou le restitue. Plage des valeurs de 0 à 2.

Alignement horizontal	Valeur affectée
gauche	0
Centré	1
droite	2

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le texte du champ d'E/S est centré horizontalement :

```
Sub IOFieldConfiguration()
'VBA378
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.AlignmentLeft = 1
End With
End Sub
```

Voir aussi

Voir aussi

Propriété AlignmentTop (Page 2064)

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet GroupDisplay (Page 1935)

Objet IOField (Page 1947)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété AlignmentTop

Description

Détermine l'alignement vertical du texte ou le restitue. Plage des valeurs de 0 à 2.

Alignement horizontal	Valeur affectée
haut	0
Centré	1
bas	2

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le texte du champ d'E/S est centré :

```
Sub IOFieldConfiguration()
'VBA379
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.AlignmentLeft = 1
.AlignmentTop = 1
End With
End Sub
```

Voir aussi

Propriété AlignmentLeft (Page 2063)

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet GroupDisplay (Page 1935)

Objet IOField (Page 1947)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété AnalogResultInfos

Description

Renvoie le listage AnalogResultInfos. Utilisez la propriété AnalogResultInfos pour définir les plages de valeurs et les valeurs des propriétés dans le Dialogue Dynamique.

Exemple

Un exemple pour l'application de la propriété AnalogResultInfos se trouve dans cette documentation, à la rubrique "Objet AnalogResultInfos (listage)".

Voir aussi

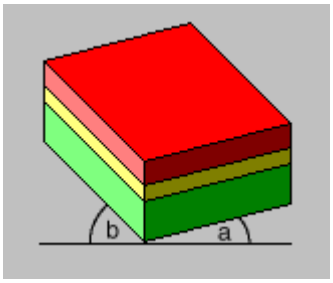
Objet DynamicDialog (Page 1912)

Objet AnalogResultInfos (liste) (Page 1875)

Propriété AngleAlpha

Description

Détermine l'angle de profondeur a pour l'effet tridimensionnel de l'objet "3DBarGraph" ou le restitue. Plage des valeurs en degrés de 0 à 90.



Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, les valeurs "15" et "45" ont été affectées aux angles de profondeur a et b :

```
Sub HMI3DBarGraphConfiguration()
'VBA380
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
'Depth-angle a = 15 degrees
.AngleAlpha = 15
'Depth-angle b = 45 degrees
.AngleBeta = 45
End With
End Sub
```

Voir aussi

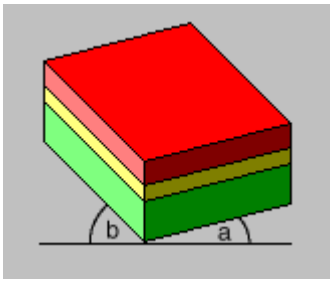
Propriété AngleBeta (Page 2067)

Objet 3DBarGraph (Page 1867)

Propriété AngleBeta

Description

Détermine l'angle de profondeur b pour l'effet tridimensionnel de l'objet "3DBarGraph" ou le restitue. Plage des valeurs en degrés de 0 à 90.



Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, les valeurs "15" et "45" ont été affectées aux angles de profondeur a et b :

```
Sub HMI3DBarGraphConfiguration()
'VBA381
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
'Depth-angle a = 15 degrees
.AngleAlpha = 15
'Depth-angle b = 45 degrees
.AngleBeta = 45
End With
End Sub
```

Voir aussi

Propriété AngleAlpha (Page 2066)

Objet 3DBarGraph (Page 1867)

Propriété Application

Description

Restitue l'application Graphics-Designer lorsque la propriété Application est utilisée sans indicateur d'objet. Lorsque la propriété Application est utilisée avec indicateur d'objet, elle restitue un objet Application qui représente l'application avec laquelle l'objet défini a été créé. Accès en lecture.

Exemple

Dans cet exemple, un objet Excel est créé et le nom de l'application sorti :

```
Sub CreateExcelApplication()
'VBA382
'
'Open Excel invisible
Dim objExcelApp As New Excel.Application
MsgBox objExcelApp
'Delete the reference to Excel and close it
Set objExcelApp = Nothing
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété ApplicationDataPath

Description

Renvoie le chemin complet de la vue active dans Graphics Designer. Accès uniquement en lecture.

Exemple

La procédure "ShowApplicationDataPath()" permet de sortir le chemin de la vue actuelle :

```
Sub ShowApplicationDataPath()
'VBA383
MsgBox Application.ApplicationDataPath
End Sub
```

Voir aussi

Propriété Application (Page 2067)
Objet Application (Page 1876)

Propriété Assignments

Description

Une liste qui contient les correspondances entre la valeur d'affichage et le texte à afficher réellement.

Les correspondances dépendent du type de liste configuré. Déterminez le type de liste avec la propriété ListType.

Le nombre d'inscriptions dépend de la longueur totale de la chaîne de caractères qui est transmise à la propriété "Assignments". Cette chaîne de caractères ne doit pas dépasser 500.000 octets. Pour vérifier, il est possible d'utiliser la fonction LenB() avant d'émettre l'accès à la propriété "Assignments".

Exemple

--

Voir aussi

Propriété ListType (Page 2260)

Objet TextList (Page 2025)

Propriété AssumeOnExit

Description

VRAI si le texte entré est validé après la sortie du champ de saisie (par ex. avec la touche <Tab> ou un clic de souris). BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le texte entré est validé à la sortie du champ d'E/S.

```
Sub IOFieldConfiguration()
'VBA385
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.AssumeOnExit = True
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet IOField (Page 1947)

Propriété AssumeOnFull

Description

TRUE, lorsque le champ de saisie est automatiquement quitté après saisie complète (le nombre de caractères prédéterminé a été entré) et que la saisie est validée. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le texte entré est validé à la sortie du champ d'E/S.

```
Sub IOFieldConfiguration()
'VBA386
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.AssumeOnFull = True
End With
End Sub
```

Voir aussi

Propriété OutputFormat (Page 2301)

Propriété DataFormat (Page 2149)

Objet IOField (Page 1947)

Propriété AutomationName

Description

Définit ou renvoie le nom d'une propriété selon le type d'objet source et cible sélectionnés de la liaison directe.

Les deux tableaux vous indiquent quand utiliser la propriété AutomationName. Un "--" signifie qu'une chaîne vide (" ") est affectée à la propriété lors de la création de l'objet DirectConnection.

Type d'objet source (propriété SourceLink)

Propriété Type	Propriété AutomationName	Propriété ObjectName
hmiSourceTypeConstant	--	Nom des constantes (par ex. nom de la vue)
hmiSourceTypeProperty	Propriété de l'objet source (par ex. "Top")	Nom de l'objet source (par ex. "Rectangle_A")
hmiSourceTypePropertyOfThisObject	--	--
hmiSourceTypeVariableDirect	--	Nom de variable
hmiSourceTypeVariableIndirect	--	Nom de variable

Type d'objet cible (propriété DestinationLink)

Propriété Type	Propriété AutomationName	Propriété ObjectName
hmiDestTypeProperty	Propriété de l'objet cible (par ex. "Left")	Nom de l'objet cible (par ex. "Rectangle_A")
hmiDestTypePropertyOfThisObject	--	--
hmiDestTypePropertyOfActualWindow	Propriété de l'objet cible (par ex. "Left")	--
hmiDestTypeVariableDirect	--	Nom de variable
hmiDestTypeVariableIndirect	--	Nom de variable
hmiDestTypeDirectMessage	--	Nom de variable
hmiDestTypeIndirectMessage	--	Nom de variable

Exemple

Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub SetZoomAndScrollPositionInActiveView()
'VBA387
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDynConnection As HMIDirectConnection
'
'Add objects to active document:
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
'
'to position and configure objects:
With objRectangleA
.Top = 100
.Left = 100
End With
```

```
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Text = "SetPosition"
End With
'
'Directconnection is initiate by mouseclick:
Set objDynConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDynConnection
'Sourceobject: Top-Property of Rectangle_A
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Targetobject: Left-Property of Rectangle_B
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
End Sub
```

Voir aussi

Propriété DestinationLink (Page 2150)

Propriété Type (Page 2378)

Propriété SourceLink (Page 2356)

Propriété ObjectName (Page 2293)

Objet SourceLink (Page 2016)

Objet DestLink (Page 1904)

Propriété AvailableDataLanguages

Description

Renvoie un listage des langues de configuration disponibles.

Exemple

La procédure "AusgabeDataLanguages()" permet de sortir toutes les langues de configuration avec leurs codes (sous forme de valeur décimale) :

```
Sub OutputDataLanguages()
'VBA388
Dim colDataLang As HMIDataLanguages
Dim objDataLang As HMIDataLanguage
Dim strLangList As String
Dim iCounter As Integer
'
'Save collection of datalanguages
'into variable "colDataLang"
Set colDataLang = Application.AvailableDataLanguages
iCounter = 1
'
'Get every languagename and the assigned ID
For Each objDataLang In colDataLang
With objDataLang
If 0 = iCounter Mod 3 Or 1 = iCounter Then
 strLangList = strLangList & vbCrLf & .LanguageID & " " & .LanguageName
    Else
 strLangList = strLangList & " / " & .LanguageID & " " & .LanguageName
    End If
End With
iCounter = iCounter + 1
Next objDataLang
MsgBox strLangList
End Sub
```

Voir aussi

Propriété LanguageName (Page 2220)

Propriété LanguageID (Page 2219)

Pour affecter des textes d'aide à des menus et à des barres d'outils (Page 1633)

Pour créer des menus multilingues (Page 1626)

Configuration multilingue avec VBA (Page 1617)

Propriété Average

Description

TRUE, lorsqu'une moyenne est constituée à partir des 10 dernières valeurs. Pour constituer une nouvelle moyenne, il faut modifier une valeur. La moyenne est réinitialisée lors d'un changement de vue. S'il n'y a par exemple qu'une seule valeur disponible après le changement de vue, la moyenne suivante est constituée : (5+0+0+0+0+0+0+0+0+0)/10=0,5.

Accès en écriture et en lecture BOOLEEN.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la formation de moyenne doit être activée :

```
Sub BarGraphConfiguration()
'VBA389
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Average = True
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété Axe

Description

Définit ou renvoie l'axe pour la représentation de la valeur de mesure. Plage des valeurs de 0 à 2.

Axe	Valeur affectée
X	0
Y	1
Z	2

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, l'axe Y est défini pour la représentation de la valeur de mesure :

```
Sub HMI3DBarGraphConfiguration()
'VBA390
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Axe = 1
End With
End Sub
```

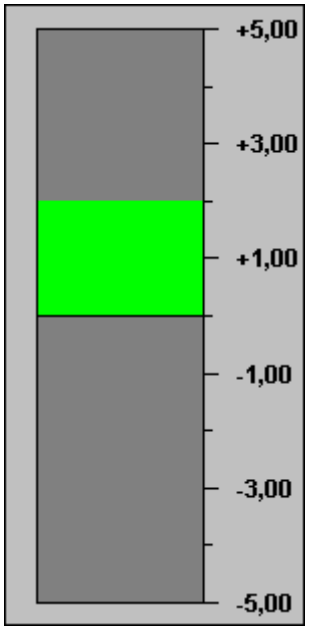
Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété AxisSection

Description

Définit la distance entre deux longues sections axiales ou la restitue. La distance est alors spécifiée en unités de graduation et dépend des valeurs maximales et minimales configurées.



Objet BarGraph (valeur minimale/valeur maximale : -5/5; AxisSection = 2)

Exemple

La procédure "BarGraphConfiguration()" permet d'accéder aux propriétés de l'objet BarGraph. Dans cet exemple, la section d'axe est définie sur "2" :

```
Sub BarGraphConfiguration()
'VBA391
Dim objBar As HMIBarGraph
Set objBar = ActiveDocument.HMIObjects.AddHMIObject("Bar1",
"HMIBarGraph")
With objBar
.AxisSection = 2
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

B

Propriété BackBorderWidth

Description

Détermine la largeur du cadre tridimensionnel en Pixel ou la restitue. La valeur pour la largeur dépend de la taille de l'objet.

Slider

Définit ou renvoie la largeur du bord en pixels. BackBorderWidth = 0 élimine l'affichage du bord pour l'Défileur.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la largeur du cadre 3D est définie sur "2" :

```
Sub ButtonConfiguration()
'VBA392
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.BackBorderWidth = 2
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Objet RoundButton (Page 2003)

Objet GroupDisplay (Page 1935)

Objet Button (Page 1886)

Propriété BackColor

Description

Détermine la couleur d'arrière-plan de l'objet ou la restitue. Accès en lecture/écriture LONG.

La couleur d'arrière-plan ne s'affiche pas si "Transparent" est paramétré comme motif de remplissage.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur d'arrière-plan est définie sur "Jaune" :

```
Sub RectangleConfiguration()
'VBA393
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BackColor = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet EllipseSegment (Page 1920)

Objet StaticText (Page 2017)

Objet Slider (Page 2013)

Objet TextList (Page 2025)

Objet RoundRectangle (Page 2006)

Objet RoundButton (Page 2003)

Objet Rectangle (Page 2000)

Objet Polygon (Page 1986)

Objet PieSegment (Page 1983)

Objet OptionGroup (Page 1977)

Objet GroupDisplay (Page 1935)

Objet GraphicObject (Page 1931)

Objet IOField (Page 1947)

Objet Ellipse (Page 1914)

Objet Document (Page 1908)

Objet Circle (Page 1890)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Objet 3DBarGraph (Page 1867)

Propriété BackColor2

Description

Définit la couleur de bargraphe pour l'affichage de la valeur actuelle ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la couleur de bargraphe est définie sur "Jaune" pour l'affichage de la valeur actuelle :

```
Sub BarGraphConfiguration()
'VBA394
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.BackColor2 = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété BackColor3

Description

Définit la couleur de l'arrière-plan de bargraphe ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la couleur d'arrière-plan du bargraphe est définie sur "Bleu" :

```
Sub BarGraphConfiguration()
'VBA395
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.BackColor3 = RGB(0, 0, 255)
End With
```

```
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété BackColorBottom

Description

Définit la couleur pour la partie inférieure/droite du Slider ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, la partie inférieure du défileur est définie sur "Bleu" :

```
Sub SliderConfiguration()
'VBA396
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.BackColorBottom = RGB(0, 0, 255)
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Propriété BackColorTop

Description

Définit la couleur pour la partie supérieure/gauche du Slider ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, la partie supérieure du défileur est définie sur "Jaune" :

```
Sub SliderConfiguration()
'VBA397
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.BackColorTop = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Propriété BackFlashColorOff

Description

Définit la couleur de l'arrière-plan d'objet pour l'état clignotant "Arrêt" ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur de l'état clignotant "Désactivé" est définie sur "Jaune" :

```
Sub RectangleConfiguration()
'VBA398
Dim objRectangle As HMIRectangle
```

```
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BackFlashColorOff = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Objet StaticText (Page 2017)

Objet Slider (Page 2013)

Objet TextList (Page 2025)

Objet RoundRectangle (Page 2006)

Objet RoundButton (Page 2003)

Objet Rectangle (Page 2000)

Objet Polygon (Page 1986)

Objet PieSegment (Page 1983)

Objet OptionGroup (Page 1977)

Objet GraphicObject (Page 1931)

Objet IOField (Page 1947)

Objet EllipseSegment (Page 1920)

Objet Ellipse (Page 1914)

Objet Circle (Page 1890)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété BackFlashColorOn

Description

Définit la couleur de l'arrière-plan d'objet pour l'état clignotant "Marche" ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur de l'état clignotant "Activé" est définie sur "Bleu" :

```
Sub RectangleConfiguration()
'VBA399
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BackFlashColorOn = RGB(0, 0, 255)
End With
End Suba
```

Voir aussi

Objet RoundButton (Page 2003)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet TextList (Page 2025)
Objet RoundRectangle (Page 2006)
Objet Rectangle (Page 2000)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet GraphicObject (Page 1931)
Objet IOField (Page 1947)
Objet EllipseSegment (Page 1920)
Objet Ellipse (Page 1914)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)
Objet BarGraph (Page 1881)

Propriété Background

Description

TRUE, lorsque l'arrière-plan d'un objet bargraphe tridimensionnel doit être visible. BOOLEAN
Accès en écriture et en lecture.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, l'arrière-plan est défini sur "Transparent" :

```
Sub HMI3DBarGraphConfiguration()
'VBA400
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Background = False
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété BackPictureAlignment

Description

Définit comme attribut "Mode de représentation" la position et la mise à l'échelle du papier peint de la vue de process.

normal	Le papier peint est représenté de manière centrée dans sa taille d'origine. Il demeure en place lorsque la vue est étirée en runtime.
étiré (fenêtre)	Le papier peint est mis à l'échelle de la plus grande des deux fenêtres, fenêtre Runtime et vue de process. En Runtime, il est mis à l'échelle de la fenêtre Runtime et la mise à l'échelle continue pendant l'étirement.
en mosaïque	Le Graphics Designer et la vue du process sont dimensionnés avec la vue en taille originale.
étirée (vue)	Le papier peint est mis à l'échelle de la taille configurée pour la vue de process. Il conserve sa taille lorsque la vue est étirée en runtime.

Propriété BackPictureName

Description

Définit le chemin et le nom du fichier qui est utilisé comme papier peint dans la vue de process, ou renvoie le chemin et le nom du fichier.

Les fichiers dans le format EMF, WMF, DB, BMP, GIF, JPG, JPEG ou ICO sont appropriés.

Quand aucun chemin n'est indiqué, le fichier est recherché dans le sous-répertoire \GraCS. Si vous entrez un autre chemin, une copie est créée dans le répertoire \GraCS.

Indication du chemin

Les formats d'indication du chemin suivants sont possibles :

- absolue : par ex. "C:\Siemens\WinCC\Icons\myIcon.ICO.
- relative : le répertoire initial pour l'indication relative du chemin est le répertoire "GraCS" du projet actuel.
- <global> : désigne le chemin d'installation de WinCC. L'indication du chemin "<global> \Icons\myIcon" correspond à l'indication de chemin donnée sous "absolu".
- <project> : Désigne le répertoire actuel du projet.

Propriété BarDepth

Description

Détermine la profondeur de bargraphe en Pixel ou la restitue.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la profondeur du bargraphe est définie sur "40" :

```
Sub HMI3DBarGraphConfiguration()
'VBA401
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.BarDepth = 40
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété BarHeight

Description

Détermine la hauteur de bargraphe en Pixel ou la restitue.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la hauteur du bargraphe est définie sur "60" :

```
Sub HMI3DBarGraphConfiguration()
'VBA402
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.BarHeight = 60
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété BarWidth

Description

Détermine la largeur de bargraphe en Pixel ou la restitue.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la largeur du bargraphe est définie sur "80" :

```
Sub HMI3DBarGraphConfiguration()
'VBA403
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.BarWidth = 80
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété BasePicReferenced

Description

TRUE, lorsque la vue assignée doit être enregistrée dans l'objet "Indicateur d'état". Faute de quoi, seul le renvoi à l'objet respectif est enregistré. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la vue affectée à l'objet Indicateur d'état doit être enregistrée :

```
Sub StatusDisplayConfiguration()
'VBA404
Dim objStatDisp As HMIStatusDisplay
Set objStatDisp = ActiveDocument.HMIObjects.AddHMIObject("Statusdisplay1",
"HMIStatusDisplay")
With objStatDisp
.BasePicReferenced = True
End With
End Sub
```

Voir aussi

Objet StatusDisplay (Page 2020)

Propriété BasePicTransColor

Description

Détermine quelle couleur de l'objet Bitmap assigné (.bmp, .dib) doit être configurée sur "transparent" ou restitue la couleur. LONG Accès en écriture et en lecture.

La couleur n'est définie sur "Transparent" que si la propriété "BasePicUseTransColor" a la valeur "True".

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la couleur "Jaune" est définie sur Transparent :

```
Sub StatusDisplayConfiguration()
'VBA405
Dim objStatDisp As HMIStatusDisplay
Set objStatDisp = ActiveDocument.HMIObjects.AddHMIObject("Statusdisplay1",
"HMIStatusDisplay")
With objStatDisp
.BasePicTransColor = RGB(255, 255, 0)
.BasePicUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété BasePicUseTransColor (Page 2089)

Objet StatusDisplay (Page 2020)

Propriété BasePicture

Description

Définit ou renvoie la vue de base pour l'objet Indicateur d'état.

La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

La propriété "BasePicReferenced" détermine dans ce contexte si la vue de base doit être enregistrée ou référencée en même temps que l'objet "Indicateur d'état".

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la vue "Testbild.BMP" est utilisée comme vue de base :

```
Sub StatusDisplayConfiguration()
'VBA406
Dim objStatDisp As HMIStatusDisplay
Set objStatDisp = ActiveDocument.HMIObjects.AddHMIObject("Statusdisplay1",
"HMIStatusDisplay")
With objStatDisp
'
'To use this example copy a Bitmap-Graphic
'to the "GraCS"-Folder of the actual project.
'Replace the picturename "Testpicture.BMP" with the name of
'the picture you copied
```

```
.BasePicture = "Testpicture.BMP"
End With
End Sub
```

Voir aussi

Propriété BasePicReferenced (Page 2087)

Objet StatusDisplay (Page 2020)

Propriété BasePicUseTransColor

Description

TRUE, lorsque la couleur configurée (propriété "BasePicTransColor") pour l'objet Bitmap doit être positionnée sur "transparent". BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la couleur "Jaune" est définie sur Transparent :

```
Sub StatusDisplayConfiguration()
'VBA407
Dim objStatDisp As HMIStatusDisplay
Set objStatDisp = ActiveDocument.HMIObjects.AddHMIObject("Statusdisplay1",
"HMIStatusDisplay")
With objStatDisp
.BasePicTransColor = RGB(255, 255, 0)
.BasePicUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété BasePicTransColor (Page 2087)

Objet StatusDisplay (Page 2020)

Propriété BaseX

Description

Définit ou renvoie pour l'objet 3DBarGraph la distance horizontale en pixels entre le bord droit du bargraphe et le bord gauche du champ de l'objet.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la distance horizontale est définie sur "80" :

```
Sub HMI3DBarGraphConfiguration()
'VBA408
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.BaseX = 80
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété BaseY

Description

Définit ou renvoie pour l'objet 3DBarGraph la distance verticale en pixels entre le bord inférieur du bargraphe et le bord supérieur du champ de l'objet.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la distance verticale est définie sur "100" :

```
Sub HMI3DBarGraphConfiguration()
'VBA409
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.BaseY = 100
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété BinaryResultInfo

Description

Renvoie l'objet BinaryResultInfo.

Exemple

Un exemple d'application de la propriété BinaryResultInfo se trouve dans cette documentation, à la rubrique "Objet BinaryResultInfo".

Voir aussi

Objet BinaryResultInfo (Page 1884)

Propriété BitNotSetValue

Description

Définit ou renvoie la valeur pour la propriété dynamisée si le bit indiqué d'une variable configurée n'est pas défini.

Définissez le bit devant être défini pour le déclenchement de la modification de valeur à l'aide de la propriété BitNumber.

Exemple

Dans l'exemple qui suit, on dynamise le rayon d'un cercle à l'aide du dialogue Dynamique, on attribue un nom de variable, on définit le bit à configurer et on affecte la valeur de propriété respective aux états "activé"/"non activé" :

```
Sub AddDynamicDialogToCircleRadiusTypeBit()
'VBA410
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_B", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBit
.Trigger.VariableTriggers(1).CycleType = hmiVariableCycleType_5s
.BitResultInfo.BitNumber = 1
.BitResultInfo.BitSetValue = 40
.BitResultInfo.BitNotSetValue = 80
End With
End Sub
```

Voir aussi

Propriété BitNumber (Page 2092)

Objet BitResultInfo (Page 1885)

Propriété BitNumber

Description

Détermine le bit dont l'état doit être modifié pour déclencher une modification de valeur ou le restitue. La variable utilisée doit être de type BYTE, WORD ou DWORD.

Exemple

Dans l'exemple qui suit, on dynamise le rayon d'un cercle à l'aide du dialogue Dynamique, on attribue un nom de variable, on définit le bit à configurer et on affecte la valeur de propriété respective aux états "activé"/"non activé" :

```
Sub AddDynamicDialogToCircleRadiusTypeBit()
'VBA411
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_B", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBit
.BitResultInfo.BitNumber = 1
.BitResultInfo.BitSetValue = 40
.BitResultInfo.BitNotSetValue = 80
End With
End Sub
```

Voir aussi

Objet BitResultInfo (Page 1885)

Référence VBA (Page 1724)

Propriété BitResultInfo

Description

Renvoie l'objet BitResultInfo.

Exemple

Un exemple d'application de la propriété BitResultInfo se trouve dans cette documentation, à la rubrique "Objet BitResultInfo".

Voir aussi

Objet BitResultInfo (Page 1885)

Propriété BitSetValue

Description

Définit ou renvoie la valeur pour la propriété dynamisée si le bit indiqué d'une variable configurée est pas défini.

Définissez le bit devant être défini pour le déclenchement de la modification de valeur à l'aide de la propriété BitNumber.

Exemple

Dans l'exemple qui suit, on dynamise le rayon d'un cercle à l'aide du dialogue Dynamique, on attribue un nom de variable, on définit le bit à configurer et on affecte la valeur de propriété respective aux états "activé"/"non activé" :

```
Sub AddDynamicDialogToCircleRadiusTypeBit()
'VBA412
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_B", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBit
.BitResultInfo.BitNumber = 1
.BitResultInfo.BitSetValue = 40
.BitResultInfo.BitNotSetValue = 80
End With
End Sub
```

Voir aussi

Propriété BitNumber (Page 2092)

Objet BitResultInfo (Page 1885)

Propriété Bold

Description

TRUE si l'attribut de police "Gras" est défini dans l'objet pour le texte dépendant des langues. BOOLEAN Accès en écriture et en lecture.

Exemple

Remarque

Pour que cet exemple s'applique, les langues concernées doivent avoir été configurées.

Dans l'exemple suivant, on configure les attributs de texte d'un bouton pour le français et pour l'anglais :

```
Sub ExampleForLanguageFonts()
'VBA413
Dim colLangFonts As HMILanguageFonts
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
objButton.Text = "Displaytext"
Set colLangFonts = objButton.LDFonts
'Set french fontproperties:
With colLangFonts.ItemByLCID(1036)
.Family = "Courier New"
.Bold = True
.Italic = False
.Underlined = True
.Size = 12
End With
'Set english fontproperties:
With colLangFonts.ItemByLCID(1033)
.Family = "Times New Roman"
.Bold = False
.Italic = True
.Underlined = False
.Size = 14
End With
End Sub
```

Voir aussi

Propriété Underlined (Page 2388)
Propriété Size (Page 2353)
Propriété Parent (Page 2303)
Propriété LanguageID (Page 2219)
Propriété Italic (Page 2212)
Propriété FontFamily (Page 2189)
Propriété Application (Page 2067)
Objet LanguageFont (Page 1950)

Propriété BorderBackColor

Description

Détermine la couleur d'arrière-plan du trait pour l'objet ou la restitue. LONG Accès en écriture et en lecture.

La couleur d'arrière-plan n'est visible que si la propriété BorderStyle > 0.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur d'arrière-plan du trait est définie sur "Jaune" :

```
Sub RectangleConfiguration()
'VBA415
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BorderBackColor = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet PieSegment (Page 1983)
Propriété BorderStyle (Page 2103)
Objet TextList (Page 2025)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet IOField (Page 1947)
Objet GraphicObject (Page 1931)
Objet EllipseArc (Page 1917)
Objet EllipseSegment (Page 1920)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)
Objet BarGraph (Page 1881)

Propriété BorderColor

Description

Détermine la couleur du trait pour l'objet ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur de trait est définie sur "Bleu" :

```
Sub RectangleConfiguration()
'VBA416
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BorderColor = RGB(0, 0, 255)
End With
End Sub
```

Voir aussi

Objet GraphicObject (Page 1931)
Objet TextList (Page 2025)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet IOField (Page 1947)
Objet EllipseArc (Page 1917)
Objet EllipseSegment (Page 1920)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)
Objet BarGraph (Page 1881)

Propriété BorderColorBottom

Description

Définit ou renvoie la couleur de la partie droite et inférieure du cadre 3D. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la couleur du cadre 3D est définie :

```
Sub ButtonConfiguration()
'VBA417
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.BorderColorBottom = RGB(255, 0, 0)
.BorderColorTop = RGB(0, 0, 255)
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Objet Button (Page 1886)

Propriété BorderColorTop

Description

Définit ou renvoie la couleur de la partie gauche et supérieure du cadre 3D. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la couleur du cadre 3D est définie :

```
Sub ButtonConfiguration()
'VBA418
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.BorderColorBottom = RGB(255, 0, 0)
.BorderColorTop = RGB(0, 0, 255)
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Objet Button (Page 1886)

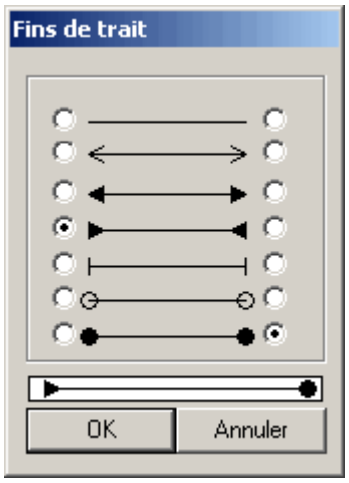
Propriété BorderEndStyle

Description

Détermine le type des fins de trait de l'objet ou le restitue. Accès en lecture/écriture LONG.

Détermination des fins de trait

Vous déterminez le type des fins de trait à l'aide d'une valeur hexadécimale à cinq positions que vous convertissez en valeur décimale.



A partir de la fenêtre "Fins de trait", déterminez les fins de trait de l'objet de la manière suivante :

- Colonne gauche : configure le début du trait. Plage de valeurs (du haut vers le bas) de 0 à 6. Le début de trait correspond au premier chiffre en valeur hexadécimale. Dans la configuration représentée, la première position a la valeur "3".
- Colonne droite : configure le fin de trait. Plage de valeurs (du haut vers le bas) de 0 à 6. La fin de trait correspond au cinquième chiffre en valeur hexadécimale. Dans la configuration représentée, la cinquième position a la valeur "6".

On obtient alors une valeur hexadécimale de "60003". Celle-ci correspond à une valeur décimale de "393219", que vous affectez à la propriété BorderEndStyle.

Exemple

La procédure "LineConfiguration()" permet d'accéder aux propriétés du trait. Dans cet exemple, le type de fin de trait est défini sur la configuration représentée ci-dessus :

```
Sub LineConfiguration()
'VBA419
Dim objLine As HMILine
Set objLine = ActiveDocument.HMIObjects.AddHMIObject("Line1", "HMILine")
With objLine
.BorderEndStyle = 393219
End With
End Sub
```

Voir aussi

Objet PolyLine (Page 1989)

Objet Line (Page 1958)

Propriété BorderFlashColorOff

Description

Définit la couleur du trait de l'objet pour l'état clignotant "Arrêt" ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur pour l'état clignotant "Désactivé" est définie sur "Noir" :

```
Sub RectangleConfiguration()
'VBA420
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BorderFlashColorOff = RGB(0, 0, 0)
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet TextList (Page 2025)
Objet RoundRectangle (Page 2006)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet GraphicObject (Page 1931)
Objet IOField (Page 1947)
Objet EllipseSegment (Page 1920)
Objet EllipseArc (Page 1917)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)

Propriété BorderFlashColorOn

Description

Définit la couleur du trait de l'objet pour l'état clignotant "Marche" ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur de l'état clignotant "Activé" est définie sur "Rouge" :

```
Sub RectangleConfiguration()
'VBA421
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BorderFlashColorOn = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet StaticText (Page 2017)
Objet StatusDisplay (Page 2020)
Objet Slider (Page 2013)
Objet TextList (Page 2025)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet GraphicObject (Page 1931)
Objet IOField (Page 1947)
Objet EllipseSegment (Page 1920)
Objet EllipseArc (Page 1917)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)

Propriété BorderStyle

Description

Détermine le type de trait pour l'objet ou le restitue. Plage des valeurs de 0 à 4.

Style de trait	Valeur affectée
———	0
— —	1
- - - - - -	2
— - —	3
—- - —	4

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, le style de trait est défini sur "1" :

```
Sub RectangleConfiguration()
'VBA422
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.BorderStyle = 1
End With
End Sub
```

Voir aussi

Objet IOField (Page 1947)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet Polygon (Page 1986)
Objet PolyLine (Page 1989)
Objet TextList (Page 2025)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet GraphicObject (Page 1931)
Objet EllipseSegment (Page 1920)
Objet EllipseArc (Page 1917)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)
Objet BarGraph (Page 1881)
Objet 3DBarGraph (Page 1867)

Propriété BorderWidth

Description

Détermine l'épaisseur du trait (en Pixel) pour l'objet ou la restitue.

Exemple

Dans l'exemple suivant, la largeur de ligne d'un cercle nouvellement inséré est définie sur "2".

```
Sub CircleConfiguration()
'VBA423
Dim objCircle As IHMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1", "HMICircle")
With objCircle
.BorderWidth = 2
End With
End Sub
```

Voir aussi

Objet IOField (Page 1947)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet TextList (Page 2025)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet GraphicObject (Page 1931)
Objet EllipseSegment (Page 1920)
Objet EllipseArc (Page 1917)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)

Propriété BottomConnectedObjectName

Description

Renvoie le nom de l'objet de démarrage qui relie le connecteur. Accès en lecture.

Exemple :

Un exemple pour l'application de la propriété BottomConnectedObjectName se trouve dans cette documentation, à la rubrique "Objet objConnection".

Voir aussi

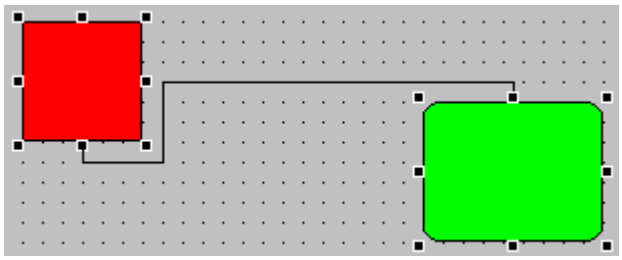
Objet objConnection (Page 1973)

Propriété BottomConnectedConnectionPointIndex

Description

Renvoie le point de raccordement de l'objet auquel est relié le connecteur.

Point de raccordement	Valeur affectée
haut	0
droite	1
bas	2
gauche	3



Exemple

Un exemple pour l'application de la propriété BottomConnectedObjectName se trouve dans cette documentation, à la rubrique "Objet objConnection".

Voir aussi

Objet objConnection (Page 1973)

Propriété BoxAlignment

Description

TRUE, lorsque les champs sont alignés à droite. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "CreateOptionGroup()" permet de créer l'objet OptionGroup avec quatre champs d'option. A chaque champ d'option est attribué le nom standard "myCustomText<Numéro>" :

```
Sub CreateOptionGroup()
'VBA424
Dim objRadioBox As HMIOptionGroup
Dim iCounter As Integer
Set objRadioBox = ActiveDocument.HMIObjects.AddHMIObject("RadioBox_1", "HMIOptionGroup")
```

```
iCounter = 1
With objRadioBox
.Height = 100
.Width = 180
.BoxCount = 4
.BoxAlignment = False
For iCounter = 1 To .BoxCount
.index = iCounter
.Text = "CustomText" & .index
Next iCounter
End With
End Sub
```

Voir aussi

Propriété BoxCount (Page 2108)

Objet OptionGroup (Page 1977)

Objet CheckBox (Page 1889)

Propriété BoxCount

Description

Détermine le nombre de champs ou le restitue. Plage des valeurs de 1 à 32.

Exemple

La procédure "CreateOptionGroup()" permet de créer l'objet OptionGroup avec quatre champs d'option. A chaque champ d'option est attribué le nom standard "myCustomText<Numéro>" :

```
Sub CreateOptionGroup()
'VBA425
Dim objRadioBox As HMIOptionGroup
Dim iCounter As Integer
Set objRadioBox = ActiveDocument.HMIObjects.AddHMIObject("RadioBox_1", "HMIOptionGroup")
iCounter = 1
With objRadioBox
.Height = 100
.Width = 180
.BoxCount = 4
.BoxAlignment = True
For iCounter = 1 To .BoxCount
.index = iCounter
.Text = "CustomText" & .index
Next iCounter
End With
End Sub
```

Voir aussi

Propriété BoxAlignment (Page 2107)
Objet OptionGroup (Page 1977)
Objet CheckBox (Page 1889)

Propriété BoxType

Description

Détermine le type de champ ou le restitue. Plage des valeurs de 0 à 2.

Type de champ	Valeur affectée
Sortie	0
Entrée	1
Champ d'E/S	2

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le type de champ "Entrée" est configuré :

```
Sub IOFieldConfiguration()
'VBA426
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.BoxType = 1
End With
End Sub
```

Voir aussi

Objet IOField (Page 1947)

Propriété Button1Width

Description

Définit ou renvoie pour l'objet Signalisation groupée la largeur du bouton 1 en pixels.

Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la largeur du bouton 1 est définie sur "50" :

```
Sub GroupDisplayConfiguration()
'VBA427
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.Button1Width = 50
End With
End Sub
```

Voir aussi

Propriété SameSize (Page 2338)

Propriété Button4Width (Page 2111)

Propriété Button3Width (Page 2111)

Propriété Button2Width (Page 2110)

Objet GroupDisplay (Page 1935)

Propriété Button2Width

Description

Définit ou renvoie pour l'objet Signalisation groupée la largeur du bouton 2 en pixels.

Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la largeur du bouton 2 est définie sur "50" :

```
Sub GroupDisplayConfiguration()
'VBA428
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.Button2Width = 50
End With
End Sub
```

Voir aussi

Propriété SameSize (Page 2338)
Propriété Button4Width (Page 2111)
Propriété Button3Width (Page 2111)
Propriété Button1Width (Page 2109)
Objet GroupDisplay (Page 1935)

Propriété Button3Width

Description

Définit ou renvoie pour l'objet Signalisation groupée la largeur du bouton 3 en pixels.

Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la largeur du bouton 3 est définie sur "50" :

```
Sub GroupDisplayConfiguration()
'VBA429
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.Button3Width = 50
End With
End Sub
```

Voir aussi

Propriété SameSize (Page 2338)
Propriété Button4Width (Page 2111)
Propriété Button2Width (Page 2110)
Propriété Button1Width (Page 2109)
Objet GroupDisplay (Page 1935)

Propriété Button4Width

Description

Définit ou renvoie pour l'objet Signalisation groupée la largeur du bouton 4 en pixels.

Si la propriété SameSize est positionnée sur TRUE, les Buttons auront tous la même largeur.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la largeur du bouton 4 est définie sur "50" :

```
Sub GroupDisplayConfiguration()
'VBA430
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.Button4Width = 50
End With
End Sub
```

Voir aussi

Propriété Button1Width (Page 2109)

Propriété SameSize (Page 2338)

Propriété Button3Width (Page 2111)

Propriété Button2Width (Page 2110)

Objet GroupDisplay (Page 1935)

Propriété ButtonColor

Description

Définit ou renvoie pour l'objet Slider la couleur du curseur. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, la couleur du curseur est définie sur "Jaune" :

```
Sub SliderConfiguration()
```

```
'VBA431
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.ButtonColor = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

C

Propriété Caption

Description

TRUE, lorsque la fenêtre de vue ou d'application a une barre de titre dans Runtime. BOOLEAN Accès en écriture et en lecture.

La propriété Caption doit être définie sur True lorsque la fenêtre de vue ou d'application doit posséder les boutons Maximiser et Fermer.

Exemple

La procédure "ApplicationWindowConfig" permet d'accéder aux propriétés de la fenêtre d'application. Dans cet exemple, la fenêtre d'application est configurée:

```
Sub ApplicationWindowConfig()
'VBA432
Dim objAppWindow As HMIApplicationWindow
Set objAppWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow",
"HMIApplicationWindow")
With objAppWindow
.Caption = True
.CloseButton = False
.Height = 200
.Left = 10
.MaximizeButton = True
.Moveable = False
.OnTop = True
.Sizeable = True
.Top = 20
.Visible = True
.Width = 250
.WindowBorder = True
End With
```

```
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)
Objet ApplicationWindow (Page 1879)

Propriété CaptionText

Description

Définit ou renvoie pour l'objet PictureWindow le nom de la fenêtre s'affichant dans Runtime.

Pour cela, la propriété Caption doit être définie sur "True".

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA433
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété CheckAlarmHigh

Description

TRUE si le seuil "Alarm high" est surveillé pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés AlarmHigh, ColorAlarmHigh et TypeAlarmHigh.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "50".

```
Sub BarGraphLimitConfiguration()
'VBA434
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeAlarmHigh = False
'Activate monitoring
.CheckAlarmHigh = True
'Set barcolor to "yellow"
.ColorAlarmHigh = RGB(255, 255, 0)
'Set upper limit to "50"
.AlarmHigh = 50
End With
End Sub
```

Voir aussi

Propriété TypeAlarmHigh (Page 2379)

Propriété ColorAlarmHigh (Page 2126)

Propriété AlarmHigh (Page 2061)

Objet BarGraph (Page 1881)

Propriété CheckAlarmLow

Description

TRUE si le seuil "Alarm low" est surveillé pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés AlarmLow, ColorAlarmLow et TypeAlarmLow.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit être déclenchée à la valeur "10".

```
Sub BarGraphLimitConfiguration()
'VBA435
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeAlarmLow = False
'Activate monitoring
.CheckAlarmLow = True
'Set barcolor to "yellow"
.ColorAlarmLow = RGB(255, 255, 0)
'Set lower limit to "10"
.AlarmLow = 10
End With
End Sub
```

Voir aussi

Propriété ColorAlarmLow (Page 2127)

Propriété TypeAlarmLow (Page 2379)

Propriété AlarmLow (Page 2062)

Objet BarGraph (Page 1881)

Propriété Checked

Description

TRUE si une coche doit apparaître devant une option de menu définie par l'utilisateur. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Supprimer objets" et d'ajouter deux options de menu ("Supprimer rectangle" et "Supprimer cercle"). En outre, une coche est affectée à la première option de menu :

```
Sub CreateMenuItem()
'VBA436
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
'
```

```
'Add new menu "Delete objects" to menubar:
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete objects")
'
'Add two menuitems to the new menu
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
Rectangles")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete Circles")
With objMenu.MenuItems
.Item("DeleteAllRectangles").Checked = True
End With
End Sub
```

Voir aussi

Propriété MenuItems (Page 2284)

Configuration des menus et des barres d'outils (Page 1619)

Propriété CheckLimitHigh4

Description

TRUE si le seuil supérieur "Réserve 4" de l'objet BarGraph doit être surveillé. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés LimitHigh4, ColorLimitHigh4 et TypeLimitHigh4.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "70".

```
Sub BarGraphLimitConfiguration()
'VBA437
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitHigh4 = False
'Activate monitoring
.CheckLimitHigh4 = True
'set barcolor to "red"
.ColorLimitHigh4 = RGB(255, 0, 0)
'Set upper limit to "70"
.LimitHigh4 = 70
End With
End Sub
```

Voir aussi

Propriété TypeLimitHigh4 (Page 2380)

Propriété LimitHigh4 (Page 2255)

Propriété ColorLimitHigh4 (Page 2129)

Objet BarGraph (Page 1881)

Propriété CheckLimitHigh5

Description

TRUE si le seuil supérieur "Réserve 5" de l'objet BarGraph doit être surveillé. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés LimitHigh5, ColorLimitHigh5 et TypeLimitHigh5.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "80".

```
Sub BarGraphLimitConfiguration()
'VBA438
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitHigh5 = False
'Activate monitoring
.CheckLimitHigh5 = True
'set barcolor to "black"
.ColorLimitHigh5 = RGB(0, 0, 0)
'Set upper limit to "80"
.LimitHigh5 = 80
End With
End Sub
```

Voir aussi

Propriété ColorLimitHigh5 (Page 2130)

Propriété TypeLimitHigh5 (Page 2381)

Propriété LimitHigh4 (Page 2255)

Objet BarGraph (Page 1881)

Propriété CheckLimitLow4

Description

TRUE si le seuil inférieur "Réserve 4" de l'objet BarGraph doit être surveillé. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés LimitLow4, ColorLimitLow4 et TypeLimitLow4.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "5".

```
Sub BarGraphLimitConfiguration()
'VBA439
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitLow4 = False
'Activate monitoring
.CheckLimitLow4 = True
'Set barcolor to "green"
.ColorLimitLow4 = RGB(0, 255, 0)
'set lower limit to "5"
.LimitLow4 = 5
End With
End Sub
```

Voir aussi

Propriété TypeLimitLow4 (Page 2382)

Propriété LimitLow4 (Page 2257)

Propriété ColorLimitLow4 (Page 2131)

Objet BarGraph (Page 1881)

Propriété CheckLimitLow5

Description

TRUE si le seuil inférieur "Réserve 5" de l'objet BarGraph doit être surveillé. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés LimitLow5, ColorLimitLow5 et TypeLimitLow5.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "0".

```
Sub BarGraphLimitConfiguration()
'VBA440
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitLow5 = False
'Activate monitoring
.CheckLimitLow5 = True
'Set barcolor to "white"
.ColorLimitLow5 = RGB(255, 255, 255)
'set lower limit to "0"
.LimitLow5 = 0
End With
End Sub
```

Voir aussi

Propriété TypeLimitLow5 (Page 2383)

Propriété LimitLow5 (Page 2257)

Propriété ColorLimitLow5 (Page 2132)

Objet BarGraph (Page 1881)

Propriété CheckToleranceHigh

Description

TRUE si le seuil "Tolerance high" est surveillé pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés ToleranceHigh, ColorToleranceHigh et TypeToleranceHigh.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "45".

```
Sub BarGraphLimitConfiguration()
'VBA441
```

```
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeToleranceHigh = False
'Activate monitoring
.CheckToleranceHigh = True
'Set barcolor to "yellow"
.ColorToleranceHigh = RGB(255, 255, 0)
'Set upper limit to "45"
.ToleranceHigh = 45
End With
End Sub
```

Voir aussi

Propriété TypeToleranceHigh (Page 2383)

Propriété ToleranceHigh (Page 2370)

Propriété ColorToleranceHigh (Page 2133)

Objet BarGraph (Page 1881)

Propriété CheckToleranceLow

Description

TRUE si le seuil "Tolerance low" est surveillé pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés ToleranceLow, ColorToleranceLow et TypeToleranceLow.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "15".

```
Sub BarGraphLimitConfiguration()
'VBA442
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeToleranceLow = False
'Activate monitoring
.CheckToleranceLow = True
'Set barcolor to "yellow"
.ColorToleranceLow = RGB(255, 255, 0)
```

```
'Set lower limit to "15"
.ToleranceLow = 15
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété TypeToleranceLow (Page 2384)

Propriété ToleranceLow (Page 2371)

Propriété ColorToleranceLow (Page 2134)

Propriété CheckWarningHigh

Description

TRUE si le seuil "Warning high" est surveillé pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés WarningHigh, ColorWarningHigh et TypeWarningHigh.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "75".

```
Sub BarGraphLimitConfiguration()
'VBA443
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeWarningHigh = False
'Activate monitoring
.CheckWarningHigh = True
'Set barcolor to "red"
.ColorWarningHigh = RGB(255, 0, 0)
'Set upper limit to "75"
.WarningHigh = 75
End With
End Sub
```

Voir aussi

Propriété WarningHigh (Page 2470)

Propriété TypeWarningHigh (Page 2385)

Propriété ColorWarningHigh (Page 2135)

Objet BarGraph (Page 1881)

Propriété CheckWarningLow

Description

TRUE si le seuil "Warning low" est surveillé pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil, la représentation lorsque ce dernier est atteint et le type d'analyse sont définis à l'aide des propriétés WarningLow, ColorWarningLow et TypeWarningLow.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "12".

```
Sub BarGraphLimitConfiguration()
'VBA444
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeWarningLow = False
'Activate monitoring
.CheckWarningLow = True
'Set barcolor to "magenta"
.ColorWarningLow = RGB(255, 0, 255)
'Set lower limit to "12"
.WarningLow = 12
End With
End Sub
```

Voir aussi

Propriété WarningLow (Page 2471)

Propriété TypeWarningLow (Page 2386)

Propriété ColorWarningLow (Page 2136)

Objet BarGraph (Page 1881)

Propriété ClearOnError

Description

TRUE si la valeur entrée dans le champ d'E/S est automatiquement effacée en cas d'entrée erronée. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le champ d'E/S doit être effacé en cas d'entrée erronée :

```
Sub IOFieldConfiguration()
'VBA445
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.ClearOnError = True
End With
End Sub
```

Voir aussi

Objet IOField (Page 1947)

Propriété ClearOnNew

Description

TRUE si la valeur entrée dans le champ d'E/S est effacée dans la mesure où le champ d'E/S contient le focus. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, la valeur entrée dans le champ d'E/S est effacée dans la mesure où il contient le focus.

```
Sub IOFieldConfiguration()
'VBA446
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.ClearOnNew = True
End With
End Sub
```

Voir aussi

Objet IOField (Page 1947)

Propriété CloseButton

Description

TRUE si les objets ApplicationWindow et PictureWindow possèdent un bouton "Fermer" dans Runtime. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ApplicationWindowConfig" permet d'accéder aux propriétés de la fenêtre d'application. Dans cet exemple, la fenêtre d'application doit posséder un bouton "Fermer" dans Runtime :

```
Sub ApplicationWindowConfig()
'VBA447
Dim objAppWindow As HMIApplicationWindow
Set objAppWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow1",
"HMIApplicationWindow")
With objAppWindow
.CloseButton = True
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Objet ApplicationWindow (Page 1879)

Propriété CollectValue

Description

Contient en Runtime comme valeur d'entrée l'état actuel des classes d'alarmes actives. LONG Accès en écriture et en lecture.

La dynamisation via une variable permet par exemple de déterminer la valeur à partir de la signalisation groupée de vues inférieures du point de vue hiérarchique.

Exemple

--

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété ColorAlarmHigh

Description

Définit ou renvoie la couleur de bargraphe pour le seuil "Alarm high". LONG Accès en écriture et en lecture.

La propriété "CheckAlarmHigh" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "50" et la couleur de bargraphe doit devenir "Rouge".

```
Sub BarGraphLimitConfiguration()
'VBA449
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeAlarmHigh = False
'Activate monitoring
.CheckAlarmHigh = True
'Set barcolor to "red"
.ColorAlarmHigh = RGB(255, 0, 0)
'Set upper limit to "50"
.AlarmHigh = 50
End With
End Sub
```

Voir aussi

Propriété CheckAlarmHigh (Page 2115)

Objet BarGraph (Page 1881)

Propriété ColorAlarmLow

Description

Définit ou renvoie la couleur de bargraphe pour le seuil "Alarm low". LONG Accès en écriture et en lecture.

La propriété "CheckAlarmLow" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "10" et la couleur de bargraphe doit devenir "Rouge".

```
Sub BarGraphLimitConfiguration()
'VBA450
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeAlarmLow = False
'Activate monitoring
.CheckAlarmLow = True
'Set barcolor to "red"
.ColorAlarmLow = RGB(255, 0, 0)
'Set lower limit to "10"
.AlarmLow = 10
End With
End Sub
```

Voir aussi

Propriété CheckAlarmLow (Page 2115)

Objet BarGraph (Page 1881)

Propriété ColorBottom

Description

Définit la couleur pour la butée inférieure/droite de l'objet Slider ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, la couleur de la butée inférieure/droite est définie sur "Rouge" :

```
Sub SliderConfiguration()
'VBA451
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.ColorBottom = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Propriété ColorChangeType

Description

TRUE lorsqu'en cas de changement de couleur sur l'objet BarGraph (par ex. lorsqu'un seuil est atteint), celui-ci doit se produire par segment. Avec le paramètre FALSE, le changement de couleur est valable pour la totalité du bargraphe. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, le changement de couleur doit se produire pour tout le bargraphe :

```
Sub BarGraphLimitConfiguration()
'VBA452
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.ColorChangeType = False
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété ColorLimitHigh4

Description

Définit la couleur pour le seuil supérieur "Reserve 4" ou la restitue. LONG Accès en écriture et en lecture.

La propriété "CheckLimitHigh4" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "70" et la couleur de bargraphe doit devenir "Rouge".

```
Sub BarGraphLimitConfiguration()
'VBA453
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
```

```
'Set analysis to absolute
.TypeLimitHigh4 = False
'Activate monitoring
.CheckLimitHigh4 = True
'Set barcolor to "red"
.ColorLimitHigh4 = RGB(255, 0, 0)
'Set upper limit to "70"
.LimitHigh4 = 70
End With
End Sub
```

Voir aussi

Propriété CheckLimitHigh4 (Page 2117)

Objet BarGraph (Page 1881)

Propriété ColorLimitHigh5

Description

Définit la couleur pour le seuil supérieur "Reserve 5" ou la restitue. LONG Accès en écriture et en lecture.

La propriété "CheckLimitHigh5" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "80" et la couleur de bargraphe doit devenir "Noir".

```
Sub BarGraphLimitConfiguration()
'VBA454
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitHigh5 = False
'Activate monitoring
.CheckLimitHigh5 = True
'Set barcolor to "black"
```

```
.ColorLimitHigh5 = RGB(0, 0, 0)
'Set upper limit to "80"
.LimitHigh5 = 80
End With
End Sub
```

Voir aussi

Propriété CheckLimitHigh5 (Page 2118)

Objet BarGraph (Page 1881)

Propriété ColorLimitLow4

Description

Définit la couleur pour le seuil inférieur "Reserve 4" ou la restitue. LONG Accès en écriture et en lecture.

La propriété "CheckLimitLow4" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "5" et la couleur de bargraphe doit devenir "Vert".

```
Sub BarGraphLimitConfiguration()
'VBA455
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitLow4 = False
'Activate monitoring
.CheckLimitLow4 = True
'Set barcolor to "green"
.ColorLimitLow4 = RGB(0, 255, 0)
'Set lower limit to "5"
.LimitLow4 = 5
End With
```

```
End Sub
```

Voir aussi

Propriété CheckLimitLow4 (Page 2119)

Objet BarGraph (Page 1881)

Propriété ColorLimitLow5

Description

Définit la couleur pour le seuil inférieur "Reserve 5" ou la restitue. LONG Accès en écriture et en lecture.

La propriété "CheckLimitLow5" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "0" et la couleur de bargraphe doit devenir "Blanc".

```
Sub BarGraphLimitConfiguration()
'VBA456
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitLow5 = False
'Activate monitoring
.CheckLimitLow5 = True
'Set barcolor to "white"
.ColorLimitLow5 = RGB(255, 255, 255)
'Set lower limit to "0"
.LimitLow5 = 0
End With
End Sub
```

Voir aussi

Propriété CheckLimitLow5 (Page 2119)

Objet BarGraph (Page 1881)

Propriété ColorToleranceHigh

Description

Définit ou renvoie la couleur pour le seuil supérieur "Tolerance High". LONG Accès en écriture et en lecture.

La propriété "CheckToleranceHigh" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "45" et la couleur de bargraphe doit devenir "Jaune".

```
Sub BarGraphLimitConfiguration()
'VBA457
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeToleranceHigh = False
'Activate monitoring
.CheckToleranceHigh = True
'Set barcolor to "yellow"
.ColorToleranceHigh = RGB(255, 255, 0)
'Set upper limit to "45"
.ToleranceHigh = 45
End With
End Sub
```

Voir aussi

Propriété CheckToleranceHigh (Page 2120)

Objet BarGraph (Page 1881)

Propriété ColorToleranceLow

Description

Définit ou renvoie la couleur pour le seuil inférieur "Tolerance Low". LONG Accès en écriture et en lecture.

La propriété "CheckToleranceLow" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "12" et la couleur de bargraphe doit devenir "Jaune".

```
Sub BarGraphLimitConfiguration()
'VBA458
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeToleranceLow = False
'Activate monitoring
.CheckToleranceLow = True
'Set barcolor to "yellow"
.ColorToleranceLow = RGB(255, 255, 0)
'Set lower limit to "15"
.ToleranceLow = 15
End With
End Sub
```

Voir aussi

Propriété CheckToleranceLow (Page 2121)

Objet BarGraph (Page 1881)

Propriété ColorTop

Description

Définit la couleur pour la butée supérieure/gauche de l'objet Slider ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, la couleur de la butée supérieure/gauche est définie sur "Orange" :

```
Sub SliderConfiguration()
'VBA459
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.ColorTop = RGB(255, 128, 0)
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Propriété ColorWarningHigh

Description

Définit ou renvoie la couleur pour le seuil supérieur "Warning high". LONG Accès en écriture et en lecture.

La propriété "CheckWarningHigh" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "75" et la couleur de bargraphe doit devenir "Rouge" :

```
Sub BarGraphLimitConfiguration()
'VBA460
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeWarningHigh = False
'Activate monitoring
.CheckWarningHigh = True
'Set barcolor to "red"
.ColorWarningHigh = RGB(255, 0, 0)
'Set upper limit to "75"
.WarningHigh = 75
End With
End Sub
```

Voir aussi

Propriété CheckWarningHigh (Page 2122)

Objet BarGraph (Page 1881)

Propriété ColorWarningLow

Description

Définit ou renvoie la couleur pour le seuil inférieur "Warning low". LONG Accès en écriture et en lecture.

La propriété "CheckWarningLow" doit avoir la valeur TRUE s'il faut que la couleur du bargraphe change au cas où le seuil serait atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à une valeur de "12" et la couleur de bargraphe doit devenir "Magenta".

```
Sub BarGraphLimitConfiguration()
'VBA461
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeWarningLow = False
'Activate monitoring
.CheckWarningLow = True
'Set barcolor to "magenta"
.ColorWarningLow = RGB(255, 0, 255)
'Set lower limit to "12"
.WarningLow = 12
End With
End Sub
```

Voir aussi

Propriété CheckWarningLow (Page 2123)

Objet BarGraph (Page 1881)

Propriété CommonVBSCode

Description

Définit pour la vue active la zone de déclaration commune subordonnée des actions ou bien la fournit en retour.

Dans l'éditeur d'action du Graphics Designer, vous configurez des actions sur des événements et des propriétés. Dans la zone de déclaration des actions, vous pouvez déclarer des variables communes et également créer des fonctions et procédures pour une vue de process. Si la vue est activée, chaque action VBS peut accéder au Runtime à ces variables, fonctions et procédures.

Si vous activez "CommonVBSCode", la chaîne de caractères est copiée dans la zone de déclaration "Evénement" et "Propriété" dans l'éditeur d'action. Un code y existant éventuellement déjà est alors écrasé. Activez donc toujours "CommonVBSCode" avant d'activer les zones de déclaration subordonnées avec "CommonVBSEventArea" ou "CommonVBSPropertyArea".

Exemple

Dans l'exemple ci-dessous, une variable commune à tous les objets de vue est déclarée dans la vue active. La zone de déclaration commune est ensuite fournie en retour :

```
Sub DefineTagInActiveDocument
ActiveDocument.CommonVBSCode = "DIM actionIsdone" & vbCrLf
MsgBox ActiveDocument.CommonVBSCode
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété CommonVBSEventArea

Description

Définit pour la vue active la zone de déclaration "Evénement" des actions ou bien la fournit en retour.

Dans l'éditeur d'action du Graphics Designer, vous configurez des actions par exemple sur des événements. Dans la zone de déclaration "Evénement", vous pouvez déclarer des variables communes et également créer des fonctions et procédures pour une vue de process. Si la vue est activée, chaque action VBS ayant été configurée pour un événement peut accéder au Runtime à ces variables, fonctions et procédures.

Si vous activez "CommonVBSEventArea", la chaîne de caractères est copiée dans la zone de déclaration "Evénement" dans l'éditeur d'action. Un code y existant éventuellement déjà est alors écrasé. Pour cette raison, lisez d'abord le code défini par exemple avec "CommonVBSCode" avant d'activer la zone de déclaration avec "CommonVBSEventArea".

Exemple

Dans l'exemple qui suit, deux variables sont déclarées dans la vue active. La zone de déclaration "Evénement" est ensuite fournie en retour :

```
Sub DefineTagInActiveDocument
ActiveDocument.CommonVBSCode = "DIM actionIsdone" & vbCrLf
ActiveDocument.CommonVBSEventArea = ActiveDocument.CommonVBSEventArea & "DIM
"eventHasOccurred"
MsgBox ActiveDocument.CommonVBSEventArea
End Sub
```

Propriété CommonVBSPropertyArea

Description

Définit pour la vue active la zone de déclaration "Propriété" des actions ou bien la fournit en retour.

Dans l'éditeur d'action de Graphics Designer, vous configurez des actions par exemple sur des propriétés. Dans la zone de déclaration "Propriété", vous pouvez à ce but déclarer des variables communes et également créer des fonctions et procédures pour une vue de process. Si la vue est activée, chaque action VBS ayant été configurée pour une propriété peut accéder au Runtime à ces variables, fonctions et procédures.

Si vous activez "CommonVBSPropertyArea", la chaîne de caractères est copiée dans la zone de déclaration "Propriété" dans l'éditeur d'action. Un code y existant éventuellement déjà est alors écrasé. Pour cette raison, lisez d'abord le code défini par exemple avec "CommonVBSCode" avant d'activer la zone de déclaration avec "CommonVBSPropertyArea".

Exemple

Dans l'exemple qui suit, deux variables sont déclarées dans la vue active. La zone de déclaration "Propriété" est ensuite fournie en retour :

```
Sub DefineTagInActiveDocument
ActiveDocument.CommonVBSCode = "DIM actionIsdone" & vbCrLf
ActiveDocument. CommonVBSPropertyArea = ActiveDocument.CommonVBSPropertyArea & "DIM
propertyIsChanged"
MsgBox ActiveDocument.CommonVBSPropertyArea
End Sub
```

Propriété CommandLine

Description

Renvoie les paramètres de démarrage sous forme de chaîne de caractères lorsque l'application est ouverte avec Exécuter > Démarrage "Grafexe.exe". Accès en lecture.

Exemple

Dans cet exemple, une alarme avec le paramètre de démarrage est sortie à l'ouverture du document.

```
Sub Document_Opened(CancelForwarding As Boolean)
'VBA462
MsgBox Application.Commandline
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété ActualPointTop

Description

TRUE si le code source d'un script en C ou VB a pu être compilé avec succès. Accès en lecture BOOLEAN.

Exemple

Dans cet exemple, on insère un bouton et un cercle dans la vue active. Dans Runtime, le rayon du cercle doit être agrandi à chaque clic sur le bouton. On utilise pour cela un script VB :

```
Sub IncreaseCircleRadiusWithVBScript()
'VBA463
Dim objButton As HMIButton
Dim objCircleA As HMICircle
Dim objEvent As HMIEvent
Dim objVBScript As HMIScriptInfo
Dim strCode As String
strCode = "Dim objCircle" & vbCrLf & "Set objCircle = "
strCode = strCode & "hmiRuntime.ActiveScreen.ScreenItems(""CircleVB"")"
strCode = strCode & vbCrLf & "objCircle.Radius = objCircle.Radius + 5"
Set objCircleA = ActiveDocument.HMIObjects.AddHMIObject("CircleVB", "HMICircle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircleA
.Top = 100
.Left = 100
End With
With objButton
.Top = 10
.Left = 10
.Width = 200
.Text = "Increase Radius"
End With
'On every mouseclick the radius will be increased:
Set objEvent = objButton.Events(1)
Set objVBScript = objButton.Events(1).Actions.AddAction(hmiActionCreationTypeVBScript)
objVBScript.SourceCode = strCode
Select Case objVBScript.Compiled
Case True
MsgBox "Compilation OK!"
Case False
MsgBox "Errors by compilation!"
End Select
End Sub
```

Voir aussi

Propriété SourceCode (Page 2357)

Objet ScriptInfo (Page 2009)

Propriété ConfigurationFileName

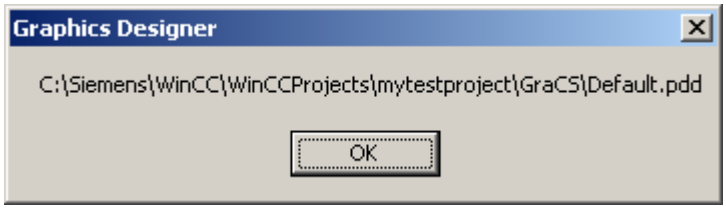
Description

Renvoie le nom de fichier et le chemin complet du fichier de configuration du projet ouvert. Accès en lecture STRING.

Exemple

La procédure "ShowConfigurationFileName()" permet de sortir le chemin du fichier de configuration de la vue actuelle :

```
Sub ShowConfigurationFileName()
'VBA464
MsgBox ActiveDocument.Application.ConfigurationFileName
End Sub
```



Voir aussi

Propriété Application (Page 2067)

Objet Application (Page 1876)

Propriété Count

Description

Renvoie le nombre d'éléments dans le listage indiqué. Accès en lecture Long.

Exemple

Dans l'exemple suivant, une vue nouvelle est créée, dans laquelle sont insérés quelques objets. Le nombre des objets insérés est sorti à la fin :

```
Sub ObjectsInActiveDocument()
'VBA465
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objDocument As Document
Set objDocument = Application.Documents.Add(hmiOpenDocumentTypeVisible)
Dim iIndex As Integer
iIndex = 1
For iIndex = 1 To 5
Set objCircle = objDocument.HMIObjects.AddHMIObject("Circle" & iIndex, "HMICircle")
Set objRectangle = objDocument.HMIObjects.AddHMIObject("Rectangle" & iIndex,
"HMIRectangle")
With objCircle
.Top = (10 * iIndex)
.Left = (10 * iIndex)
End With
With objRectangle
.Top = ((10 * iIndex) + 50)
.Left = (10 * iIndex)
End With
Next iIndex
MsgBox "There are " & objDocument.HMIObjects.Count & " objects in the document"
End Sub
```

Voir aussi

Objet VariableTriggers (liste) (Page 2049)
Objet Views (liste) (Page 2052)
Objet VariableStateValues (liste) (Page 2046)
Objet ToolbarItems (liste) (Page 2034)
Objet Toolbars (liste) (Page 2029)
Objet SymbolLibraries (liste) (Page 2024)
Objet Selection (liste) (Page 2010)
Objet Properties (liste) (Page 1992)
Objets HMIObjects (liste) (Page 1945)
Objet MenuItems (liste) (Page 1970)
Objet Menus (liste) (Page 1965)
Objet Layers (Liste) (Page 1956)
Objet LanguageTexts (liste) (Page 1954)
Objet LanguageFonts (liste) (Page 1951)
Objet GroupedObjects (liste) (Page 1938)
Objet FolderItems (liste) (Page 1929)
Objet Events (liste) (Page 1924)
Objet Documents (liste) (Page 1911)
Objet HMIDefaultObjects (liste) (Page 1939)
Objet DataLanguages (liste) (Page 1903)
Objet ConnectionPoints (liste) (Page 1898)
Objet AnalogResultInfos (liste) (Page 1875)
Objet Actions (liste) (Page 1872)

Propriété CurrentDataLanguage

Description

Définit la langue de configuration ou renvoie le code langue sous forme de valeur décimale. Accès en lecture/écriture LONG.

Exemple

La procédure "ShowDataLanguage()" permet de sortir la langue de configuration actuellement définie :

```
Sub ShowDataLanguage()
'VBA466
MsgBox Application.CurrentDataLanguage
End Sub
```

Voir aussi

Propriété Application (Page 2067)

Evénement DataLanguageChanged (Page 1741)

Configuration multilingue avec VBA (Page 1617)

Propriété CurrentDesktopLanguage

Description

Renvoie le code langue de la langue d'interface actuellement définie sous forme de valeur décimale. Accès en lecture LONG.

Exemple

La procédure "ShowDesktopLanguage()" permet de sortir la langue d'interface actuellement définie :

```
Sub ShowDesktopLanguage()
'VBA467
MsgBox Application.CurrentDesktopLanguage
End Sub
```

Voir aussi

Propriété Application (Page 2067)

Objet Application (Page 1876)

Evénement DesktopLanguageChanged (Page 1742)

Configuration multilingue avec VBA (Page 1617)

Propriété CursorControl

Description

TRUE, lorsqu'en mode curseur alpha ce dernier, après avoir quitté le champ, rejoint le champ suivant selon la séquence TAB. BOOLEAN Accès en écriture et en lecture.

Pour cela, la propriété CursorMode doit être définie sur TRUE.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le curseur doit passer au champ suivant après la sortie du champ. Pour cela, la propriété Cursormode doit être définie sur TRUE au préalable :

```
Sub IOFieldConfiguration()
'VBA468
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
Application.ActiveDocument.CursorMode = True
With objIOField
.CursorControl = True
End With
End Sub
```

Voir aussi

Propriété TabOrderAlpha (Page 2363)

Propriété TabOrderSwitch (Page 2362)

Propriété CursorMode (Page 2145)

Propriété ActiveDocument (Page 2055)

Objet TextList (Page 2025)

Objet IOField (Page 1947)

Propriété CursorMode

Description

TRUE si le mode "Curseur alpha" doit être activé. FALSE si le mode "Curseur Maj" doit être activé. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, le mode "Curseur alpha" est activé :

```
Sub ActiveDocumentConfiguration()
'VBA469
Application.ActiveDocument.CursorMode = True
End Sub
```

Voir aussi

Propriété CursorControl (Page 2145)

Propriété ActiveDocument (Page 2055)

Objet Documents (liste) (Page 1911)

Propriété CustomMenus

Description

Renvoie un listage des menus définis par l'utilisateur disponibles.

Exemple

La procédure "ShowCustomMenuInformation()" permet de sortir les valeurs Key et Label de tous les menus définis par l'utilisateur de la vue actuelle :

```
Sub ShowCustomMenuInformation()
'VBA470
Dim strKey As String
Dim strLabel As String
Dim strOutput As String
Dim iIndex As Integer
For iIndex = 1 To ActiveDocument.CustomMenus.Count
strKey = ActiveDocument.CustomMenus(iIndex).Key
strLabel = ActiveDocument.CustomMenus(iIndex).Label
strOutput = strOutput & vbCrLf & "Key: " & strKey & "  Label: " & strLabel
Next iIndex
If 0 = ActiveDocument.CustomMenus.Count Then
strOutput = "There are no custommenus for the document created."
End If
MsgBox strOutput
End Sub
```

Voir aussi

Propriété Application (Page 2067)
Propriété ActiveDocument (Page 2055)
Objet Menu (Page 1964)

Propriété CustomToolbars

Description

Renvoie un listage des barres d'outils définies par l'utilisateur disponibles.

Exemple

La procédure "ShowCustomToolbarInformation()" permet de sortir la valeur Key de toutes les barres d'outils définies par l'utilisateur de la vue actuelle :

```
Sub ShowCustomToolbarInformation()
'VBA471
Dim strKey As String
Dim strOutput As String
Dim iIndex As Integer
For iIndex = 1 To ActiveDocument.CustomToolbars.Count
strKey = ActiveDocument.CustomToolbars(iIndex).Key
strOutput = strOutput & vbCrLf & "Key: " & strKey
Next iIndex
If 0 = ActiveDocument.CustomToolbars.Count Then
strOutput = "There are no toolbars created for this document."
    End If
MsgBox strOutput
End Sub
```

Voir aussi

Propriété Application (Page 2067)
Propriété ActiveDocument (Page 2055)
Objet Toolbar (Page 2028)

Propriété CycleName

Description

Renvoie le nom du déclenchement en fonction d'une variable indiqué. Accès en lecture.

Exemple

--

Voir aussi

Objet VariableTrigger (Page 2048)

Propriété CycleTime

Description

Renvoie le temps de cycle du déclenchement en fonction d'une variable indiqué. Accès en lecture.

Exemple

--

Voir aussi

Objet VariableTrigger (Page 2048)

Propriété CycleType

Description

Définit ou renvoie le type de cycle.

Exemple

La procédure "DynamicToRadiusOfNewCircle(hmiCircle As IHMICircle)" permet de créer une dynamique pour le rayon d'un cercle. Dans cet exemple, le rayon du cercle doit être défini toutes les deux secondes :

```
Sub DynamicToRadiusOfNewCircle()
'VBA474
Dim objCircle As hmiCircle
Dim VariableTrigger As HMIVariableTrigger
Set objCircle = Application.ActiveDocument.HMIObjects.AddHMIObject("Circle1", "HMICircle")
Set VariableTrigger = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVariableDirect,
"NewDynamic1")
VariableTrigger.CycleType = hmiVariableCycleType_2s
End Sub
```

Voir aussi

Objet VariableTrigger (Page 2048)

Dynamisation des propriétés d'une vue ou d'un objet (Page 1682)

D

Propriété DataFormat

Description

Définit ou renvoie le type de données de l'objet IOField. Plage des valeurs de 0 à 3.

Type de données	Valeur affectée
Binaire	0
Décimal	1
Chaîne de caractères	2
Hexadécimal	3

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le type de données du champ d'E/S est défini sur "Décimal" :

```
Sub IOFieldConfiguration()
'VBA475
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.DataFormat = 1
End With
End Sub
```

Voir aussi

Objet IOField (Page 1947)

Propriété DefaultHMIObjects

Description

Renvoie le listage HMIDefaultObjects.

Exemple

La procédure "ShowDefaultObjectNames()" permet de sortir les noms de tous les objets contenus dans le listage HMIDefaultObjects :

```
Sub ShowDefaultObjectNames()
'VBA476
Dim strOutput As String
Dim iIndex As Integer
For iIndex = 1 To Application.DefaultHMIObjects.Count
strOutput = strOutput & vbCrLf & Application.DefaultHMIObjects(iIndex).ObjectName
Next iIndex
MsgBox strOutput
End Sub
```

Voir aussi

Objet HMIDefaultObjects (liste) (Page 1939)

Propriété DestinationLink

Description

Renvoie l'objet Destination. Utilisez la propriété DestinationLink pour configurer l'objet cible dans le cadre d'une liaison directe.

Exemple

On utilise la propriété DestinationLink pour renvoyer un objet DestLink. Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub DirectConnection()
'VBA477
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDirConnection As HMIDirectConnection
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
```

```
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 100
.Text = "SetPosition"
End With
'
'Directconnection is initiated by mouseclick:
Set objDirConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDirConnection
'Sourceobject: Property "Top" of Rectangle_A
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Targetobject: Property "Left" of Rectangle_B
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
End Sub
```

Voir aussi

Propriété AutomationName (Page 2070)

Propriété ObjectName (Page 2293)

Propriété Type (Page 2378)

Objet DirectConnection (Page 1906)

Propriété Direction

Description

Définit ou renvoie l'orientation du bargraphe. BOOLEAN Accès en écriture et en lecture.

Slider

Définit ou renvoie la position de l'objet Slider. BOOLEAN Accès en écriture et en lecture.

Position/Axe du bargraphe	valeur affectée
Vertical/Négatif	TRUE
Horizontal/Positif	FALSE

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, la position de l'objet Slider est définie sur "vertical" :

```
Sub SliderConfiguration()
'VBA478
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.Direction = True
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Objet 3DBarGraph (Page 1867)

Propriété DisableVBAEvents

Description

TRUE si la gestion des événements est désactivée. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "DisableVBAEvents()" permet de désactiver la gestion des événements :

```
Sub DisableVBAEvents()
'VBA479
Application.DisableVBAEvents = False
End Sub
```

Voir aussi

Objet Application (Page 1876)

Gestion des événements (Page 1704)

Propriété DisplayName

Description

Renvoie le nom de l'attribut de la propriété. Accès en lecture STRING.

Ainsi, l'expression MsgBox ActiveDocument.HMIObjects("Cercle_1").Properties("Height").DisplayName renvoie comme résultat la "hauteur".

Exemple

La procédure "ShowAllObjectDisplayNames()" permet de sortir les noms de tous les attributs de la propriété dans une zone de message :

```
Sub ShowAllObjectDisplayNames()
'VBA480
Dim strOutput As String
Dim iIndex1 As Integer
iIndex1 = 1
strOutput = "List of all properties-displaynames from object """ &
Application.DefaultHMIObjects(1).ObjectName & """" & vbCrLf & vbCrLf
For iIndex1 = 1 To Application.DefaultHMIObjects(1).Properties.Count
strOutput = strOutput & Application.DefaultHMIObjects(1).Properties(iIndex1).DisplayName &
" / "
Next iIndex1
MsgBox strOutput
End Sub
```

Voir aussi

Objet Property (Page 1993)

Propriété DisplayOptions

Description

Définit l'assignation des objets "Bouton" ou "Bouton rond" ou retourne la valeur. Plage des valeurs de 0 à 3.

Affectation	Valeur affectée
Graphique ou texte	0
Graphique et texte	1
Texte seulement	2
Graphique seulement	3

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton.

Dans cet exemple, le Button est affecté à "Graphique et texte" :

```
Sub ButtonConfiguration()
'VBA814
Dim objbutton As HMIButton
Set objbutton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objbutton
.DisplayOptions = 1
End With
End Sub
```

Voir aussi

Objet Button (Page 1886)

Propriété DisplayText

Description

Renvoie la valeur de la propriété "Label" ou "TooltipText" des objets suivants (accès en lecture STRING) :

- Objet Menu
- Objet MenuItem
- Objet ToolbarItem

Exemple

La procédure "ShowLabelTexts()" permet de sortir tous les textes du premier menu défini par l'utilisateur dans la vue active :

```
Sub ShowLabelTexts()
'VBA481
Dim objLangText As HMILanguageText
Dim iIndex As Integer
For iIndex = 1 To ActiveDocument.CustomMenus(1).LDLabelTexts.Count
Set objLangText = ActiveDocument.CustomMenus(1).LDLabelTexts(iIndex)
MsgBox objLangText.DisplayName
Next iIndex
End Sub
```

Voir aussi

Propriété TooltipText (Page 2373)

Propriété Label (Page 2218)

Objet ToolbarItem (Page 2031)

Objet LanguageText (Page 1953)

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Propriété Documents

Description

Renvoie le listage Documents contenant toutes les vues ouvertes. L'ordre des vues ouvertes est défini chronologiquement.

Exemple

Dans l'exemple qui suit, le nom de toutes les vues ouvertes est sorti :

```
Sub ShowDocuments()
'VBA482
Dim colDocuments As Documents
Dim objDocument As Document
Dim strOutput As String
Set colDocuments = Application.Documents
strOutput = "List of all opened documents:" & vbCrLf
For Each objDocument In colDocuments
strOutput = strOutput & vbCrLf & objDocument.Name
Next objDocument
MsgBox strOutput
End Sub
```

Voir aussi

Propriété Application (Page 2067)

Objet Application (Page 1876)

Propriété Dynamic

Description

Renvoie la dynamisation d'une propriété.

Exemple

Utilisez la propriété Dynamic par ex. lorsque vous souhaitez renvoyer une dynamique disponible. Dans l'exemple suivant, toutes les dynamisations des propriétés d'objet éventuellement existantes sont sorties dans la vue activée :

```
Sub ShowPropertiesDynamicsofAllObjects()
'VBA483
Dim objObject As HMIObject
Dim colObjects As HMIObjects
Dim colProperties As HMIProperties
Dim objProperty As HMIProperty
Dim strOutput As String
Set colObjects = Application.ActiveDocument.HMIObjects
For Each objObject In colObjects
Set colProperties = objObject.Properties
For Each objProperty In colProperties
If 0 <> objProperty.DynamicStateType Then
  strOutput = strOutput & vbCrLf & objObject.ObjectName & " - " & objProperty.DisplayName
& ": Statetype " & objProperty.Dynamic.DynamicStateType
    End If
Next objProperty
Next objObject
MsgBox strOutput
End Sub
```

Voir aussi

Objet Property (Page 1993)

e

Propriété EditAtOnce

Description

TRUE, lorsque l'entrée peut immédiatement avoir lieu sans aucune autre action lors du saut sur le champ avec la touche <Tab>. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, au passage du champ d'E/S, l'entrée doit s'effectuer immédiatement :

```
Sub IOFieldConfiguration()
'VBA484
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
```

```
With objIOField
.EditAtOnce = True
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet IOField (Page 1947)

Propriété ElseCase

Description

Définit ou renvoie la valeur pour la propriété dynamisée en dehors de la plage de valeurs configurée.

Exemple

Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on crée trois plages de valeurs analogiques :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA485
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.Add 50, 40
.AnalogResultInfos.Add 100, 80
.AnalogResultInfos.ElseCase = 100
End With
End Sub
```

Voir aussi

Objet AnalogResultInfos (liste) (Page 1875)

Objet AnalogResultInfo (Page 1874)

Méthode Add (listage AnalogResultInfos) (Page 1766)

Propriété Enabled

Description

TRUE si le menu, l'option de menu ou l'icône est activé et peut être sélectionné. Cela n'est valable que pour les menus et les barres d'outils définis par l'utilisateur. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Supprimer objets" et d'ajouter deux options de menu ("Supprimer rectangle" et "Supprimer cercle"). Dans cet exemple, la deuxième option du menu défini par l'utilisateur "Supprimer objets" apparaît en grisé et ne peut pas être sélectionnée dans Graphics Designer :

```
Sub DisableMenuItem()
'VBA486
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
'
'Add a new menu "Delete objects"
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete objects")
'
'Add two menuitems to the new menu
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
rectangles")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete circles")
'
'Disable menuitem "Delete circles"
With ActiveDocument.CustomMenus("DeleteObjects").MenuItems("DeleteAllCircles")
.Enabled = False
End With
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Configuration des menus et des barres d'outils (Page 1619)

Propriété EndAngle

Description

Définit ou renvoie pour les objets CircularArc, EllipseArc, EllipseSegment et PieSegment la fin de l'objet. L'indication se fait dans le sens horaire, en degrés, en commençant à 12:00 h.

Exemple

La procédure "PieSegmentConfiguration()" permet d'accéder aux propriétés du segment de cercle. Dans cet exemple, le segment de cercle commence à 40° et finit à 180°:

```
Sub PieSegmentConfiguration()
'VBA487
Dim objPieSegment As HMIPieSegment
Set objPieSegment = ActiveDocument.HMIObjects.AddHMIObject("PieSegment1", "HMIPieSegment")
With objPieSegment
.StartAngle = 40
.EndAngle = 180
End With
End Sub
```

Voir aussi

Propriété StartAngle (Page 2359)

Objet PieSegment (Page 1983)

Objet EllipseSegment (Page 1920)

Objet EllipseArc (Page 1917)

Objet CircularArc (Page 1893)

Propriété Events

Description

Renvoie le listage Events. Utilisez la propriété Events pour définir l'événement devant déclencher une action. Définissez l'événement devant être configuré à l'aide du numéro d'index :

- Pour programmer dans VBA une action liée à une propriété, utilisez la propriété "Events(9)", l'"Index 1" représentant l'événement "en cas de modification".
- Pour programmer dans VBA une action liée à un objet, utilisez la propriété "Events(Index)", l'"Index" représentant l'événement déclencheur (voir tableau) :

Indice	EventType (dépend de l'objet utilisé)
0	hmiEventTypeNotDefined
1	hmiEventTypeMouseClick
2	hmiEventTypeMouseLButtonDown
3	hmiEventTypeMouseLButtonUp
4	hmiEventTypeMouseRButtonDown
5	hmiEventTypeMouseRButtonUp
6	hmiEventTypeKeyboardDown
7	hmiEventTypeKeyboardUp

Indice	EventType (dépend de l'objet utilisé)
8	hmiEventTypeFocusEnter
9	hmiEventTypeObjectChange
10	hmiEventTypeOpenPicture
11	hmiEventTypePictureOpen
12	hmiEventTypePictureClose
13	hmiEventTypeObjectDefined
14	hmiEventTypeFocusEnter
15	hmiEventTypeLastTriggerType
16	hmiEventTypeObjSpecificTriggerStart

Exemple

Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub DirectConnection()
'VBA488
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDirConnection As HMIDirectConnection
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 100
.Text = "SetPosition"
End With
'
'Directconnection is initiated by mouseclick:
Set objDirConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDirConnection
'Sourceobject: Property "Top" of Rectangle_A
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
```

```
'Targetobject: Property "Left" of Rectangle_B
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
End Sub
```

Voir aussi

Objet Events (liste) (Page 1924)

Configuration d'actions commandées par événement avec VBA (Page 1693)

Propriété EventType

Description

Renvoie le type d'événement configuré pour l'objet indiqué.

Index	EventType (dépend de l'objet utilisé)
0	hmiEventTypeNotDefined
1	hmiEventTypeMouseClick
2	hmiEventTypeMouseLButtonDown
3	hmiEventTypeMouseLButtonUp
4	hmiEventTypeMouseRButtonDown
5	hmiEventTypeMouseRButtonUp
6	hmiEventTypeKeyboardDown
7	hmiEventTypeKeyboardUp
8	hmiEventTypeFocusEnter
9	hmiEventTypeObjectChange
10	hmiEventTypeOpenPicture
11	hmiEventTypePictureOpen
12	hmiEventTypePictureClose
13	hmiEventTypeObjectDefined
14	hmiEventTypeFocusEnter
15	hmiEventTypeLastTriggerType
16	hmiEventTypeObjSpecificTriggerStart

Exemple

Utilisez la propriété EventType pour éditer un événement déjà configuré. Dans l'exemple qui suit, l'événement "Clic de souris" est configuré dans un premier temps, puis modifié en "Enfoncé" :

```
Sub AddActionToObjectTypeCScript()
'VBA489
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB", "HMICircle")
'
'C-action is initiated by click on object circle
Set objEvent = objCircle.Events(1)
Set objCScript = objEvent.Actions.AddAction(hmiActionCreationTypeCScript)
MsgBox "the type of the projected event is " & objEvent.EventType
End Sub
```

Voir aussi

Objet Events (liste) (Page 1924)

Configuration d'actions commandées par événement avec VBA (Page 1693)

Propriété Exponent

Description

TRUE si la représentation des nombres pour l'objet BarGraph doit s'effectuer avec des exposants (par ex. "1,00e+000"). BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la représentation des nombres du bargraphe doit s'effectuer avec des exposants :

```
Sub BarGraphConfiguration()
'VBA490
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Exponent = True
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété ExtendedOperation

Description

TRUE si la réglette de l'objet Slider est définie sur la valeur finale correspondante (Valeur minimale/Valeur maximale). Cela s'effectue par clic de souris dans la zone hors du réglage actuel du défileur. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, la propriété ExtendedOperation est définie sur TRUE :

```
Sub SliderConfiguration()
'VBA491
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.ExtendedOperation = True
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Propriété ExtendedZoomingEnable

Description

VRAI si la vue peut être agrandie ou réduite par la roulette de la souris en Runtime pour la vue de process sélectionnée. Appuyez à cet effet sur la touche <Ctrl> tout en tournant la roulette de la souris. Le facteur de zoom s'agrandit si la roulette de la souris est tournée de la paume vers l'extérieur.

BOOLEAN Accès en écriture et en lecture.

Conditions préalables à l'utilisation de la fonction de zoom :

- Pilote de souris Logitech ou Microsoft-Intellimouse
- La roulette de la souris doit être réglée sur "Autoscroll".
- Dans les propriétés de l'ordinateur, onglet "Graphics Runtime", la fonction "Zoom étendu" doit être activée pour toutes les vues du process.

Exemple

La procédure "DocConfiguration()" permet d'accéder aux propriétés d'une vue.

Dans cet exemple, la propriété ExtendedZoomingEnable est définie sur VRAI :

```
Sub DocConfiguration()
'VBA815
Dim objDoc As Document
Set objDoc = ActiveDocument
With objDoc
.ExtendedZoomingEnable = True
End With
End Sub
```

f

Propriété FaceplateType

Description

Définit le type de bloc d'affichage de l'instance de bloc d'affichage et fournit son nom en retour. Le type de bloc d'affichage est "Const" et il ne peut donc être défini qu'une seule fois.

Utilisation

Utilisez la méthode Add pour créer un nouvel objet "Instance de bloc d'affichage" dans la vue. Vous accédez à la propriété FaceplateType par le biais de "Properties.Item(3)" :

```
Sub FaceplateInstance_and_Properties()
'VBA847
Dim objFaceplateInstance As HMIFaceplateObject

Set objFaceplateInstance = ActiveDocument.HMIObjects.AddHMIObject("Instance de bloc
d'affichage", "HMIFaceplateObject")
objFaceplateInstance.Properties.Item(3).value = "Faceplate1.fpt"
MsgBox "Faceplate """ & objFaceplateInstance.Properties.Item(3).value & """ is used."

End Sub
```

Propriété FillColor

Description

Détermine la couleur du motif de remplissage pour l'objet ou la restitue. Accès en lecture/écriture LONG.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, la couleur d'arrière-plan est définie sur "Jaune" :

```
Sub RectangleConfiguration()
'VBA493
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.FillColor = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet Button (Page 1886)

Objet StaticText (Page 2017)

Objet Slider (Page 2013)

Objet TextList (Page 2025)

Objet RoundRectangle (Page 2006)

Objet RoundButton (Page 2003)

Objet Rectangle (Page 2000)

Objet Polygon (Page 1986)

Objet PieSegment (Page 1983)

Objet OptionGroup (Page 1977)

Objet GroupDisplay (Page 1935)

Objet GraphicObject (Page 1931)

Objet IOField (Page 1947)

Objet EllipseSegment (Page 1920)

Objet Ellipse (Page 1914)

Objet Document (Page 1908)

Objet Circle (Page 1890)

Objet CheckBox (Page 1889)

Objet BarGraph (Page 1881)

Objet 3DBarGraph (Page 1867)

Propriété Filling

Description

TRUE si l'objet peut être rempli d'une ligne de cadre fermée (par ex. Circle ou Rectangle) (représente aussi par ex. le taux de remplissage d'un réservoir). BOOLEAN Accès en écriture et en lecture.

Définissez le taux de remplissage de l'objet à l'aide de la propriété FillingIndex.

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, le rectangle peut être utilisé comme affichage du taux de remplissage :

```
Sub RectangleConfiguration()
'VBA494
```

```
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.Filling = True
End With
End Sub
```

Voir aussi

Propriété FillingIndex (Page 2167)

Objet StaticText (Page 2017)

Objet Slider (Page 2013)

Objet RoundRectangle (Page 2006)

Objet RoundButton (Page 2003)

Objet Rectangle (Page 2000)

Objet Polygon (Page 1986)

Objet PieSegment (Page 1983)

Objet OptionGroup (Page 1977)

Objet GraphicObject (Page 1931)

Objet EllipseSegment (Page 1920)

Objet Ellipse (Page 1914)

Objet Circle (Page 1890)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété FillingIndex

Description

Définit la valeur en % (par rapport à la hauteur de l'objet) de remplissage de l'objet avec la ligne de cadre fermée (par ex. Circle ou Rectangle).

Le niveau de remplissage est représenté par la couleur d'arrière-plan actuelle. L'arrière-plan non rempli est transparent

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, le rectangle est rempli à 50 % :

```
Sub RectangleConfiguration()
```

```
'VBA495
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.Filling = True
.FillingIndex = 50
End With
End Sub
```

Voir aussi

Objet PieSegment (Page 1983)

Propriété FillColor (Page 2164)

Propriété BackColor (Page 2076)

Objet StaticText (Page 2017)

Objet Slider (Page 2013)

Objet RoundRectangle (Page 2006)

Objet RoundButton (Page 2003)

Objet Rectangle (Page 2000)

Objet Polygon (Page 1986)

Objet OptionGroup (Page 1977)

Objet GraphicObject (Page 1931)

Objet EllipseSegment (Page 1920)

Objet Ellipse (Page 1914)

Objet Circle (Page 1890)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété FillingDirection

Description

0 = L'objet possédant un cadre fermé est rempli de bas en haut.

1 = L'objet possédant un cadre fermé est rempli de haut en bas.

2 = L'objet possédant un cadre fermé est rempli de gauche à droite.

3 = L'objet possédant un cadre fermé est rempli de droite à gauche.

Accès en écriture et en lecture.
Vous définissez le sens de remplissage de l'objet à l'aide de la propriété "FillingDirection".

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, le rectangle est rempli de la gauche vers la droite.

```
Sub RectangleConfiguration()
'VBAxxx
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.FillingDirection = 2
End With
End Sub
```

Propriété FillStyle

Description

Détermine le motif de remplissage pour l'objet ou le restitue.

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
< Transparent >	65536				
< Massif >	0				
	1048576		196611		196627
	1048577		196612		196628
	1048578		196613		196629
	1048579		196614		196630
	1048832		196615		196631
	1048833		196616		196632
	1048834		196617		196633
	1048835		196618		196634
	131072		196619		196635

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
	131073		196620		196636
	131074		196621		196637
	131075		196622		196638
	131076		196623		196639
	196608		196624		196640
	196609		196625		196641
	196610		196626		196642

Exemple

La procédure "RectangleConfiguration()" accède aux propriétés du rectangle. Dans cet exemple, le motif de remplissage est défini sur la valeur "196642" :

```
Sub RectangleConfiguration()
'VBA496
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.FillStyle = 196642
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)
Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet IOField (Page 1947)
Objet GraphicObject (Page 1931)
Objet EllipseSegment (Page 1920)
Objet Ellipse (Page 1914)
Objet Document (Page 1908)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)
Objet BarGraph (Page 1881)

Propriété FillStyle2

Description

Définit le motif de remplissage du bargraphe pour l'objet BarGraph ou le retourne.

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
< Transparent >	65536				
< Massif >	0				
	1048576		196611		196627
	1048577		196612		196628
	1048578		196613		196629

Motif de remplissage	Valeur	Motif de remplissage	Valeur	Motif de remplissage	Valeur
	1048579		196614		196630
	1048832		196615		196631
	1048833		196616		196632
	1048834		196617		196633
	1048835		196618		196634
	131072		196619		196635
	131073		196620		196636
	131074		196621		196637
	131075		196622		196638
	131076		196623		196639
	196608		196624		196640
	196609		196625		196641
	196610		196626		196642

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, le motif du bargraphe est défini sur "196642" :

```
Sub BarGraphConfiguration()
'VBA497
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.FillStyle2 = 196642
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété FillStyleAlignment

Description

Détermine l'orientation du motif de remplissage pour la vue de process.

Normal	Le motif de remplissage se réfère à la vue de process. Lors de l'étirement de la vue en runtime, aucune mise à l'échelle n'est effectuée.
étiré (fenêtre)	Le motif de remplissage se réfère à la fenêtre dans Graphics Designer. Lors de l'étirement de la vue en runtime, la mise à l'échelle est effectuée.

Propriété FlashBackColor

Description

TRUE, lorsque le clignotement de l'arrière-plan de l'objet est activé. BOOLEAN Accès en écriture et en lecture.

Remarque

La modification de l'attribut ne désactive pas automatiquement l'attribut "Style Windows".

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, le clignotement de l'arrière-plan est activé :

```
Sub RectangleConfiguration()
'VBA498
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.FlashBackColor = True
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Objet StaticText (Page 2017)

Objet Slider (Page 2013)

Objet TextList (Page 2025)

Objet RoundRectangle (Page 2006)

Objet Rectangle (Page 2000)

Objet Polygon (Page 1986)

Objet PieSegment (Page 1983)

Objet OptionGroup (Page 1977)

Objet GraphicObject (Page 1931)

Objet IOField (Page 1947)

Objet EllipseSegment (Page 1920)

Objet Ellipse (Page 1914)

Objet Circle (Page 1890)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Propriété FlashBorderColor

Description

TRUE, lorsque le clignotement du trait de l'objet est activé. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, le clignotement du trait est activé :

```
Sub RectangleConfiguration()
'VBA499
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.FlashBorderColor = True
End With
End Sub
```

Voir aussi

Objet StaticText (Page 2017)
Objet StatusDisplay (Page 2020)
Objet Slider (Page 2013)
Objet TextList (Page 2025)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet GraphicObject (Page 1931)
Objet IOField (Page 1947)
Objet EllipseSegment (Page 1920)
Objet EllipseArc (Page 1917)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)

Propriété FlashFlashPicture

Description

TRUE, lorsque le clignotement de la vue clignotante est activé. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, le clignotement de la vue clignotante est activé :

```
Sub StatusDisplayConfiguration()
'VBA500
Dim objsDisplay As HMIStatusDisplay
```

```
Set objsDisplay = ActiveDocument.HMIObjects.AddHMIObject("StatusDisplay1",
"HMIStatusDisplay")
With objsDisplay
.FlashFlashPicture = True
End With
End Sub
```

Voir aussi

Objet StatusDisplay (Page 2020)

Propriété FlashForeColor

Description

TRUE, lorsque le clignotement du texte est activé. BOOLEAN Accès en écriture et en lecture.

Remarque

La modification de l'attribut ne désactive pas automatiquement l'attribut "Style Windows".

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, le clignotement du texte est activé :

```
Sub ButtonConfiguration()
'VBA501
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FlashForeColor = True
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété FlashPicReferenced

Description

TRUE si la vue clignotante affectée à l'objet Indicateur d'état doit être enregistrée. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la vue affectée à l'objet Indicateur d'état doit être enregistrée :

```
Sub StatusDisplayConfiguration()
'VBA502
Dim objStatusDisplay As HMIStatusDisplay
Set objStatusDisplay = ActiveDocument.HMIObjects.AddHMIObject("StatusDisplay1",
"HMIStatusDisplay")
With objStatusDisplay
.FlashPicReferenced = True
End With
End Sub
```

Voir aussi

Objet StatusDisplay (Page 2020)

Propriété FlashPicTransColor

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à la vue clignotante doit être configurée sur "transparent" ou restitue la couleur. LONG Accès en écriture et en lecture.

La couleur n'est définie sur "Transparent" que si la propriété "FlashPicUseTransColor" a la valeur "True".

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la couleur "Jaune" est définie sur Transparent :

```
Sub StatusDisplayConfiguration()
'VBA503
Dim objStatusDisplay As HMIStatusDisplay
Set objStatusDisplay = ActiveDocument.HMIObjects.AddHMIObject("StatusDisplay1",
"HMIStatusDisplay")
With objStatusDisplay
.FlashPicTransColor = RGB(255, 255, 0)
.FlashPicUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété FlashPicUseTransColor (Page 2179)

Objet StatusDisplay (Page 2020)

Propriété FlashPicture

Description

Définit ou renvoie la vue clignotante pour l'objet Indicateur d'état.

La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

La propriété "FlashPicReferenced" définit quant à elle si la vue clignotante doit être enregistrée ou référencée avec l'objet Indicateur d'état.

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la vue "Testbild.BMP" est utilisée comme vue clignotante :

```
Sub StatusDisplayConfiguration()
'VBA504
Dim objStatusDisplay As HMIStatusDisplay
Set objStatusDisplay = ActiveDocument.HMIObjects.AddHMIObject("StatusDisplay1",
"HMIStatusDisplay")
With objStatusDisplay
'
'To use this example copy a Bitmap-Graphic
'to the "GraCS"-Folder of the actual project.
'Replace the picturename "Testpicture.BMP" with the name of
'the picture you copied
```

```
.FlashPicture = "Testpicture.BMP"
End With
End Sub
```

Voir aussi

Propriété FlashPicReferenced (Page 2177)

Objet StatusDisplay (Page 2020)

Propriété FlashPicUseTransColor

Description

TRUE, lorsque la couleur configurée (propriété "FlashPicTransColor") pour l'objet Bitmap assigné à la vue clignotante doit être positionnée sur "transparent". BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "StatusDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la couleur "Jaune" est définie sur Transparent :

```
Sub StatusDisplayConfiguration()
'VBA505
Dim objStatusDisplay As HMIStatusDisplay
Set objStatusDisplay = ActiveDocument.HMIObjects.AddHMIObject("StatusDisplay1",
"HMIStatusDisplay")
With objStatusDisplay
.FlashPicTransColor = RGB(255, 255, 0)
.FlashPicUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété FlashPicTransColor (Page 2177)

Objet StatusDisplay (Page 2020)

Propriété FlashRate

Description

Définit ou renvoie la fréquence de clignotement de l'objet GroupDisplay. Plage des valeurs de 0 à 2.

Fréquence de clignotement	Valeur affectée
Lente (env. 0,5 Hz)	0
Moyenne (env. 2 Hz)	1
Rapide (env. 8 Hz)	2

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence dépend du système et du matériel (nombre d'objets, processeur, mémoire, cycle de rafraîchissement, etc.).

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la fréquence de clignotement est définie sur "Moyenne" :

```
Sub GroupDisplayConfiguration()
'VBA506
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.FlashRate = 1
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété FlashRateBackColor

Description

Définit la fréquence de clignotement pour l'arrière-plan de l'objet ou la restitue. Plage des valeurs de 0 à 2.

Fréquence de clignotement	Valeur affectée
Lente (env. 0,5 Hz)	0
Moyenne (env. 2 Hz)	1
Rapide (env. 8 Hz)	2

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence dépend du système et du matériel (nombre d'objets, processeur, mémoire, cycle de rafraîchissement, etc.).

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la fréquence de clignotement de l'arrière-plan est définie sur "Moyenne" :

```
Sub ButtonConfiguration()
'VBA507
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FlashRateBackColor = 1
End With
End Sub
```

Voir aussi

Objet StaticText (Page 2017)
Objet Slider (Page 2013)
Objet TextList (Page 2025)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet GraphicObject (Page 1931)
Objet IOField (Page 1947)
Objet EllipseSegment (Page 1920)
Objet Ellipse (Page 1914)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)
Objet BarGraph (Page 1881)

Propriété FlashRateBorderColor

Description

Définit la fréquence de clignotement du trait de l'objet ou la restitue. Plage des valeurs de 0 à 2.

Fréquence de clignotement	Valeur affectée
Lente (env. 0,5 Hz)	0
Moyenne (env. 2 Hz)	1
Rapide (env. 8 Hz)	2

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence dépend du système et du matériel (nombre d'objets, processeur, mémoire, cycle de rafraîchissement, etc.).

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la fréquence de clignotement du trait est définie sur "Moyenne" :

```
Sub ButtonConfiguration()
'VBA508
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FlashRateBorderColor = 1
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet TextList (Page 2025)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet PolyLine (Page 1989)
Objet Polygon (Page 1986)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet Line (Page 1958)
Objet GraphicObject (Page 1931)
Objet IOField (Page 1947)
Objet EllipseSegment (Page 1920)
Objet EllipseArc (Page 1917)
Objet Ellipse (Page 1914)
Objet CircularArc (Page 1893)
Objet Circle (Page 1890)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)

Propriété FlashRateFlashPic

Description

Définit ou renvoie la fréquence de clignotement de l'indicateur d'état. Plage des valeurs de 0 à 2.

Fréquence de clignotement	Valeur affectée
Lente (env. 0,5 Hz)	0
Moyenne (env. 2 Hz)	1
Rapide (env. 8 Hz)	2

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence dépend du système et du matériel (nombre d'objets, processeur, mémoire, cycle de rafraîchissement, etc.).

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de l'indicateur d'état. Dans cet exemple, la fréquence de clignotement de la vue clignotante est définie sur "Moyenne" :

```
Sub StatusDisplayConfiguration()
'VBA509
Dim objStatusDisplay As HMIStatusDisplay
Set objStatusDisplay = ActiveDocument.HMIObjects.AddHMIObject("StatusDisplay1",
"HMIStatusDisplay")
With objStatusDisplay
.FlashRateFlashPic = 1
End With
End Sub
```

Voir aussi

Objet StatusDisplay (Page 2020)

Propriété FlashRateForeColor

Description

Définit la fréquence de clignotement pour le texte de l'objet ou la restitue. Plage des valeurs de 0 à 2.

Fréquence de clignotement	Valeur affectée
Lente (env. 0,5 Hz)	0
Moyenne (env. 2 Hz)	1
Rapide (env. 8 Hz)	2

Remarque

Comme le clignotement est réalisé par le logiciel, la fréquence dépend du système et du matériel (nombre d'objets, processeur, mémoire, cycle de rafraîchissement, etc.).

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la fréquence de clignotement du texte est définie sur "Moyenne" :

```
Sub ButtonConfiguration()
'VBA510
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FlashRateForeColor = 1
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété Folder

Description

Renvoie un dossier de la bibliothèque du module.

Exemple

La procédure "ShowFolderItems()" permet d'accéder aux bibliothèques de symboles. Dans cet exemple, les noms de tous les dossiers de la bibliothèque de symboles globale et de la bibliothèque de symboles du projet sont sortis :

```
Sub ShowFolderItems()
'VBA511
Dim colFolderItems As HMIFolderItems
Dim objFolderItem As HMIFolderItem
Dim iAnswer As Integer
Dim iMaxFolder As Integer
Dim iMaxSymbolLib As Integer
Dim iSymbolLibIndex As Integer
Dim iSubFolderIndex As Integer
Dim strSubFolderName As String
Dim strFolderItemName As String
'To determine the number of symbollibraries:
iMaxSymbolLib = Application.SymbolLibraries.Count
iSymbolLibIndex = 1
For iSymbolLibIndex = 1 To iMaxSymbolLib
With Application.SymbolLibraries(iSymbolLibIndex)
Set colFolderItems = .FolderItems
'
'To determine the number of folders in actual symbollibrary:
iMaxFolder = .FolderItems.Count
MsgBox "Number of FolderItems in " & .Name & " : " & iMaxFolder
'
'Output of all subfoldernames from actual folder:
For Each objFolderItem In colFolderItems
  iSubFolderIndex = 1
  For iSubFolderIndex = 1 To iMaxFolder
    strFolderItemName = objFolderItem.DisplayName
    If 0 <> objFolderItem.Folder.Count Then
      strSubFolderName = objFolderItem.Folder(iSubFolderIndex).DisplayName
      iAnswer = MsgBox("SymbolLibrary: " & .Name & vbCrLf & "act. Folder: " &
strFolderItemName & vbCrLf & "act. Subfolder: " & strSubFolderName, vbOKCancel)
      '
      'If "Cancel" is clicked, continued with next FolderItem
      If vbCancel = iAnswer Then
        Exit For
      End If
    Else
      MsgBox "There are no subfolders in " & objFolderItem.DisplayName
        Exit For
    End If
  Next iSubFolderIndex
```

```
Next objFolderItem
End With
Next iSymbolLibIndex
End Sub
```

Voir aussi

Objet SymbolLibraries (liste) (Page 2024)

Objet SymbolLibrary (Page 2023)

Objet FolderItems (liste) (Page 1929)

Objet FolderItem (Page 1927)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Propriété FolderItems

Description

Renvoie un listage contenant tous les dossiers de la bibliothèque de symboles.

Exemple

La procédure "ShowFolderItems()" permet d'accéder aux bibliothèques de symboles. Dans cet exemple, les noms de tous les dossiers de la bibliothèque de symboles globale et de la bibliothèque de symboles du projet sont sortis :

```
Sub ShowFolderItems()
'VBA512
Dim colFolderItems As HMIFolderItems
Dim objFolderItem As HMIFolderItem
Dim iAnswer As Integer
Dim iMaxFolder As Integer
Dim iMaxSymbolLib As Integer
Dim iSymbolLibIndex As Integer
Dim iSubFolderIndex As Integer
Dim strSubFolderName As String
Dim strFolderItemName As String
'To determine the number of symbollibraries:
iMaxSymbolLib = Application.SymbolLibraries.Count
iSymbolLibIndex = 1
For iSymbolLibIndex = 1 To iMaxSymbolLib
With Application.SymbolLibraries(iSymbolLibIndex)
Set colFolderItems = .FolderItems
'
'To determine the number of folders in actual symbollibrary:
iMaxFolder = .FolderItems.Count
MsgBox "Number of FolderItems in " & .Name & " : " & iMaxFolder
'
```

```
'Output of all subfoldernames from actual folder:
For Each objFolderItem In colFolderItems
  iSubFolderIndex = 1
  For iSubFolderIndex = 1 To iMaxFolder
    strFolderItemName = objFolderItem.DisplayName
    If 0 <> objFolderItem.Folder.Count Then
      strSubFolderName = objFolderItem.Folder(iSubFolderIndex).DisplayName
      iAnswer = MsgBox("SymbolLibrary: " & .Name & vbCrLf & "act. Folder: " &
strFolderItemName & vbCrLf & "act. Subfolder: " & strSubFolderName, vbOKCancel)
      '
      'If "Cancel" is clicked, continued with next FolderItem
      If vbCancel = iAnswer Then
        Exit For
    End If
    Else
      MsgBox "There are no subfolders in " & objFolderItem.DisplayName
        Exit For
    End If
  Next iSubFolderIndex
Next objFolderItem
End With
Next iSymbolLibIndex
End Sub
```

Voir aussi

Objet FolderItem (Page 1927)

Objet SymbolLibraries (liste) (Page 2024)

Objet SymbolLibrary (Page 2023)

Objet FolderItems (liste) (Page 1929)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Propriété FontBold

Description

TRUE, lorsque le texte reçoit l'attribut d'écriture "gras" dans l'objet. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, l'attribut de police est défini sur "gras" :

```
Sub ButtonConfiguration()
'VBA513
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FontBold = True
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Propriété FontFamily

Description

Définit ou renvoie la police en fonction de la langue.

Exemple

Dans l'exemple suivant, on configure les attributs de texte d'un bouton pour le français et pour l'anglais :

```
Sub ExampleForLanguageFonts()
'VBA492
Dim colLangFonts As HMILanguageFonts
Dim objButton As HMIButton
Dim iStartLangID As Integer
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
iStartLangID = Application.CurrentDataLanguage
With objButton
.Text = "Command"
```

```
.Width = 100
End With
Set colLangFonts = objButton.LDFonts
'
'To do typesettings for french:
With colLangFonts.ItemByLCID(1036)
.Family = "Courier New"
.Bold = True
.Italic = False
.Underlined = True
.Size = 12
End With
'
'To do typesettings for english:
With colLangFonts.ItemByLCID(1033)
.Family = "Times New Roman"
.Bold = False
.Italic = True
.Underlined = False
.Size = 14
End With
With objButton
Application.CurrentDataLanguage = 1036
.Text = "Command"
MsgBox "Datalanguage is changed in french"
Application.CurrentDataLanguage = 1033
.Text = "Command"
MsgBox "Datalanguage is changed in english"
Application.CurrentDataLanguage = iStartLangID
MsgBox "Datalanguage is changed back to startlanguage."
End With
End Sub
```

Voir aussi

Propriété Underlined (Page 2388)
Propriété Size (Page 2353)
Propriété Parent (Page 2303)
Propriété Italic (Page 2212)
Propriété LanguageID (Page 2219)
Propriété Bold (Page 2094)
Propriété Application (Page 2067)
Objet LanguageFont (Page 1950)

Propriété FontItalic

Description

TRUE, lorsque le texte reçoit l'attribut d'écriture "italique" dans l'objet. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, l'attribut de police est défini sur "italique" :

```
Sub ButtonConfiguration()
'VBA514
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FontItalic = True
End With
End Sub
```

Voir aussi

Objet StaticText (Page 2017)

Objet TextList (Page 2025)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Propriété FontName

Description

Détermine la police de caractères du texte dans l'objet ou la restitue.

Vous disposez de toutes les polices installées sous Windows.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la police est définie sur "Arial" :

```
Sub ButtonConfiguration()
'VBA515
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FontName = "Arial"
End With
End Sub
```

Voir aussi

Objet CheckBox (Page 1889)

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Propriété FontSize

Description

Détermine en points la taille de police du texte dans l'objet ou la restitue.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la taille de la police est définie sur 10 points :

```
Sub ButtonConfiguration()
'VBA516
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FONTSIZE = 10
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Propriété FontUnderline

Description

TRUE, lorsque le texte reçoit l'attribut d'écriture "souligné" dans l'objet. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, l'attribut de police est défini sur "souligné" :

```
Sub ButtonConfiguration()
'VBA517
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.FontUnderline = True
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Propriété ForeColor

Description

Détermine la couleur du texte dans l'objet ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la couleur de la police est définie sur "Rouge" :

```
Sub ButtonConfiguration()
'VBA518
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.ForeColor = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet Button (Page 1886)

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet CheckBox (Page 1889)

Objet BarGraph (Page 1881)

Propriété ForeFlashColorOff

Description

Définit la couleur du texte pour l'état clignotant "Arrêt" ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la couleur de police pour l'état clignotant "Désactivé" est définie sur "Rouge" :

```
Sub ButtonConfiguration()
'VBA519
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.ForeFlashColorOff = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet CheckBox (Page 1889)

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

Propriété ForeFlashColorOn

Description

Définit la couleur du texte pour l'état clignotant "Marche" ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, la couleur de police pour l'état clignotant "Activé" est définie sur "Blanc" :

```
Sub ButtonConfiguration()
'VBA520
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.ForeFlashColorOn = RGB(255, 255, 255)
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet GroupDisplay (Page 1935)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Objet BarGraph (Page 1881)

G-H

Propriété GlobalColorScheme

Description

Définit si les couleurs définies dans le schéma de couleurs global du design actuel sont utilisées pour cet objet.

oui	Adopte les couleurs du schéma de couleurs global défini pour ce type d'objet.
non	Adopte les couleurs du schéma de couleurs individuel, paramétré sous "Couleurs" pour ce type d'objet.

Exemple

--

Propriété GlobalShadow

Description

Définit si l'objet est représenté avec l'ombrage déterminé dans le design actif.

oui	Adopte l'ombrage global défini pour ce type d'objet.
non	Aucune ombre.

Exemple

--

Propriété Grid

Description

TRUE si le quadrillage est activé dans la vue active. BOOLEAN Accès en écriture et en lecture.

Le quadrillage n'est visible que pendant la phase de configuration.

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, le quadrillage est activé pour la vue active :

```
Sub ActiveDocumentConfiguration()
'VBA521
Application.ActiveDocument.Grid = True
End Sub
```

Voir aussi

Propriété GridWidth (Page 2199)

Propriété GridHeight (Page 2198)

Propriété GridColor (Page 2198)

Propriété ActiveDocument (Page 2055)

Propriété Application (Page 2067)

Objet Document (Page 1908)

Objet Application (Page 1876)

Propriété GridColor

Description

Définit ou renvoie la couleur du quadrillage pendant la phase de configuration. La propriété Grid doit être définie sur TRUE pour que le quadrillage soit affiché. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, la couleur du quadrillage pour la vue active est définie sur "Bleu" :

```
Sub ActiveDocumentConfiguration()
'VBA522
Application.ActiveDocument.Grid = True
Application.ActiveDocument.GridColor = RGB(0, 0, 255)
End Sub
```

Voir aussi

Propriété Grid (Page 2197)

Propriété ActiveDocument (Page 2055)

Propriété Application (Page 2067)

Objet Document (Page 1908)

Objet Application (Page 1876)

Propriété GridHeight

Description

Définit ou renvoie la hauteur du quadrillage (en pixels) de la vue actuelle pendant la phase de configuration. La propriété Grid doit être définie sur TRUE pour que le quadrillage soit affiché.

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, la hauteur du quadrillage pour la vue active est définie sur "8" :

```
Sub ActiveDocumentConfiguration()
'VBA523
Application.ActiveDocument.Grid = True
Application.ActiveDocument.GridHeight = 8
End Sub
```

Voir aussi

Propriété GridWidth (Page 2199)

Propriété Grid (Page 2197)

Propriété ActiveDocument (Page 2055)

Propriété Application (Page 2067)

Objet Document (Page 1908)

Objet Application (Page 1876)

Propriété GridWidth

Description

Définit ou renvoie la largeur du quadrillage (en pixels) de la vue actuelle pendant la phase de configuration. La propriété Grid doit être définie sur TRUE pour que le quadrillage soit affiché.

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, la largeur du quadrillage pour la vue active est définie sur "8" :

```
Sub ActiveDocumentConfiguration()
'VBA524
Application.ActiveDocument.Grid = True
Application.ActiveDocument.GridWidth = 8
End Sub
```

Voir aussi

Propriété Grid (Page 2197)

Propriété GridHeight (Page 2198)

Propriété ActiveDocument (Page 2055)

Propriété Application (Page 2067)

Objet Document (Page 1908)

Objet Application (Page 1876)

Propriété GroupParent

Description

Renvoie l'objet prioritaire de l'objet groupe. Accès uniquement en lecture.

Exemple

--

Voir aussi

Objet Group (Page 1934)

Propriété ActiveDocument (Page 2055)

Objet GroupedObjects (liste) (Page 1938)

Objet Document (Page 1908)

Objet Application (Page 1876)

Propriété GroupedHMIObjects

Description

Renvoie un listage contenant tous les objets du groupe actuel.

Exemple

Dans cet exemple, l'objet groupe Group1 est créé à partir d'objets. On ajoute ensuite un segment d'ellipse à l'objet groupe :

```
Sub CreateGroup()
'VBA526
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
```

```
Dim objEllipseSegment As HMIEllipseSegment
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
Set objGroup = ActiveDocument.Selection.CreateGroup
objGroup.ObjectName = "Group1"
Set objEllipseSegment = ActiveDocument.HMIObjects.AddHMIObject("EllipseSegment",
"HMIEllipseSegment")
'
'Add one object to the existing group
objGroup.GroupedHMIObjects.Add ("EllipseSegment")
End Sub
```

Voir aussi

Objet Group (Page 1934)

Propriété Height

Description

Définit ou renvoie la hauteur de l'objet (Document, View, Object) en pixels.

Remarque pour les objets Document et View :

La valeur prédéfinie correspond à la résolution de l'écran définie par le système d'exploitation dans le sens vertical. La valeur peut être supérieure à la résolution de l'écran actuelle. La vue peut être déplacée à l'aide des barres de défilement.

La hauteur de vue définissable max. est de 10000 pixels.

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, la hauteur de la vue actuelle est définie sur "1600" :

```
Sub ActiveDocumentConfiguration()
'VBA527
```

```
Application.ActiveDocument.Height = 1600
End Sub
```

Voir aussi

Objet View (Page 2050)

Objet HMIObject (Page 1943)

Objet Document (Page 1908)

Propriété Hide

Description

TRUE si la vue indiquée est ouverte de façon "visible". BOOLEAN Accès en écriture et en lecture.

Utilisez la propriété Hide pour vérifier par exemple si une vue est ouverte visible ou invisible. D'autres éditeurs WinCC (exemple : CrossReference) ouvrent des vues invisibles, ce qui veut dire qu'elles ne s'affichent pas dans Graphics Designer. Si par exemple, vous utilisez l'événement DocumentOpened, vous pouvez, au moyen de la propriété Hide, empêcher l'exécution du code dans l'événement en vérifiant si la propriété Hide est sur FALSE.

A l'aide de la méthode Add et de la méthode Open, vous pouvez définir d'ouvrir une vue visible ou invisible.

Remarque

Si vous réglez la vue sur "invisible" (Hide = FALSE), vous ne pouvez la joindre que par le listage Documents. Dans Graphics Designer, la vue n'est plus disponible.

Exemple

Dans l'exemple suivant, on voit si la vue est visible ou invisible lors de son ouverture :

```
Private Sub Document_Opened(CancelForwarding As Boolean)
'VBA802
MsgBox Me.Hide
End Sub
```

Voir aussi

Méthode Open (Page 1842)

Méthode Add (listage Documents) (Page 1769)

Objet Document (Page 1908)

Propriété HiddenInput

Description

TRUE, lorsque la valeur d'entrée n'est pas affichée pendant la saisie. Un * s'affiche pour chaque caractère. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, la saisie doit être masquée :

```
Sub IOFieldConfiguration()
'VBA528
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.HiddenInput = True
End With
End Sub
```

Voir aussi

Objet IOField (Page 1947)

Propriété HMIObjects

Description

Renvoie un listage contenant tous les objets de la vue indiquée.

Pour renvoyer un élément du listage HMIObjects, vous pouvez utiliser le numéro d'index ou le nom d'objet.

Exemple

Utilisez la méthode "AddHMIObject(ObjectName, ProgID)" pour insérer un nouvel objet dans une vue. :

```
Sub AddCircle()
'VBA529
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("my Circle", "HMICircle")
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété Hotkey

Description

Définit ou renvoie pour l'objet Button la touche de fonction pour le clic de souris.

Touche de fonction	Valeur affectée
F1	112
F2	113
F3	114
F4	115
F5	116
F6	117
F7	118
F8	119
F9	120
F10	121
F11	122
F12	123

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, le bouton peut également être activé avec la touche de fonction "F5" :

```
Sub ButtonConfiguration()
'VBA530
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.Hotkey = 116
End With
End Sub
```

Voir aussi

Objet Button (Page 1886)

Propriété Hysteresis

Description

TRUE si l'affichage doit s'effectuer avec une hystérésis pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, l'affichage doit s'effectuer avec une hystérésis :

```
Sub BarGraphConfiguration()
'VBA531
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Hysteresis = True
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété HysteresisRange

Description

Définit ou renvoie l'hystérésis en % de la valeur d'affichage.

La propriété "Hysteresis" doit avoir la valeur TRUE pour que l'hystérésis puisse être calculée.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, l'hystérésis est définie sur "4 %" :

```
Sub BarGraphConfiguration()
'VBA532
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Hysteresis = True
.HysteresisRange = 4
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété Hysteresis (Page 2205)

I - K

Propriété Icon

Description

Définit l'icône (*.ICO, chemin complet et nom de fichier) pour le bouton d'une barre d'outils personnalisée ou renvoie le chemin et le nom de fichier.

Indication du chemin

Les formats d'indication du chemin suivants sont possibles :

- absolue : par ex. "C:\Siemens\WinCC\Icons\myIcon.ICO.
- relative : le répertoire initial pour l'indication relative du chemin est le répertoire "GraCS" du projet actuel.
- <global> : désigne le chemin d'installation de WinCC. L'indication du chemin "<global> \Icons\myIcon" correspond à l'indication de chemin donnée sous "absolu".
- <project> : désigne le répertoire de projet actuel (voir exemple).

Exemple

La procédure "CreateToolbar()" crée une barre d'outils personnalisée à deux icônes :

```
Sub CreateToolbar()
'VBA533
Dim objToolbar As HMIToolbar
Dim objToolbarItem As HMIToolbarItem
Dim strFileWithPath
Set objToolbar = ActiveDocument.CustomToolbars.Add("Tool1_1")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(1, "ti1_1",
"myFirstToolbaritem")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(2, "ti1_2",
"mySecondToolbaritem")
'
'To use this example copy a *.ICO-Graphic
'to the "GraCS"-Folder of the actual project.
'Replace the filename "EZSTART.ICO" in the next commandline
'with the name of the ICO-Graphic you copied
strFileWithPath = Application.ApplicationDataPath & "EZSTART.ICO"
'
'To assign the symbol-icon to the first toolbaritem
objToolbar.ToolbarItems(1).Icon = strFileWithPath
End Sub
```

Voir aussi

Objet ToolbarItems (liste) (Page 2034)

Objet ToolbarItem (Page 2031)

Insertion d'une nouvelle icône d'outil dans une barre d'outils (Page 1630)

Pour créer une nouvelle barre d'outils spécifique à une application (Page 1629)

Propriété IndependentWindow

Description

Définit si la représentation de la fenêtre de vue en runtime dépend de la vue de process dans laquelle la fenêtre de vue a été configurée.

oui	La taille et la position de la fenêtre de vue ne dépendent pas de la vue de process et sont uniquement définies par l'attribut "Mode de fenêtre".
non	La taille et la position de la fenêtre de vue changent en fonction du déplacement ou de la mise à l'échelle de la vue de process.

Propriété Index

Description

Indicateur d'état

Détermine l'état (0 à 255) ou le restitue. Vous pouvez spécifier une vue de base et une vue clignotante pour chaque valeur d'état.

Objet Line

Définit le point de départ et le point terminal d'une ligne et donc la direction. Pour chaque point de départ et point de fin, définissez les coordonnées à l'aide des propriétés ActualPointLeft et ActualPointTop.

Objet Polygon, objet PolyLine et objet TubePolyline

Définit ou retourne le numéro du point d'angle dont vous voulez modifier ou faire afficher les coordonnées de position.

Objets CheckBox et OptionGroup

Définit les numéros (1 à 32) du champ dont vous voulez définir le texte, ou les retourne.

Objet ComboBox et objet ListBox

Définit les numéros (1 à 32) de la ligne dont vous voulez définir le texte, ou les retourne.

Exemple 1 : Trait

Dans l'exemple qui suit, un trait est inséré dans la vue active et les points de départ et de fin sont définis :

```
Sub LineAdd()
'VBA682
Dim objLine As HMILine
Dim objEvent As HMIEvent
Set objLine = ActiveDocument.HMIObjects.AddHMIObject("myLine", "HMILine")
With objLine
.BorderColor = RGB(255, 0, 0)
.index = hmiLineIndexTypeStartPoint
.ActualPointLeft = 12
.ActualPointTop = 34
.index = hmiLineIndexTypeEndPoint
.ActualPointLeft = 74
.ActualPointTop = 64
End With
End Sub
```

Exemple 2 : Trait polygonal

Pour que cet exemple fonctionne, insérez un trait polygonal "Trait polygonal1" dans la vue active. La procédure "PolyLineCoordsOutput" permet de sortir les coordonnées de tous les angles du trait polygonal :

```
Sub PolyLineCoordsOutput()
'VBA534
Dim iPcIndex As Integer
Dim iPosX As Integer
Dim iPosY As Integer
Dim iIndex As Integer
Dim objPolyLine As HMIPolyLine
Set objPolyLine = Application.ActiveDocument.HMIObjects.AddHMIObject("PolyLine1",
"HMIPolyLine")
'
'Determine number of corners from "PolyLine1":
iPcIndex = objPolyLine.PointCount
'
'Output of x/y-coordinates from every corner:
For iIndex = 1 To iPcIndex
With objPolyLine
.index = iIndex
iPosX = .ActualPointLeft
iPosY = .ActualPointTop
MsgBox iIndex & ". corner:" & vbCrLf & "x-coordinate: " & iPosX & vbCrLf & "y-coordinate:
" & iPosY
End With
Next iIndex
End Sub
```

Exemple 3 : Case à cocher

La procédure "CreateOptionGroup()" permet de créer l'objet OptionGroup avec quatre champs d'option. A chaque champ d'option est attribué le nom standard "myCustomText<Numéro>" :

```
Sub CreateOptionGroup()
'VBA535
Dim objRadioBox As HMIOptionGroup
Dim iIndex As Integer
Set objRadioBox = ActiveDocument.HMIObjects.AddHMIObject("RadioBox_1", "HMIOptionGroup")
With objRadioBox
.Height = 100
.Width = 180
.BoxCount = 4
For iIndex = 1 To .BoxCount
.index = iIndex
.Text = "myCustomText" & .index
Next iIndex
End With
End Sub
```

Voir aussi

Objet Line (Page 1958)

Propriété FlashPicture (Page 2178)

Propriété BasePicture (Page 2088)

Propriété ActualPointTop (Page 2057)

Propriété ActualPointLeft (Page 2056)

Objet StatusDisplay (Page 2020)

Objet PolyLine (Page 1989)

Objet Polygon (Page 1986)

Objet OptionGroup (Page 1977)

Propriété InheritState

Description

Définit si les états des propriétés "Affichage" et "Autorisation de conduite" de l'objet utilisateur sont transmissibles pour les objets individuels de l'objet utilisateur.

Propriété InputValue

Description

Définit la valeur indiquée par l'utilisateur dans le champ d'E/S. La valeur n'est pas affichée dans le champ d'E/S lors de la définition de la propriété.

Si vous voulez que la valeur saisie soit affichée dans le champ d'E/S après confirmation via la pression de la touche <Entrée>, configurez une liaison directe entre les propriétés "Valeur d'entrée" et "Valeur de sortie". La liaison directe n'est intéressante que si aucune liaison de variable n'a été configurée pour la valeur de sortie, mais que l'utilisateur désire néanmoins interroger la valeur saisie, par ex. via un script.

Exemple

Propriété IsActive

Description

Renvoie TRUE si la copie de la vue actuelle est active. Accès en lecture BOOLEAN.

Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, une copie de la vue actuelle est créée et sortie pour indiquer si elle est active.

```
Sub ActiveDocumentConfiguration()
'VBA537
Application.ActiveDocument.Views.Add
'If you comment out the following line
'and recall the procedure, the output of
'the messagebox is different
Application.ActiveDocument.Views(1).Activate
'
'Output state of copy:
MsgBox Application.ActiveDocument.Views(1).IsActive
End Sub
```

Voir aussi

Propriété ActiveDocument (Page 2055)

Objet View (Page 2050)

Propriété IsConnectedToProject

Description

Renvoie TRUE si la liaison du projet est disponible. Accès en lecture BOOLEAN.

Exemple

La procédure "ConnectCheck()" permet de vérifier si une liaison de projet existe et de sortir le résultat :

```
Sub ConnectCheck()
'VBA538
Dim bCheck As Boolean
Dim strStatus As String
bCheck = Application.IsConnectedToProject
If bCheck = True Then
strStatus = "yes"
    Else
strStatus = "no"
    End If
MsgBox "Connection to project available: " & strStatus
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété IsDynamicable

Description

TRUE si une propriété peut être dynamisée. Accès en lecture BOOLEAN.

Exemple

L'événement HMIObjectPropertyChanged se déclenche toujours lorsque vous modifiez une propriété d'objet dans Graphics Designer. Dans cet exemple, le nom de la propriété et sa valeur sont sortis. On vérifie également si la propriété peut être dynamisée :

```
Sub Document_HMIObjectPropertyChanged(ByVal Property As IHMIProperty, CancelForwarding As
Boolean)
'VBA539
Dim objProp As HMIProperty
Dim strStatus As String
Set objProp = Property
'
'Checks whether property is dynamicable
If objProp.IsDynamicable = True Then
strStatus = "yes"
    Else
strStatus = "no"
    End If
MsgBox "Property: " & objProp.Name & vbCrLf & "Value: " & objProp.value & vbCrLf &
"Dynamicable: " & strStatus
End Sub
```

Pour plus d'informations sur les événements, reportez-vous à la rubrique "Exécuter des macros VBA dans Graphics Designer".

Voir aussi

Objet Property (Page 1993)

Objet HMIObject (Page 1943)

Evénement HMIObjectPropertyChanged (Page 1749)

Exécution de macros VBA dans le Graphics Designer (Page 1612)

Propriété Italic

Description

TRUE si l'attribut de police "Italique" est défini dans l'objet pour le texte dépendant des langues. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple suivant, on configure les attributs de texte d'un bouton pour le français et pour l'anglais :

```
Sub ExampleForLanguageFonts()
'VBA540
```

```
Dim objLangFonts As HMILanguageFonts
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
objButton.Text = "Hello"
Set objLangFonts = objButton.LDFonts
'
'To make fontsettings for french:
With objLangFonts.ItemByLCID(1036)
.Family = "Courier New"
.Bold = True
.Italic = False
.Underlined = True
.Size = 12
End With
'
'To make fontsettings for english:
With objLangFonts.ItemByLCID(1033)
.Family = "Times New Roman"
.Bold = False
.Italic = True
.Underlined = False
.Size = 14
End With
End Sub
```

Voir aussi

Propriété Underlined (Page 2388)

Propriété Size (Page 2353)

Propriété Parent (Page 2303)

Propriété LanguageID (Page 2219)

Propriété FontFamily (Page 2189)

Propriété Bold (Page 2094)

Propriété Application (Page 2067)

Objet LanguageFont (Page 1950)

Propriété Item

Description

Renvoie un élément à partir d'un listage. Selon l'objet indiqué, vous pouvez utiliser le numéro d'index ou le nom pour renvoyer un élément donné.

Exemple

Cet exemple montre deux cas d'indexation. Pour que l'exemple fonctionne, définissez un objet groupe ("Group1") avec deux objets. La hauteur du deuxième objet d'un groupe est sortie :

```
Sub GetHeight()
'VBA541
Dim objGroup As HMIGroup
'Next line uses the property "Item" to get a group by name
Set objGroup = ActiveDocument.HMIObjects.Item("Group1")
'Otherwise next line uses index to identify a groupobject
MsgBox "The height of object 2 is: " & objGroup.GroupedHMIObjects.Item(2).Height
End Sub
```

Voir aussi

Objet VariableTriggers (liste) (Page 2049)

Objet VariableStateValues (liste) (Page 2046)

Objet AnalogResultInfos (liste) (Page 1875)

Propriété ItemBorderBackColor

Description

Définit ou renvoie la couleur d'arrière-plan des lignes de séparation de la liste de sélection de l'objet TextList. LONG Accès en écriture et en lecture.

La couleur d'arrière-plan n'est visible que si la propriété ItemBorderStyle > 0.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, la couleur d'arrière-plan des lignes de séparation est réglée sur "Rouge" :

```
Sub TextListConfiguration()
'VBA542
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.ItemBorderStyle = 1
.ItemBorderBackColor = RGB(255, 0, 0)
```

```
End With
End Sub
```

Voir aussi

Propriété ItemBorderStyle (Page 2216)

Objet TextList (Page 2025)

Propriété ItemBorderColor

Description

Définit ou renvoie la couleur des lignes de séparation de la liste de sélection de l'objet TextList. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, la couleur des lignes de séparation est réglée sur "Blanc" :

```
Sub TextListConfiguration()
'VBA543
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.ItemBorderStyle = 1
.ItemBorderColor = RGB(255, 255, 255)
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété ItemBorderStyle

Description

Définit ou renvoie le style de ligne de séparation de l'objet TextList. Plage des valeurs de 0 à 4.

Style de trait	Valeur affectée
———	0
— —	1
- - - - - -	2
— - —	3
—- - —	4

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, le style de ligne de séparation est défini sur "1" :

```
Sub TextListConfiguration()
'VBA544
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.ItemBorderStyle = 1
.ItemBorderBackColor = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété ItemBorderWidth

Description

Définit ou renvoie la largeur de ligne de séparation de la liste de sélection de l'objet TextList en pixels.

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, la largeur de ligne de séparation est définie sur "4" :

```
'Sub E_628_TextListConfiguration()
```

```
Sub E_629_TextListConfiguration()
'VBA545
Dim objTextList As HMITextList
Set objTextList =
ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.ItemBorderWidth = 4
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété Key

Description

Renvoie la désignation identifiant l'entrée (option de menu ou icône) du menu personnalisé ou de la barre d'outils personnalisée. Accès en lecture.

Utilisez la propriété Key pour déterminer l'entrée sur laquelle le clic a été effectué. Pour cela, les événements "MenuItemClicked" et "ToolBarItemClicked" sont disponibles par ex.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Supprimer objets" et d'ajouter deux options de menu ("Supprimer rectangle" et "Supprimer cercle").

```
Sub CreateMenuItem()
'VBA546
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
'
'Add new menu "Delete objects" to menubar:
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete objects")
'
'Adds two menuitems to menu "Delete objects"
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
Rectangles")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete Circles")
End Sub
```

Avec l'événement "MenuItemClicked", vous pouvez lier les options de menu par ex. à des appels de procédure. Dans cet exemple, le nom des options de menu est sorti :

```
Sub Document_MenuItemClicked(ByVal MenuItem As IHMIMenuItem)
'VBA547
Dim strClicked As String
Dim objMenuItem As HMIMenuItem
Set objMenuItem = MenuItem
'
'"strClicked can get two values:
'(1) "DeleteAllRectangles" and
'(2) "DeleteAllCircles"
strClicked = objMenuItem.Key
'
'To analyse "strClicked" with "Select Case"
Select Case strClicked
Case "DeleteAllRectangles"
'
'Instead of "MsgBox" a procedurecall (e.g. "Call <Nom de procédure>") can stay here
MsgBox "'Delete rectangle' was clicked"
Case "DeleteAllCircles"
MsgBox "'Delete Circles' was clicked"
End Select
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Méthode InsertToolbarItem (Page 1831)

Méthode InsertMenuItem (Page 1826)

Evénement ToolbarItemClicked (Page 1761)

Evénement MenuItemClicked (Page 1755)

Création de menus et de barres d'outils personnalisés (Page 1620)

L

La

Propriété Label

Description

Renvoie la désignation du menu défini par l'utilisateur ou de l'option de menu dans la langue actuellement définie. Accès en lecture.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Supprimer objets" et d'ajouter deux options de menu ("Supprimer rectangle" et "Supprimer cercle"). Dans cet exemple, les textes sont sortis :

```
Sub CreateMenuItem()
'VBA548
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim iIndex As Integer
iIndex = 1
'
'Add new menu "Delete objects" to menubar
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete objects")
'
'Adds two menuitems to menu "Delete objects"
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
rectangles")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete circles")
MsgBox ActiveDocument.CustomMenus(1).Label
For iIndex = 1 To objMenu.MenuItems.Count
MsgBox objMenu.MenuItems(iIndex).Label
Next iIndex
End Sub
```

Voir aussi

Propriété CustomMenus (Page 2146)

Objet MenuItems (liste) (Page 1970)

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Propriété LanguageID

Description

Renvoie le code langue de la langue de configuration sous forme de valeur décimale. Accès en lecture LONG.

Exemple

La procédure "DataLanguages()" permet de sortir les langues de configuration avec leur code :

```
Sub DataLanguages()
'VBA549
Dim colDataLang As HMIDataLanguages
```

```
Dim objDataLang As HMIDataLanguage
Dim nLangID As Long
Dim strLangName As String
Dim iAnswer As Integer
Set colDataLang = Application.AvailableDataLanguages
For Each objDataLang In colDataLang
nLangID = objDataLang.LanguageID
strLangName = objDataLang.LanguageName
iAnswer = MsgBox(nLangID & " " & strLangName, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objDataLang
End Sub
```

Voir aussi

Objet DataLanguages (liste) (Page 1903)

Objet DataLanguage (Page 1902)

Propriété LanguageName

Description

Renvoie la langue de configuration. Accès en lecture STRING.

Exemple

La procédure "DataLanguages()" permet de sortir les langues de configuration avec leur code :

```
Sub DataLanguages()
'VBA550
Dim colDataLang As HMIDataLanguages
Dim objDataLang As HMIDataLanguage
Dim nLangID As Long
Dim strLangName As String
Dim iAnswer As Integer
Set colDataLang = Application.AvailableDataLanguages
For Each objDataLang In colDataLang
nLangID = objDataLang.LanguageID
strLangName = objDataLang.LanguageName
iAnswer = MsgBox(nLangID & " " & strLangName, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objDataLang
End Sub
```

Voir aussi

Objet DataLanguages (liste) (Page 1903)

Objet DataLanguage (Page 1902)

Propriété LanguageSwitch

Description

Définit le lieu d'enregistrement des textes de coordonnée dépendant des langues ou renvoie la valeur. BOOLEAN Accès en écriture et en lecture.

TRUE, lorsque les textes sont gérés dans la bibliothèque de textes. Les traductions en d'autres langues s'effectuent dans la bibliothèque de textes.

FALSE, lorsque les textes sont directement gérés dans l'objet. Les traductions en d'autres langues peuvent se faire à l'aide de Text Distributor.

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, les textes doivent être gérés dans la bibliothèque de textes :

```
Sub TextListConfiguration()
'VBA551
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.LanguageSwitch = True
End With
End Sub
```

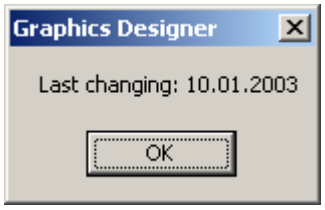
Voir aussi

Objet TextList (Page 2025)

Propriété LastChange

Description

Renvoie la date de la dernière modification de la vue active. Accès en lecture.



Exemple

La procédure "ActiveDocumentConfiguration()" permet d'accéder aux propriétés de la vue actuelle dans Graphics Designer. Dans cet exemple, la dernière modification de la vue actuelle est sortie :

```
Sub ActiveDocumentConfiguration()
'VBA552
Dim varLastDocChange As Variant
varLastDocChange = Application.ActiveDocument.LastChange
MsgBox "Last changing: " & varLastDocChange
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété Layer

Description

Définit ou renvoie le plan (Layer) de la vue dans lequel se trouve un objet. 32 niveaux sont au total disponibles, le niveau "0" étant le plus bas et le niveau "31" le plus élevé.

Dans un niveau, ce sont les objets configurés en premier qui se trouvent au dernier plan.

Remarque

La numérotation commence à "1" dans VBA. Un "objRectangle.Layer = 1" se trouve donc dans le plan inférieur.

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, l'objet Rectangle est inséré dans le plan "4" :

```
Sub RectangleConfiguration()
'VBA553
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.Layer = 4
End With
End Sub
```

Voir aussi

Objet HMIObject (Page 1943)

Edition de niveaux avec VBA (Page 1650)

Propriété Layer00Checked

Description

TRUE si le seuil 0 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer00Value et Layer00Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 0 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA554
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer00Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer00Value (Page 2225)

Propriété Layer00Color (Page 2224)

Objet 3DBarGraph (Page 1867)

Propriété Layer00Color

Description

Définit ou renvoie la couleur du seuil 0 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer00Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 0 :

```
Sub HMI3DBarGraphConfiguration()
'VBA555
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer00Checked = True
.Layer00Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer00Checked (Page 2223)

Objet 3DBarGraph (Page 1867)

Propriété Layer00Value

Description

Définit ou renvoie la valeur du "Seuil 0" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer00Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "0" est définie pour le seuil 0 :

```
Sub HMI3DBarGraphConfiguration()
'VBA556
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer00Checked = True
.Layer00Value = 0
End With
End Sub
```

Voir aussi

Propriété Layer00Checked (Page 2223)

Objet 3DBarGraph (Page 1867)

Propriété Layer01Checked

Description

TRUE si le seuil 1 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer01Value et Layer01Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 1 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA557
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer01Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer01Value (Page 2227)

Propriété Layer01Color (Page 2226)

Objet 3DBarGraph (Page 1867)

Propriété Layer01Color

Description

Définit ou renvoie la couleur du seuil 1 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer01Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 1 :

```
Sub HMI3DBarGraphConfiguration()
'VBA558
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer01Checked = True
.Layer01Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer01Checked (Page 2225)

Objet 3DBarGraph (Page 1867)

Propriété Layer01Value

Description

Définit ou renvoie la valeur du "Seuil 1" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer01Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "10" est définie pour le seuil 1 :

```
Sub HMI3DBarGraphConfiguration()
'VBA559
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer01Checked = True
.Layer01Value = 10
End With
End Sub
```

Voir aussi

Propriété Layer01Checked (Page 2225)

Objet 3DBarGraph (Page 1867)

Propriété Layer02Checked

Description

TRUE si le seuil 2 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer02Value et Layer02Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 2 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA560
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer02Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer02Value (Page 2229)

Propriété Layer02Color (Page 2228)

Objet 3DBarGraph (Page 1867)

Propriété Layer02Color

Description

Définit ou renvoie la couleur du seuil 2 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer02Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 2 :

```
Sub HMI3DBarGraphConfiguration()
'VBA561
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer02Checked = True
.Layer02Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer02Checked (Page 2227)

Objet 3DBarGraph (Page 1867)

Propriété Layer02Value

Description

Définit ou renvoie la valeur du "Seuil 2" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer02Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "20" est définie pour le seuil 2 :

```
Sub HMI3DBarGraphConfiguration()
'VBA562
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer02Checked = True
.Layer02Value = 0
End With
End Sub
```

Voir aussi

Propriété Layer02Checked (Page 2227)

Objet 3DBarGraph (Page 1867)

Propriété Layer03Checked

Description

TRUE si le seuil 3 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer03Value et Layer03Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 3 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA563
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer03Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer03Value (Page 2231)

Propriété Layer03Color (Page 2230)

Objet 3DBarGraph (Page 1867)

Propriété Layer03Color

Description

Définit ou renvoie la couleur du seuil 3 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer03Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 3 :

```
Sub HMI3DBarGraphConfiguration()
'VBA564
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer03Checked = True
.Layer03Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer03Checked (Page 2229)

Objet 3DBarGraph (Page 1867)

Propriété Layer03Value

Description

Définit ou renvoie la valeur du "Seuil 3" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer03Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "30" est définie pour le seuil 3 :

```
Sub HMI3DBarGraphConfiguration()
'VBA565
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer03Checked = True
.Layer03Value = 30
End With
End Sub
```

Voir aussi

Propriété Layer03Checked (Page 2229)

Objet 3DBarGraph (Page 1867)

Propriété Layer04Checked

Description

TRUE si le seuil 4 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer04Value et Layer04Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 4 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA566
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer04Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer04Value (Page 2233)

Propriété Layer04Color (Page 2232)

Objet 3DBarGraph (Page 1867)

Propriété Layer04Color

Description

Définit ou renvoie la couleur du seuil 4 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer04Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 4 :

```
Sub HMI3DBarGraphConfiguration()
'VBA567
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer04Checked = True
.Layer04Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer04Checked (Page 2231)

Objet 3DBarGraph (Page 1867)

Propriété Layer04Value

Description

Définit ou renvoie la valeur du "Seuil 4" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer00Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "40" est définie pour le seuil 4 :

```
Sub HMI3DBarGraphConfiguration()
'VBA568
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer04Checked = True
.Layer04Value = 40
End With
End Sub
```

Voir aussi

Propriété Layer00Checked (Page 2223)

Objet 3DBarGraph (Page 1867)

Propriété Layer05Checked

Description

TRUE si le seuil 5 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer05Value et Layer05Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 5 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA569
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer05Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer05Value (Page 2235)

Propriété Layer05Color (Page 2234)

Objet 3DBarGraph (Page 1867)

Propriété Layer05Color

Description

Définit ou renvoie la couleur du seuil 5 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer05Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 5 :

```
Sub HMI3DBarGraphConfiguration()
'VBA570
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer05Checked = True
.Layer05Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer05Checked (Page 2233)

Objet 3DBarGraph (Page 1867)

Propriété Layer05Value

Description

Définit ou renvoie la valeur du "Seuil 5" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer05Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "50" est définie pour le seuil 5 :

```
Sub HMI3DBarGraphConfiguration()
'VBA571
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer05Checked = True
.Layer05Value = 50
End With
End Sub
```

Voir aussi

Propriété Layer05Checked (Page 2233)

Objet 3DBarGraph (Page 1867)

Propriété Layer06Checked

Description

TRUE si le seuil 6 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer06Value et Layer06Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 6 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA572
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer06Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer06Value (Page 2237)

Propriété Layer06Color (Page 2236)

Objet 3DBarGraph (Page 1867)

Propriété Layer06Color

Description

Définit ou renvoie la couleur du seuil 6 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer06Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 6 :

```
Sub HMI3DBarGraphConfiguration()
'VBA573
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer06Checked = True
.Layer06Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer06Checked (Page 2235)

Objet 3DBarGraph (Page 1867)

Propriété Layer06Value

Description

Définit ou renvoie la valeur du "Seuil 6" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer06Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "60" est définie pour le seuil 6 :

```
Sub HMI3DBarGraphConfiguration()
'VBA574
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer06Checked = True
.Layer06Value = 60
End With
End Sub
```

Voir aussi

Propriété Layer06Checked (Page 2235)

Objet 3DBarGraph (Page 1867)

Propriété Layer07Checked

Description

TRUE si le seuil 7 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer07Value et Layer07Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 7 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA575
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer07Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer07Value (Page 2239)

Propriété Layer07Color (Page 2238)

Objet 3DBarGraph (Page 1867)

Propriété Layer07Color

Description

Définit ou renvoie la couleur du seuil 7 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer07Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 7 :

```
Sub HMI3DBarGraphConfiguration()
'VBA576
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer07Checked = True
.Layer07Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer07Checked (Page 2237)

Objet 3DBarGraph (Page 1867)

Propriété Layer07Value

Description

Définit ou renvoie la valeur du "Seuil 7" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer07Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "70" est définie pour le seuil 7 :

```
Sub HMI3DBarGraphConfiguration()
'VBA577
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer07Checked = True
.Layer07Value = 70
End With
End Sub
```

Voir aussi

Propriété Layer07Checked (Page 2237)

Objet 3DBarGraph (Page 1867)

Propriété Layer08Checked

Description

TRUE si le seuil 8 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer08Value et Layer08Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 8 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA578
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer08Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer08Value (Page 2241)

Propriété Layer08Color (Page 2240)

Objet 3DBarGraph (Page 1867)

Propriété Layer08Color

Description

Définit ou renvoie la couleur du seuil 8 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer08Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 8 :

```
Sub HMI3DBarGraphConfiguration()
'VBA579
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer08Checked = True
.Layer08Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer08Checked (Page 2239)

Objet 3DBarGraph (Page 1867)

Propriété Layer08Value

Description

Définit ou renvoie la valeur du "Seuil 8" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer08Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "80" est définie pour le seuil 8 :

```
Sub HMI3DBarGraphConfiguration()
'VBA580
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer08Checked = True
.Layer08Value = 80
End With
End Sub
```

Voir aussi

Propriété Layer08Checked (Page 2239)

Objet 3DBarGraph (Page 1867)

Propriété Layer09Checked

Description

TRUE si le seuil 9 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer09Value et Layer09Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 9 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA581
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer09Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer09Value (Page 2243)

Propriété Layer09Color (Page 2242)

Objet 3DBarGraph (Page 1867)

Propriété Layer09Color

Description

Définit ou renvoie la couleur du seuil 9 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer09Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 9 :

```
Sub HMI3DBarGraphConfiguration()
'VBA582
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer09Checked = True
.Layer09Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer09Checked (Page 2241)

Objet 3DBarGraph (Page 1867)

Propriété Layer09Value

Description

Définit ou renvoie la valeur du "Seuil 9" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer09Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "90" est définie pour le seuil 9 :

```
Sub HMI3DBarGraphConfiguration()
'VBA583
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer09Checked = True
.Layer09Value = 90
End With
End Sub
```

Voir aussi

Propriété Layer09Checked (Page 2241)

Objet 3DBarGraph (Page 1867)

Propriété Layer10Checked

Description

TRUE si le seuil 10 doit être surveillé pour l'objet 3D-BarGraph. BOOLEAN Accès en écriture et en lecture.

Le seuil et la représentation sont déterminés par les propriétés Layer10Value et Layer10Color.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, le seuil 10 doit être surveillé :

```
Sub HMI3DBarGraphConfiguration()
'VBA584
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
```

```
.Layer10Checked = True
End With
End Sub
```

Voir aussi

Propriété Layer10Value (Page 2245)

Propriété Layer10Color (Page 2244)

Objet 3DBarGraph (Page 1867)

Propriété Layer10Color

Description

Définit ou renvoie la couleur du seuil 10 de l'objet 3DBarGraph. LONG Accès en écriture et en lecture.

Lorsque la surveillance du seuil est activée (Propriété Layer10Checked), le bargraphe prend la couleur de cet attribut lorsque le seuil est atteint.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la couleur "Magenta" est définie pour le seuil 10 :

```
Sub HMI3DBarGraphConfiguration()
'VBA585
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer10Checked = True
.Layer10Color = RGB(255, 0, 255)
End With
End Sub
```

Voir aussi

Propriété Layer10Checked (Page 2243)

Objet 3DBarGraph (Page 1867)

Propriété Layer10Value

Description

Définit ou renvoie la valeur du "Seuil 10" pour l'objet 3DBarGraph.

La surveillance n'est active que si la propriété Layer10Checked a la valeur TRUE.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur "100" est définie pour le seuil 10 :

```
Sub HMI3DBarGraphConfiguration()
'VBA586
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.Layer10Checked = True
.Layer10Value = 100
End With
End Sub
```

Voir aussi

Propriété Layer10Checked (Page 2243)

Objet 3DBarGraph (Page 1867)

Propriété LayerDecluttering

Description

TRUE si l'affichage et le masquage des objets est activé selon le zoom minimal et le zoom maximal définis pour un plan. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple qui suit, on définit les paramètres du niveau le plus bas de la vue active :

```
Sub ConfigureSettingsOfLayer
```

```
'VBA587
Dim objLayer As HMILayer
Set objLayer = ActiveDocument.Layers(1)
With objLayer
'configure "Layer 0"
.MinZoom = 10
.MaxZoom = 100
.Name = "Configured with VBA"
End With
'define fade-in and fade-out of objects:
With ActiveDocument
.LayerDecluttering = True
.ObjectSizeDecluttering = True
.SetDeclutterObjectSize 50, 100
End With
End Sub
```

Voir aussi

Objet Document (Page 1908)

Edition de niveaux avec VBA (Page 1650)

Propriété Layers

Description

Renvoie un listage contenant les propriétés des plans de la vue actuelle.

Remarque

Lorsque la propriété "Layers" est modifiée, l'ordre des objets HMI peut changer dans la liste HMIObjects.

Exemple

La procédure "LayerInfo()" permet de sortir pour chaque plan de la vue actuelle les noms et les configurations de zoom définies :

```
Sub LayerInfo()
'VBA588
Dim colLayers As HMILayers
Dim objLayer As HMILayer
Dim iAnswer As Integer
Set colLayers = ActiveDocument.Layers
For Each objLayer In colLayers
With objLayer
```

```
iAnswer = MsgBox("Layername: " & .Name & vbCrLf & "max. zoom:  " & .MaxZoom & vbCrLf & "min.
zoom:  " & .MinZoom, vbOKCancel)
End With
If vbCancel = iAnswer Then Exit For
Next objLayer
End Sub
```

Voir aussi

Propriété Name (Page 2289)

Propriété MinZoom (Page 2287)

Propriété MaxZoom (Page 2270)

Objet Layers (Liste) (Page 1956)

Objet Layer (Page 1955)

LD - Lo

Propriété LDFonts

Description

Renvoie un listage contenant les codes langues des polices configurées.

Exemple

Pour renvoyer le listage LanguageFonts, on utilise la propriété LDFonts. Dans l'exemple qui suit, les codes de langue des polices configurées sont sortis :

```
Sub ShowLanguageFont()
'VBA589
Dim colLanguageFonts As HMILanguageFonts
Dim objLanguageFont As HMILanguageFont
Dim objButton As HMIButton
Dim iMax As Integer
Dim iAnswer As Integer
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
Set colLanguageFonts = objButton.LDFonts
iMax = colLanguageFonts.Count
For Each objLanguageFont In colLanguageFonts
iAnswer = MsgBox("Projected fonts: " & iMax & vbCrLf & "Language-ID: " &
objLanguageFont.LanguageID, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objLanguageFont
End Sub
```

Voir aussi

Objet StaticText (Page 2017)
Objet OptionGroup (Page 1977)
Objet LanguageFonts (liste) (Page 1951)
Objet CheckBox (Page 1889)
Objet Button (Page 1886)

Propriété LDLabelTexts

Description

Renvoie un listage contenant les textes (en langue étrangère) du menu défini par l'utilisateur ou de l'option de menu.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Supprimer objets" et d'ajouter deux options de menu ("Supprimer rectangle" et "Supprimer cercle"). Dans cet exemple, des textes de menu en langue étrangère sont créés :

```
Sub CreateMenuItem()
'VBA590
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objLangText As HMILanguageText
'
'Add new menu "Delete objects" to menubar:
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete objects")
'
'Add two menuitems to the new menu
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
rectangles")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete circles")
'
'Define foreign-language labels for menu "Delete objects":
Set objLangText = objMenu.LDLabelTexts.Add(1033, "English_Delete objects")
Set objLangText = objMenu.LDLabelTexts.Add(1032, "Greek_Delete objects")
Set objLangText = objMenu.LDLabelTexts.Add(1034, "Spanish_Delete objects")
Set objLangText = objMenu.LDLabelTexts.Add(1036, "French_Delete objects")
End Sub
```

La procédure "LDLabelInfo()" permet de supprimer les textes configurés pour le menu "Supprimer objets" :

```
Sub LDLabelInfo()
'VBA591
Dim colLangTexts As HMILanguageTexts
```

```
Dim objLangText As HMILanguageText
Dim iAnswer As Integer
'
'Save all labels of menu into collection "colLangTexts":
Set colLangTexts = ActiveDocument.CustomMenus("DeleteObjects").LDLabelTexts
For Each objLangText In colLangTexts
iAnswer = MsgBox(objLangText.DisplayName, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objLangText
End Sub
```

Voir aussi

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Propriété LDNames

Description

Renvoie un listage contenant les désignations en langue étrangère d'un classeur de la bibliothèque de modules ou d'un niveau.

Exemple

Utilisez la propriété LDNames pour renvoyer le listage LanguageTexts. Dans l'exemple suivant, sont sortis toutes les désignations de niveaux en langue étrangère :

Explication : ce que montre l'exemple

```
Sub LDLabelInfo()
'VBA592
Dim colLayerLngTexts As HMILanguageTexts
Dim objLayerLngText As HMILanguageText
Dim iIndex As Integer
Dim iAnswer As Integer
Dim strResult As String
iIndex = 1
For iIndex = 1 To ActiveDocument.Layers.Count
'
'Save all labels of layers into collection of "colLayerLngTexts":
Set colLayerLngTexts = ActiveDocument.Layers(iIndex).LDNames
For Each objLayerLngText In colLayerLngTexts
strResult = strResult & vbCrLf & objLayerLngText.LanguageID & " - " &
objLayerLngText.DisplayName
Next objLayerLngText
iAnswer = MsgBox(strResult, vbOKCancel)
strResult = ""
If vbCancel = iAnswer Then Exit For
```

```
Next iIndex
End Sub
```

Voir aussi

Objet Layer (Page 1955)

Objet LanguageTexts (liste) (Page 1954)

Objet FolderItem (Page 1927)

Propriété LDStatusTexts

Description

Renvoie un listage qui contient les textes (en langue étrangère) des lignes d'état d'une icône ou d'une option de menu personnalisées.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Delete Objects" et d'ajouter deux options ("Delete Rectangles" et "Delete Circles") : Dans cet exemple, des textes de lignes d'état en langue étrangère sont créés :

```
Sub CreateMenuItem()
'VBA593
Dim objMenu As HMIMenu
Dim objMenuItem1 As HMIMenuItem
Dim objMenuItem2 As HMIMenuItem
Dim objLangStateText As HMILanguageText
'
'Add new menu "Delete objects" to menubar:
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete objects")
'
'Add two menuitems to the new menu
Set objMenuItem1 = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
rectangles")
Set objMenuItem2 = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete
circles")
'
'Define foreign-language labels for menuitem "Delete rectangles":
Set objLangStateText = objMenuItem1.LDStatusTexts.Add(1033, "English_Delete rectangles")
Set objLangStateText = objMenuItem1.LDStatusTexts.Add(1032, "Greek_Delete rectangles")
Set objLangStateText = objMenuItem1.LDStatusTexts.Add(1034, "Spanish_Delete rectangles")
Set objLangStateText = objMenuItem1.LDStatusTexts.Add(1036, "French_Delete rectangles")
End Sub
```

La procédure "LDStatusTextInfo()" permet de sortir les textes configurés de lignes d'état pour le menu "Delete Objects" :

```
Sub LDStatusTextInfo()
'VBA594
Dim colMenuItems As HMIMenuItems
Dim objMenuItem As HMIMenuItem
Dim colStatusLngTexts As HMILanguageTexts
Dim objStatusLngText As HMILanguageText
Dim strResult As String
Dim iAnswer As Integer
Set colMenuItems = ActiveDocument.CustomMenus("DeleteObjects").MenuItems
For Each objMenuItem In colMenuItems
strResult = "Statustexts of menuitem """ & objMenuItem.Label & """"
Set colStatusLngTexts = objMenuItem.LDStatusTexts
For Each objStatusLngText In colStatusLngTexts
strResult = strResult & vbCrLf & objStatusLngText.DisplayName
Next objStatusLngText
iAnswer = MsgBox(strResult, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objMenuItem
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Propriété LDTexts

Description

Renvoie un listage contenant les textes (en langue étrangère) d'un objet.

Exemple

La procédure "LDTextInfo()" permet de sortir les textes configurés pour l'objet Button. Afin que cet exemple fonctionne, créez dans Graphics Designer l'objet "monBouton" et configurez plusieurs textes en langue étrangère :

```
Sub LDTextInfo()
'VBA595
Dim colLDLngTexts As HMILanguageTexts
Dim objLDLngText As HMILanguageText
Dim objButton As HMIButton
Dim iAnswer As Integer
Set objButton = ActiveDocument.HMIObjects("myButton")
Set colLDLngTexts = objButton.LDTexts
For Each objLDLngText In colLDLngTexts
```

```
iAnswer = MsgBox(objLDLngText.DisplayName, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objLDLngText
End Sub
```

Voir aussi

Objet Button (Page 1886)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet CheckBox (Page 1889)

Propriété LDTooltipTexts

Description

Renvoie un listage contenant les textes d'info-bulle (en langue étrangère) de l'icône personnalisée ou de l'objet.

Exemple

La procédure "CreateToolbar()" crée une barre d'outils personnalisée à deux icônes. Deux textes d'info-bulle en langue étrangère sont attribués à la première icône :

```
Sub CreateToolbar()
'VBA596
Dim objToolbar As HMIToolbar
Dim objToolbarItem As HMIToolbarItem
Dim objLangText As HMILanguageText
Dim strFileWithPath
'
'Create toolbar with two toolbar-items:
Set objToolbar = ActiveDocument.CustomToolbars.Add("Tool1_1")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(1, "ti1_1",
"myFirstToolbaritem")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(2, "ti1_2",
"mySecondToolbaritem")
'
'In order that the example runs correct copy a *.ICO-Graphic
'into the "GraCS"-Folder of the actual project.
'Replace the filename "EZSTART.ICO" in the next commandline
'with the name of the ICO-Graphic you copied
strFileWithPath = Application.ApplicationDataPath & "EZSTART.ICO"
'
'
'To assign the symbol-icon to the first toolbaritem
objToolbar.ToolbarItems(1).Icon = strFileWithPath
'
```

```
'Define foreign-language tooltiptexts
Set objLangText = objToolbar.ToolbarItems(1).LDTooltipTexts.Add(1036, "French_Tooltiptext")
Set objLangText = objToolbar.ToolbarItems(1).LDTooltipTexts.Add(1034,
"Spanish_Tooltiptext")
End Sub
```

La procédure "LDTooltipInfo()" permet de sortir tous les textes d'info-bulle configurés pour la première icône de la première barre d'outils personnalisée :

```
Sub LDTooltipInfo()
'VBA597
Dim colLangTexts As HMILanguageTexts
Dim objLangText As HMILanguageText
Dim iAnswer As Integer
Set colLangTexts = ActiveDocument.CustomToolbars(1).ToolbarItems(1).LDTooltipTexts
For Each objLangText In colLangTexts
iAnswer = MsgBox(objLangText.DisplayName, vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next objLangText
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet HMIObject (Page 1943)

Propriété Left

Description

Définit ou renvoie la coordonnée x de l'objet (mesurée à partir du bord supérieur gauche de la vue) en pixels. La coordonnée X se réfère à l'angle gauche supérieur du rectangle entourant l'objet.

Objet View

Définit ou renvoie la coordonnée x de la fenêtre (mesurée à partir du bord supérieur gauche de l'espace de travail de Graphics Designers) en pixels.

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés du rectangle. Dans cet exemple, le rectangle est déplacé de 40 pixels vers la droite :

```
Sub RectangleConfiguration()
'VBA598
Dim objRectangle As HMIRectangle
```

```
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.Left = 40
End With
End Sub
```

Voir aussi

Objet View (Page 2050)

Objet HMIObject (Page 1943)

Propriété LeftComma

Description

Définit ou renvoie le nombre de chiffres avant la virgule (0 à 20) de l'objet BarGraph.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, le nombre de chiffres avant la virgule est défini sur "4" :

```
Sub BarGraphConfiguration()
'VBA599
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.LeftComma = 4
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété LightEffect

Description

TRUE si l'effet de lumière de l'objet 3DBarGraph est activé. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, l'effet de lumière est activé :

```
Sub HMI3DBarGraphConfiguration()
'VBA600
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
.LightEffect = True
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété LimitHigh4

Description

Définit ou renvoie pour l'objet BarGraph le seuil supérieur pour la "Réserve 4".

La propriété CheckLimitHigh4 doit être positionnée sur TRUE pour que le seuil "Reserve 4" puisse être surveillé.

Déterminez avec la propriété "TypeLimitHigh4" le type d'exploitation (en pour cent ou absolu).

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "70".

```
Sub BarGraphLimitConfiguration()
'VBA601
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitHigh4 = False
'Activate monitoring
.CheckLimitHigh4 = True
'Set barcolor to "red"
.ColorLimitHigh4 = RGB(255, 0, 0)
'Set upper limit to "70"
.LimitHigh4 = 70
End With
```

```
End Sub
```

Voir aussi

Propriété TypeLimitHigh4 (Page 2380)

Propriété CheckLimitHigh4 (Page 2117)

Objet BarGraph (Page 1881)

Propriété LimitHigh5

Description

Définit ou renvoie pour l'objet BarGraph le seuil supérieur pour la "Réserve 5".

La propriété CheckLimitHigh5 doit être positionnée sur TRUE pour que le seuil "Reserve 5" puisse être surveillé.

Déterminez avec la propriété "TypeLimitHigh5" le type d'exploitation (en pour cent ou absolu).

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "80".

```
Sub BarGraphLimitConfiguration()
'VBA602
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitHigh5 = False
'Activate monitoring
.CheckLimitHigh5 = True
'Set barcolor to "black"
.ColorLimitHigh5 = RGB(0, 0, 0)
'Set upper limit to "80"
.LimitHigh4 = 80
End With
End Sub
```

Voir aussi

Propriété TypeLimitHigh5 (Page 2381)

Propriété CheckLimitHigh5 (Page 2118)

Objet BarGraph (Page 1881)

Propriété LimitLow4

Description

Définit ou renvoie pour l'objet BarGraph le seuil inférieur pour la "Réserve 4".

La propriété CheckLimitLow4 doit être positionnée sur TRUE pour que le seuil "Reserve 4" puisse être surveillé.

Déterminez avec la propriété "TypeLimitLow4" le type d'exploitation (en pour cent ou absolu).

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "5".

```
Sub BarGraphLimitConfiguration()
'VBA603
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitLow4 = False
'Activate monitoring
.CheckLimitLow4 = True
'Set barcolor to "green"
.ColorLimitLow4 = RGB(0, 255, 0)
'Set lower limit to "5"
.LimitLow4 = 5
End With
End Sub
```

Voir aussi

Propriété CheckLimitLow4 (Page 2119)

Propriété TypeLimitLow4 (Page 2382)

Objet BarGraph (Page 1881)

Propriété LimitLow5

Description

Définit ou renvoie pour l'objet BarGraph le seuil inférieur pour la "Réserve 5".

La propriété CheckLimitLow5 doit être positionnée sur TRUE pour que le seuil "Reserve 5" puisse être surveillé.

Déterminez avec la propriété "TypeLimitLow5" le type d'exploitation (en pour cent ou absolu).

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "0".

```
Sub BarGraphLimitConfiguration()
'VBA604
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis to absolute
.TypeLimitLow5 = False
'Activate monitoring
.CheckLimitLow5 = True
'Set barcolor to "white"
.ColorLimitLow5 = RGB(255, 255, 255)
'Set lower limit to "0"
.LimitLow5 = 0
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété TypeLimitLow5 (Page 2383)

Propriété CheckLimitLow5 (Page 2119)

Propriété LimitMax

Description

Définit ou renvoie pour l'objet IOField le seuil supérieur sous forme de valeur absolue en fonction du format de données.

Si la valeur à afficher dépasse le seuil supérieur, elle est identifiée par une série de *** comme étant non représentable.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le seuil supérieur pour une valeur décimale est défini sur "100" :

```
Sub IOFieldConfiguration()
'VBA605
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
```

```
.DataFormat = 1
.LimitMax = 100
End With
End Sub
```

Voir aussi

Propriété DataFormat (Page 2149)

Objet IOField (Page 1947)

Propriété LimitMin

Description

Définit ou renvoie pour l'objet IOField le seuil inférieur sous forme de valeur absolue en fonction du format de données.

Si la valeur à afficher dépasse le seuil supérieur, elle est identifiée par une série de *** comme étant non représentable.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le seuil supérieur pour une valeur décimale est défini sur "0" :

```
Sub IOFieldConfiguration()
'VBA606
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.DataFormat = 1
.LimitMin = 0
End With
End Sub
```

Voir aussi

Propriété DataFormat (Page 2149)

Objet IOField (Page 1947)

Propriété LineJoinStyle

Description

Définit de quelle manière les angles d'un polygone tuyau sont représentés.

carré Les tuyaux sont reliés entre eux sans arrondi au niveau des angles.
rond Les tuyaux sont arrondis au niveau des angles, côté extérieur.

Exemple

Propriété ListType

Description

Définit ou renvoie pour l'objet TextList le type de liste. Plage des valeurs de 0 à 2.

Type de liste	Valeur affectée
Décimal	0
Binaire	1
Bit	2

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, le type de liste est défini valeur de "Décimal" :

```
Sub TextListConfiguration()
'VBA607
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.ListType = 0
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété LockBackColor

Description

Définit ou renvoie pour l'objet GroupDisplay la couleur d'arrière-plan du bouton pour une position de mesure verrouillée. LONG Accès en écriture et en lecture.

La propriété LockStatus doit avoir la valeur TRUE pour que la couleur d'arrière-plan puisse s'afficher.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour une position de mesure verrouillée est réglée sur "Rouge" :

```
Sub GroupDisplayConfiguration()
'VBA608
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.LockStatus = True
.LockBackColor = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Propriété LockStatus (Page 2263)

Objet GroupDisplay (Page 1935)

Propriété LockedByCreatorID

Description

VRAI si une vue a été créée et/ou référencée par le gestionnaire SIMATIC. Accès en lecture BOOLEAN.

Si une vue a été créée dans le gestionnaire SIMATIC, vous pouvez l'éditer et ensuite l'enregistrer ultérieurement dans WinCC. Vous ne pouvez toutefois pas supprimer la vue dans

WinCC. Le gestionnaire SIMATIC gère à cet effet pour chaque vue un indicatif appelé CreatorID, qu'il est impossible de modifier dans WinCC.

Vous pouvez éditer la vue dans WinCC, mais il est impossible de l'écraser par une vue WinCC (LockedByCreatorID = FAUX). Il est recommandé de vérifier avec la méthode SaveAs la valeur de la propriété LockedByCreatorID d'un fichier existant, et ce avant l'écriture. Si une vue de ce type est enregistrée avec la méthode SaveAs dans une vue WinCC nouvelle (ou encore inexistante) ou existant déjà, CreatorID n'accompagne pas le transfert.

Exemple 1

Dans l'exemple suivant, une vue créée avec le gestionnaire SIMATIC (LockedByCreatorID = VRAI) est ouverte, éditée et enregistrée. La valeur de la propriété LockedByCreatorID n'est pas modifiée.

```
Sub SaveDocAs_1()
'VBA810
'open an existing file, change it and save it
Dim docOld As Document
Const strFile As String = "Simatic_001.Pdl"
'
Set docOld = Application.Documents.Open(Application.ApplicationDataPath & strFile,
hmiOpenDocumentTypeInvisible)
docOld.Width = docOld.Width + 1
docOld.Save
'
MsgBox "LockedByCreatorID = " & docOld.LOckedByCreatorID, vbOKOnly, "Result"
'
docOld.Close
Set docOld = Nothing
'
End Sub
```

Exemple 2

Dans cet exemple, une nouvelle vue est enregistrée avec la méthode SaveAs en tant que nouveau fichier. La propriété LockedByCreatorID est vérifiée pour déterminer si la vue peut être enregistrée. La propriété LockedByCreator est réinitialisée dans le nouveau fichier.

```
Sub SaveDocAs_2()
'VBA811
'create a new file and overwrite it to an existing file,
'if it is not 'locked by CreatorID'
Dim docNew As Document
Dim docOld As Document
Const strFile As String = "Simatic_001.Pdl"
'
Set docNew = Application.Documents.Add(hmiOpenDocumentTypeInvisible)
Set docOld = Application.Documents.Open(Application.ApplicationDataPath & strFile,
hmiOpenDocumentTypeInvisible)
'
```

```
If docOld.LockedByCreatorID = False Then
docOld.Close
docNew.SaveAs(Application.ApplicationDataPath & strFile)
    Else
MsgBox "File cannot be stored (LockedByCreatorID). ", vbOKOnly, "Result"
    End If
'
docOld.Close
docNew.Close
Set docOld = Nothing
Set docNew = Nothing
'
End Sub
```

Voir aussi

Méthode SaveAs (Page 1854)

Objet Document (Page 1908)

Propriété LockStatus

Description

TRUE si une position de mesure verrouillée doit être affichée pour l'objet GroupDisplay. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour une position de mesure verrouillée est réglée sur "Rouge" :

```
Sub GroupDisplayConfiguration()
'VBA609
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.LockStatus = True
.LockBackColor = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété LockText

Description

Définit pour l'objet GroupDisplay le texte du bouton pour une position de mesure verrouillée.

La propriété LockStatus doit avoir la valeur TRUE pour que le texte puisse s'afficher.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, le texte d'une position de mesure verrouillée est défini valeur de "Désactivé" :

```
Sub GroupDisplayConfiguration()
'VBA610
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.LockStatus = True
.LockText = "verrouillé"
End With
End Sub
```

Voir aussi

Propriété LockStatus (Page 2263)

Objet GroupDisplay (Page 1935)

Propriété LockTextColor

Description

Définit ou renvoie pour l'objet GroupDisplay la couleur du texte du bouton pour une position de mesure verrouillée. LONG Accès en écriture et en lecture.

La propriété LockStatus doit avoir la valeur TRUE pour que la couleur d'arrière-plan puisse s'afficher.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur du texte du bouton pour une position de mesure verrouillée est réglée sur "Jaune" :

```
Sub GroupDisplayConfiguration()
'VBA611
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.LockStatus = True
.LockTextColor = RGB(0, 255, 255)
End With
End Sub
```

Voir aussi

Propriété LockStatus (Page 2263)

Objet GroupDisplay (Page 1935)

Propriété LongStrokesBold

Description

TRUE si les sections longues doivent être affichées en gras lors de la représentation de l'échelle de l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, les grandes divisions ne doivent pas être affichées en gras :

```
Sub BarGraphConfiguration()
'VBA612
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.LongStrokesBold = False
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété LongStrokesOnly

Description

TRUE si seules les sections longues doivent être affichées lors de la représentation de l'échelle de l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, seules les sections longues doivent être affichées :

```
Sub BarGraphConfiguration()
'VBA613
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.LongStrokesOnly = True
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété LongStrokesSize

Description

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph.

Exemple

Dans cet exemple, la longueur des sections d'axe est définie sur "10" :

```
Sub BarGraphConfiguration()
'VBA614
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.LongStrokesSize = 10
End With
End Sub
```

Voir aussi

Propriété AxisSection (Page 2075)

Objet BarGraph (Page 1881)

Propriété LongStrokesTextEach

Description

Définit ou renvoie les sections devant être étiquetées lors de la représentation de l'échelle de l'objet BarGraph (1 = chaque section, 2 = toutes les deux sections, etc.).

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, une étiquette est apposée toutes les trois sections :

```
Sub BarGraphConfiguration()
'VBA615
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.LongStrokesTextEach = 3
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

M

Propriété Macro

Description

Définit pour une option de menu personnalisée ou une icône la macro VBA devant être exécutée à la sélection.

Exemple

Dans l'exemple qui suit, un menu définie par l'utilisateur est créé avec deux options qui appellent deux macros VBA différentes :

```
Sub CreateDocumentMenusUsingMacroProperty()
'VBA616
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "My first menuitem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "My second menuitem")
'
'To assign a macro to every menuitem:
With ActiveDocument.CustomMenus("DocMenu1")
.MenuItems("dmItem1_1").Macro = "TestMacro1"
.MenuItems("dmItem1_2").Macro = "TestMacro2"
End With
End Sub
```

Vous pouvez appeler ces deux procédures à l'aide des options de menu du menu défini par l'utilisateur DocMenu1 :

```
Sub TestMacro1()
MsgBox "TestMacro1 is executed"
End Sub
```

```
Sub TestMacro2()
MsgBox "TestMacro2 is executed"
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Pour affecter des macros VBA à des menus et à des barres d'outils (Page 1636)

Propriété Marker

Description

TRUE si les seuils doivent être affichés sous forme de valeur d'échelle pour l'objet BarGraph. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, les seuils sont affichés sous forme de valeurs d'échelle :

```
Sub BarGraphConfiguration()
'VBA617
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Marker = True
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété Max

Description

Détermine la valeur absolue en cas d'affichage maximal des valeurs ou la restitue.

Cette valeur s'affiche si la graduation est active.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la valeur absolue est définie sur "10" :

```
Sub BarGraphConfiguration()
'VBA618
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Max = 10
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Objet BarGraph (Page 1881)

Objet 3DBarGraph (Page 1867)

Propriété MaximizeButton

Description

TRUE si l'objet ApplicationWindow peut être agrandi dans Runtime. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ApplicationWindowConfig" permet d'accéder aux propriétés de la fenêtre d'application. Dans cet exemple, la fenêtre d'application contient un bouton Maximiser dans Runtime :

```
Sub ApplicationWindowConfig()
'VBA619
Dim objAppWindow As HMIApplicationWindow
Set objAppWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow1",
"HMIApplicationWindow")
With objAppWindow
.MaximizeButton = True
End With
End Sub
```

Voir aussi

Objet ApplicationWindow (Page 1879)

Propriété MaxZoom

Description

Renvoie le niveau de zoom maximal pour le plan.

Exemple

La procédure "LayerInfo()" permet de sortir pour chaque plan de la vue actuelle les noms et les configurations de zoom définies :

```
Sub LayerInfo()
'VBA620
Dim colLayers As HMILayers
Dim objSingleLayer As HMILayer
Dim iAnswer As Integer
Set colLayers = ActiveDocument.Layers
For Each objSingleLayer In colLayers
With objSingleLayer
```

```
iAnswer = MsgBox("Layername: " & .Name & vbCrLf & "Min. zoom:  " & .MinZoom & vbCrLf & "Max.
zoom:  " & .MaxZoom, vbOKCancel)
End With
If vbCancel = iAnswer Then Exit For
Next objSingleLayer
End Sub
```

Voir aussi

Objet Layer (Page 1955)

Edition de niveaux avec VBA (Page 1650)

Propriété MCGUBackColorOff

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie non acquittée" la couleur de l'arrière-plan pour l'état clignotant "Désactivé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour l'état clignotant "Désactivé" est réglée sur "Rouge" :

```
Sub GroupDisplayConfiguration()
'VBA621
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCGUBackColorOff = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCGUBackColorOn

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie non acquittée" la couleur de l'arrière-plan pour l'état clignotant "Activé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour l'état clignotant "Activé" est réglée sur "Blanc" :

```
Sub GroupDisplayConfiguration()
'VBA622
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCGUBackColorOn = RGB(255, 255, 255)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCGUBackFlash

Description

TRUE si l'arrière-plan de l'objet GroupDisplay doit clignoter à la sortie non acquittée d'une alarme. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan clignote lorsqu'une alarme n'est pas acquittée :

```
Sub GroupDisplayConfiguration()
'VBA623
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCGUBackFlash = True
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCGUTextColorOff

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie non acquittée" la couleur du texte pour l'état clignotant "Désactivé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan du texte pour l'état clignotant "Désactivé" est réglée sur "Bleu" :

```
Sub GroupDisplayConfiguration()
'VBA624
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCGUTextColorOff = RGB(0, 0, 255)
End With
```

```
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCGUTextColorOn

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie non acquittée" la couleur d'arrière-plan du texte pour l'état clignotant "Activé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan du texte pour l'état clignotant "Activé" est réglée sur "Noir" :

```
Sub GroupDisplayConfiguration()
'VBA625
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCGUTextColorOn = RGB(0, 0, 0)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCGUTextFlash

Description

TRUE si la police de l'objet GroupDisplay doit clignoter à la sortie non acquittée d'une alarme. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la police clignote lorsqu'une alarme n'est pas acquittée :

```
Sub GroupDisplayConfiguration()
'VBA626
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCGUTextFlash = True
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKOBackColorOff

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Arrivée" la couleur d'arrière-plan pour l'état clignotant "Désactivé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour l'état clignotant "Désactivé" est réglée sur "Rouge" :

```
Sub GroupDisplayConfiguration()
'VBA627
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKOBackColorOff = RGB(255, 0, 0)
End With
```

```
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKOBackColorOn

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Arrivée" la couleur d'arrière-plan pour l'état clignotant "Activé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour l'état clignotant "Activé" est réglée sur "Blanc" :

```
Sub GroupDisplayConfiguration()
'VBA628
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKOBackColorOn = RGB(255, 255, 255)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKOBackFlash

Description

TRUE si l'arrière-plan de l'objet GroupDisplay doit clignoter à l'arrivée d'une alarme. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan clignote lorsqu'une alarme n'est pas acquittée :

```
Sub GroupDisplayConfiguration()
'VBA629
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKOBackFlash = True
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKOTextColorOff

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Arrivée" la couleur du texte pour l'état clignotant "Désactivé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan du texte pour l'état clignotant "Désactivé" est réglée sur "Bleu" :

```
Sub GroupDisplayConfiguration()
'VBA630
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKOTextColorOff = RGB(0, 0, 255)
End With
```

```
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKOTextColorOn

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Arrivée" la couleur d'arrière-plan du texte pour l'état clignotant "Activé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan du texte pour l'état clignotant "Activé" est réglée sur "Noir" :

```
Sub GroupDisplayConfiguration()
'VBA631
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKOTextColorOn = RGB(0, 0, 0)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKOTextFlash

Description

TRUE si la police de l'objet GroupDisplay doit clignoter à l'arrivée d'une alarme. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la police clignote lorsqu'une alarme n'est pas acquittée :

```
Sub GroupDisplayConfiguration()
'VBA632
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKOTextFlash = True
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKQBackColorOff

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie acquittée" la couleur de l'arrière-plan pour l'état clignotant "Désactivé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour l'état clignotant "Désactivé" est réglée sur "Rouge" :

```
Sub GroupDisplayConfiguration()
'VBA633
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKQBackColorOff = RGB(255, 0, 0)
End With
```

```
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKQBackColorOn

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie acquittée" la couleur de l'arrière-plan pour l'état clignotant "Activé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour l'état clignotant "Activé" est réglée sur "Blanc" :

```
Sub GroupDisplayConfiguration()
'VBA634
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKQBackColorOn = RGB(255, 255, 255)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKQBackFlash

Description

TRUE si l'arrière-plan de l'objet GroupDisplay doit clignoter à la sortie acquittée d'une alarme. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan clignote lorsqu'une alarme n'est pas acquittée :

```
Sub GroupDisplayConfiguration()
'VBA635
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKQBackFlash = True
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKQTextColorOff

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie acquittée" la couleur du texte pour l'état clignotant "Désactivé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan du texte pour l'état clignotant "Désactivé" est réglée sur "Bleu" :

```
Sub GroupDisplayConfiguration()
'VBA636
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKQTextColorOff = RGB(0, 0, 255)
End With
```

```
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKQTextColorOn

Description

Définit ou renvoie pour l'objet GroupDisplay pour l'état "Partie acquittée" la couleur d'arrière-plan du texte pour l'état clignotant "Activé". LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan du texte pour l'état clignotant "Activé" est réglée sur "Noir" :

```
Sub GroupDisplayConfiguration()
'VBA637
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKQTextColorOn = RGB(0, 0, 0)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCKQTextFlash

Description

TRUE si la police de l'objet GroupDisplay doit clignoter à la sortie acquittée d'une alarme. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la police clignote lorsqu'une alarme n'est pas acquittée :

```
Sub GroupDisplayConfiguration()
'VBA638
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MCKQTextFlash = True
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété MCText

Description

Définit ou renvoie pour l'objet GroupDisplay le texte pour la classe d'alarmes correspondante.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, le texte "Alarm High" est défini pour la classe d'alarmes Alarm High :

```
Sub GroupDisplayConfiguration()
'VBA639
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MessageClass = 0
.MCText = "Alarm High"
End With
End Sub
```

Voir aussi

Propriété MessageClass (Page 2286)

Objet GroupDisplay (Page 1935)

Propriété MenuItems

Description

Renvoie un listage contenant toutes les options de menu du menu défini par l'utilisateur.

Exemple

La procédure "CreateMenuItem()" permet de créer le menu "Delete Objects" et d'ajouter deux options ("Delete Rectangles" et "Delete Circles") : Dans cet exemple, les textes sont sortis :

```
Sub CreateMenuItem()
'VBA640
Dim objMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim iIndex As Integer
iIndex = 1
'
'Add new menu "Delete objects" to the menubar:
Set objMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DeleteObjects", "Delete objects")
'
'Add two menuitems to menu "Delete objects"
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(1, "DeleteAllRectangles", "Delete
rectangles")
Set objMenuItem = objMenu.MenuItems.InsertMenuItem(2, "DeleteAllCircles", "Delete circles")
'
'Output label of menu:
MsgBox ActiveDocument.CustomMenus(1).Label
'
'Output labels of all menuitems:
For iIndex = 1 To objMenu.MenuItems.Count
MsgBox objMenu.MenuItems(iIndex).Label
Next iIndex
End Sub
```

Voir aussi

Objet Menu (Page 1964)

Objet MenuItem (Page 1967)

Propriété MenuItemType

Description

Renvoie le type d'une option de menu personnalisée. Accès en lecture.

Valeur renvoyée	Type de l'option de menu
0	Tiret (Separator)
1	Sous-menu (SubMenu)
2	Option de menu (MenuItem)

Exemple

La procédure "ShowMenuTypes()" permet de sortir les types d'options de menu pour le premier menu défini par l'utilisateur :

```
Sub ShowMenuTypes()
'VBA641
Dim iMaxMenuItems As Integer
Dim iMenuItemType As Integer
Dim strMenuItemType As String
Dim iIndex As Integer
iMaxMenuItems = ActiveDocument.CustomMenus(1).MenuItems.Count
For iIndex = 1 To iMaxMenuItems
iMenuItemType = ActiveDocument.CustomMenus(1).MenuItems(iIndex).MenuItemType
Select Case iMenuItemType
Case 0
 strMenuItemType = "Trait de séparation (Separator)"
Case 1
 strMenuItemType = "Sous-menu (SubMenu)"
Case 2
 strMenuItemType = "Option de menu (MenuItem)"
End Select
MsgBox iIndex & ". Menuitemtype: " & strMenuItemType
Next iIndex
End Sub
```

Voir aussi

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Propriété MessageClass

Description

Définit pour l'objet GroupDisplay le type d'alarme correspondant (Alarm High, Alarm Low, Warnung High, Warnung Low, ...) pour lequel sont configurés les paramètres d'attribut "Texte affiché", "Arrivée", "Arrivée acquitté" et "Partie non acquittée".

MessageClass	Valeur affectée
AlarmHigh	0
AlarmLow	1
WarningHigh	2
WarningLow	3
Tolérance supérieure	4
Tolérance inférieure	5
Contrôle de process AS, incident	6
Contrôle de process AS, erreur	7
Contrôle de process OS, incident	8

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, la couleur d'arrière-plan pour l'état "Désactivé" du type d'alarme "AlarmHigh" est définie valeur de "Rouge" :

```
Sub GroupDisplayConfiguration()
'VBA642
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.MessageClass = 0
.MCGUBackColorOff = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété Min

Description

Détermine la valeur absolue en cas d'affichage minimal des valeurs ou la restitue.

Cette valeur s'affiche si la graduation est active.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la valeur absolue est définie sur "1" :

```
Sub BarGraphConfiguration()
'VBA643
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Min = 1
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Objet BarGraph (Page 1881)

Objet 3DBarGraph (Page 1867)

Propriété MinZoom

Description

Renvoie le niveau de zoom minimal pour le plan.

Exemple

La procédure "LayerInfo()" permet de sortir pour chaque plan de la vue actuelle les noms et les configurations de zoom définies :

```
Sub LayerInfo()
'VBA644
Dim colLayers As HMILayers
Dim objLayer As HMILayer
Dim strMaxZoom As String
Dim strMinZoom As String
Dim strLayerName As String
Dim iAnswer As Integer
Set colLayers = ActiveDocument.Layers
For Each objLayer In colLayers
```

```
With objLayer
strMinZoom = .MinZoom
strMaxZoom = .MaxZoom
strLayerName = .Name
iAnswer = MsgBox("Layername: " & strLayerName & vbCrLf & "Min. zoom:  " & strMinZoom &
vbCrLf & "Max. zoom:  " & strMaxZoom, vbOKCancel)
End With
If vbCancel = iAnswer Then Exit For
Next objLayer
End Sub
```

Voir aussi

Objet Layer (Page 1955)

Edition de niveaux avec VBA (Page 1650)

Propriété Modified

Description

TRUE si le code source d'un script ou d'une vue a été modifié. Accès en lecture BOOLEAN.

Exemple

Dans l'exemple qui suit, on vérifie si la vue active a été modifiée :

```
Sub CheckModificationOfActiveDocument()
'VBA645
Dim strCheck As String
Dim bModified As Boolean
bModified = ActiveDocument.Modified
Select Case bModified
Case True
strCheck = "Active document is modified"
Case False
strCheck = "Active document is not modified"
End Select
MsgBox strCheck
End Sub
```

Voir aussi

Objet ScriptInfo (Page 2009)

Objet Document (Page 1908)

Propriété Moveable

Description

TRUE si les objets ApplicationWindow et PictureWindow peuvent être déplacés dans Runtime. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ApplicationWindowConfig" permet d'accéder aux propriétés de la fenêtre d'application. Dans cet exemple, la fenêtre d'application doit pouvoir être déplacée dans Runtime :

```
Sub ApplicationWindowConfig()
'VBA646
Dim objAppWindow As HMIApplicationWindow
Set objAppWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow1",
"HMIApplicationWindow")
With objAppWindow
.Moveable = True
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Objet ApplicationWindow (Page 1879)

N-O

Propriété Name

Description

Renvoie le nom de l'objet. Accès en lecture STRING.

Exemple

La procédure "LayerInfo()" permet de sortir pour chaque plan de la vue actuelle les noms et les configurations de zoom définies :

```
Sub LayerInfo()
'VBA647
Dim colLayers As HMILayers
Dim objLayer As HMILayer
```

```
Dim strMaxZoom As String
Dim strMinZoom As String
Dim strLayerName As String
Dim iAnswer As Integer
Set colLayers = ActiveDocument.Layers
For Each objLayer In colLayers
With objLayer
strMinZoom = .MinZoom
strMaxZoom = .MaxZoom
strLayerName = .Name
iAnswer = MsgBox("Layername: " & strLayerName & vbCrLf & "Min. zoom:  " & strMinZoom &
vbCrLf & "Max. zoom:  " & strMaxZoom, vbOKCancel)
End With
If vbCancel = iAnswer Then Exit For
Next objLayer
End Sub
```

Voir aussi

Objet Trigger (Page 2035)

Objet SymbolLibrary (Page 2023)

Objet Property (Page 1993)

Objet HMIObject (Page 1943)

Objet Layer (Page 1955)

Objet FolderItem (Page 1927)

Objet Document (Page 1908)

Objet Application (Page 1876)

Propriété Name (FolderItem)

Description

Renvoie le nom interne de l'objet indiqué du type "FolderItem". Accès en lecture.

Exemple

Dans cet exemple, le nom interne de l'objet "PC" se trouvant dans la bibliothèque de module globale est sorti :

```
Sub ShowInternalNameOfFolderItem()
'VBA536
Dim objGlobalLib As HMISymbolLibrary
Set objGlobalLib = Application.SymbolLibraries(1)
MsgBox objGlobalLib.FolderItems(2).Folder(2).Folder.Item(1).Name
End Sub
```

Voir aussi

Objet FolderItem (Page 1927)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Propriété NegativeValue

Description

Utilisez la propriété BinaryResultInfo pour restaurer un objet BinaryResultInfo.

Exemple

Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on affecte une valeur de propriété à chaque plage de valeurs binaire :

```
Sub AddDynamicDialogToCircleRadiusTypeBinary()
'VBA648
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_C", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBool
.BinaryResultInfo.NegativeValue = 20
.BinaryResultInfo.PositiveValue = 40
End With
End Sub
```

Voir aussi

Référence VBA (Page 1724)

Propriété PositiveValue (Page 2321)

Objet BinaryResultInfo (Page 1884)

Propriété Number

Description

Renvoie le nombre de plans d'un objet de type "Layer". Le comptage commence à 1. Le premier plan nommé "Plan0" renvoie en conséquence la valeur "0". Accès en lecture.

Exemple

Dans cet exemple, le nom et le numéro du plan ainsi que l'index sont sortis :

```
Sub ShowLayerWithNumbers()
'VBA803
Dim colLayers As HMILayers
Dim objLayer As HMILayer
Dim iAnswer As Integer
Dim iIndex As Integer
iIndex = 1
Set colLayers = ActiveDocument.Layers
For Each objLayer In colLayers
iAnswer = MsgBox("Layername: " & objLayer & vbCrLf & "Layernumber: " & objLayer.Number &
vbCrLf & "Layersindex: " & iIndex, vbOKCancel)
iIndex = iIndex + 1
If vbCancel = iAnswer Then Exit For
Next objLayer
End Sub
```

Voir aussi

Objet Layer (Page 1955)

Propriété NumberLines

Description

TextList

Définit pour l'objet "TextList" le nombre de lignes que la liste de sélection doit contenir ou fournit la valeur en retour. Si le nombre, la taille de police et le type de police des lignes configurées ne correspondent pas aux dimensions de l'objet, la liste de sélection est complétée par une barre de défilement verticale.

Champ de combinaison et champ de liste

Définit pour les objets "Champ de combinaison" et "Champ de liste" le nombre de lignes de texte, ou fournit la valeur en retour. Vous pouvez paramétrer au maximum 32.000 lignes.

La valeur de l'attribut "Nombre de lignes" indique en même temps le seuil supérieur de l'attribut "Index" dans le groupe de propriétés "Police". La modification de la valeur entraîne les conséquences suivantes :

- Augmentation du nombre : de nouvelles lignes sont ajoutées en bas. Vous changez l'inscription standard du nouveau champ à l'aide de l'attribut "Texte" dans le groupe de propriétés "Police".
- Diminution du nombre : toutes les lignes, pour lesquelles la valeur de l'attribut "Index" est supérieure au nouveau nombre, sont supprimées.

Exemple

La procédure "TextListConfiguration()" accède aux propriétés de l'objet "TextList". Dans cet exemple, une liste de sélection est créée et le nombre de lignes visibles est défini sur trois :

```
Sub TextListConfiguration()
'VBA649
Dim objTextList As HMITextList
'
'Insérer une nouvelle Textliste dans la vue présente :
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.NumberLines=3
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété ObjectName

Description

Définit ou renvoie en fonction du type des objet cible et source sélectionnés de la liaison directe le nom des constantes, de l'objet ou des variables.

Les deux tableaux vous indiquent quand utiliser la propriété ObjectName. Un "--" signifie qu'une chaîne vide (" ") est affectée à la propriété lors de la création de l'objet DirectConnection.

Type d'objet source (propriété SourceLink)

Propriété Type	Propriété AutomationName	Propriété ObjectName
hmiSourceTypeConstant	--	Nom des constantes (par ex. nom de la vue)
hmiSourceTypeProperty	Propriété de l'objet source (par ex. "Top")	Nom de l'objet source (par ex. "Rectangle_A")
hmiSourceTypePropertyOfThisObject	--	--
hmiSourceTypeVariableDirect	--	Nom de variable
hmiSourceTypeVariableIndirect	--	Nom de variable

Type d'objet cible (propriété DestinationLink)

Propriété Type	Propriété AutomationName	Propriété ObjectName
hmiDestTypeProperty	Propriété de l'objet cible (par ex. "Left")	Nom de l'objet cible (par ex. "Rectangle_A")
hmiDestTypePropertyOfThisObject	--	--

Propriété Type	Propriété AutomationName	Propriété ObjectName
hmiDestTypePropertyOfActualWindow	Propriété de l'objet cible (par ex. "Left")	--
hmiDestTypeVariableDirect	--	Nom de variable
hmiDestTypeVariableIndirect	--	Nom de variable
hmiDestTypeDirectMessage	--	Nom de variable
hmiDestTypeIndirectMessage	--	Nom de variable

Exemple

Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub DirectConnection()
'VBA650
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDirConnection As HMIDirectConnection
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 100
.Text = "SetPosition"
End With
'
'Directconnection is initiated by mouseclick:
Set objDirConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDirConnection
'Sourceobject: Property "Top" of Rectangle_A
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Targetobject: Property "Left" of Rectangle_B
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
```

```
End With
End Sub
```

Voir aussi

Propriété Type (Page 2378)

Propriété SourceLink (Page 2356)

Propriété DestinationLink (Page 2150)

Propriété AutomationName (Page 2070)

Objet SourceLink (Page 2016)

Objet DestLink (Page 1904)

Propriété ObjectSizeDecluttering

Description

VRAI si les objets de la vue indiquée doivent être masqués hors de deux tailles configurées. BOOLEAN Accès en écriture et en lecture.

Définissez la zone de taille avec la méthode SetDeclutterObjectSize.

Exemple

Dans l'exemple qui suit, on définit les paramètres du niveau le plus bas de la vue active :

```
Sub ConfigureSettingsOfLayer
'VBA651
Dim objLayer As HMILayer
Set objLayer = ActiveDocument.Layers(1)
With objLayer
'Configure "Layer 0"
.MinZoom = 10
.MaxZoom = 100
.Name = "Configured with VBA"
End With
'Define fade-in and fade-out of objects:
With ActiveDocument
.LayerDecluttering = True
.ObjectSizeDecluttering = True
.SetDeclutterObjectSize 50, 100
End With
End Sub
```

Voir aussi

Objet Document (Page 1908)

Edition de niveaux avec VBA (Page 1650)

Propriété OffsetLeft

Description

Détermine la distance entre la vue et le bord gauche de la fenêtre de vue, ou la restitue.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA652
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété OffsetTop

Description

Détermine la distance entre la vue et le bord supérieur de la fenêtre de vue, ou la restitue.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA653
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété OnTop

Description

TRUE si l'objet ApplicationWindow reste toujours au premier plan dans Runtime. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ApplicationWindowConfig" permet d'accéder aux propriétés de la fenêtre d'application. Dans cet exemple, la fenêtre d'application doit toujours rester au premier plan dans Runtime :

```
Sub ApplicationWindowConfig()
'VBA654
Dim objAppWindow As HMIApplicationWindow
Set objAppWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow1",
"HMIApplicationWindow")
```

```
With objAppWindow
.OnTop = True
End With
End Sub
```

Voir aussi

Objet ApplicationWindow (Page 1879)

Propriété Operation

Description

TRUE si l'objet peut être commandé dans Runtime. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans cet exemple, l'état des autorisations de conduite de tous les objets est sorti dans la vue active :

```
Sub ShowOperationStatusOfAllObjects()
'VBA655
Dim objObject As HMIObject
Dim bStatus As Boolean
Dim strStatus As String
Dim strName As String
Dim iMax As Integer
Dim iIndex As Integer
Dim iAnswer As Integer
iMax = ActiveDocument.HMIObjects.Count
iIndex = 1
For iIndex = 1 To iMax
strName = ActiveDocument.HMIObjects(iIndex).ObjectName
bStatus = ActiveDocument.HMIObjects(iIndex).Operation
Select Case bStatus
Case True
strStatus = "yes"
Case False
strStatus = "no"
End Select
iAnswer = MsgBox("Object: " & strName & vbCrLf & "Operator-Control enable: " & strStatus,
vbOKCancel)
If vbCancel = iAnswer Then Exit For
Next iIndex
If 0 = iMax Then MsgBox "No objects in the active document."
End Sub
```

Voir aussi

Objet HMIObject (Page 1943)

Objet Document (Page 1908)

Propriété OperationMessage

Description

TRUE, lorsqu'une alarme doit être émise après qu'une commande ait été exécutée. Le motif de la commande ne peut être entré que si la propriété "OperationReport" est définie valeur de "True". BOOLEAN Accès en écriture et en lecture.

La commande est envoyée au système d'alarme et archivée. Une alarme peut p. ex. être affichée sur une ligne d'alarme par le système d'alarme.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, une commande doit être envoyée au système d'alarme :

```
Sub IOFieldConfiguration()
'VBA656
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.OperationReport = True
.OperationMessage = True
End With
End Sub
```

Voir aussi

Propriété OperationReport (Page 2300)

Objet TextList (Page 2025)

Objet Slider (Page 2013)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet CheckBox (Page 1889)

Propriété OperationReport

Description

TRUE, lorsque la raison pour une commande doit être également listée. BOOLEAN Accès en écriture et en lecture.

Lors de la commande de l'objet dans Runtime, un dialogue apparaît dans lequel l'utilisateur peut entrer le motif de la commande sous forme de texte. La commande est envoyée au système d'alarme et archivée.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, une commande doit être envoyée au système d'alarme :

```
Sub IOFieldConfiguration()
'VBA657
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.OperationReport = True
.OperationMessage = True
End With
End Sub
```

Voir aussi

Propriété OperationMessage (Page 2299)

Objet TextList (Page 2025)

Objet Slider (Page 2013)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet CheckBox (Page 1889)

Propriété Orientation

Description

TRUE, lorsque le texte est représenté horizontalement dans l'objet. BOOLEAN Accès en écriture et en lecture.

Remarque

Seul le texte est représenté horizontalement ou verticalement. La position de l'objet reste inchangée.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, le texte est représenté verticalement :

```
Sub ButtonConfiguration()
'VBA658
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.Width = 150
.Height = 150
.Text = "Text is displayed vertical"
.Orientation = False
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet IOField (Page 1947)

Objet CheckBox (Page 1889)

Objet Button (Page 1886)

Propriété OutputFormat

Description

Définit la représentation de la valeur de sortie ou renvoie la valeur définie. La représentation dépend du format de données.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, le type de données du champ d'E/S est défini valeur de "Décimal". La valeur de sortie est affichée avec deux chiffres avant la virgule et trois chiffres après la virgule :

```
Sub IOFieldConfiguration()
'VBA659
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.DataFormat = 1
.OutputFormat = "99,999"
End With
End Sub
```

Voir aussi

Propriété DataFormat (Page 2149)

Objet IOField (Page 1947)

Propriété OutputValue

Description

Définit le préréglage pour la valeur à afficher ou le restitue .

En Runtime, cette valeur est utilisée lorsqu'au moment de l'activation de la vue la variable correspondante n'est ni raccordée, ni mise à jour.

Exemple

La procédure "IOFieldConfiguration()" permet d'accéder aux propriétés du champ d'E/S. Dans cet exemple, la valeur de sortie est définie sur "00" :

```
Sub IOFieldConfiguration()
'VBA660
Dim objIOField As HMIIOField
Set objIOField = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
With objIOField
.OutputValue = "00"
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)
Objet IOField (Page 1947)

P-Q

Propriété Parent

Description

Renvoie l'objet supérieur de l'objet indiqué. Accès en lecture.

Exemple

Dans l'exemple qui suit, la vue active est copiée et le nom de la vue est renvoyé ensuite à l'aide de la propriété Parent :

```
Sub ExampleForParent()
'VBA661
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
MsgBox objView.Parent.Name
End Sub
```

Voir aussi

Objet Toolbars (liste) (Page 2029)
Objet Menu (Page 1964)
Objet Document (Page 1908)
Objet Views (liste) (Page 2052)
Objet View (Page 2050)
Objet VariableTriggers (liste) (Page 2049)
Objet VariableTrigger (Page 2048)
Objet VariableStateValues (liste) (Page 2046)
Objet VariableStateValue (Page 2045)
Objet Trigger (Page 2035)
Objet ToolbarItems (liste) (Page 2034)
Objet ToolbarItem (Page 2031)
Objet Toolbar (Page 2028)
Objet TextList (Page 2025)
Objet SymbolLibraries (liste) (Page 2024)
Objet SymbolLibrary (Page 2023)
Objet StatusDisplay (Page 2020)
Objet StaticText (Page 2017)
Objet SourceLink (Page 2016)
Objet Slider (Page 2013)
Objet Selection (liste) (Page 2010)
Objet ScriptInfo (Page 2009)
Objet RoundRectangle (Page 2006)
Objet RoundButton (Page 2003)
Objet Rectangle (Page 2000)
Objet Properties (liste) (Page 1992)
Objet Property (Page 1993)
Objet PolyLine (Page 1989)
Objet Polygon (Page 1986)
Objet PictureWindow (Page 1980)
Objet PieSegment (Page 1983)
Objet OptionGroup (Page 1977)
Objet OLEObject (Page 1975)
Objet MenuItems (liste) (Page 1970)
Objet MenuItem (Page 1967)
Objet Menus (liste) (Page 1965)
Objet Line (Page 1958)
Objet Layers (Liste) (Page 1956)
Objet Layer (Page 1955)
Objet LanguageTexts (liste) (Page 1954)

Propriété PasswordLevel

Description

Détermine l'autorisation pour la commande (p. ex. aucune saisie ou aucun déclenchement d'action) pour l'objet.

PasswordLevel	Valeur affectée
<Aucune protection d'accès>	0
Gestion des utilisateurs	1
Saisie de valeur	2
Commande de process	3
Edition de vue	4
Changement de vue	5
Sélection de fenêtre	6
Recopie d'écran	7
Acquittement d'alarmes	8
Verrouillage d'alarmes	9
Déverrouillage d'alarmes	10
Edition d'alarme	11
Démarrer archivage	12
Arrêter archivage	13
Edition d'archives	14
Edition d'archive	15
Edition d'action	16
Gestionnaire de projet	17
Activer Remote	1000
Configurer Remote	1001
Contrôle seulement	1002

Les autorisations de conduite doivent être définies au préalable dans le User Administrator.

Exemple

--

Voir aussi

Objet HMIObject (Page 1943)

Propriété Path

Description

Renvoie le chemin complet du répertoire dans lequel la vue indiquée est enregistrée. Accès en lecture.

Exemple

Dans cet exemple, le chemin du répertoire de la vue active est sorti :

```
Sub ShowDocumentPath()
'VBA663
MsgBox ActiveDocument.Path
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété Pathname

Description

Renvoie le chemin d'accès interne dans la bibliothèque de module de l'objet indiqué de type "FolderItem". Accès en lecture.

Exemple

Dans cet exemple, le chemin d'accès interne de l'objet "PC" se trouvant dans la bibliothèque de module globale est sorti :

```
Sub ShowInternalNameOfFolderItem()
'VBA664
Dim objGlobalLib As HMISymbolLibrary
Set objGlobalLib = Application.SymbolLibraries(1)
MsgBox objGlobalLib.FolderItems(2).Folder(2).Folder.Item(1).PathName
End Sub
```

Voir aussi

Objet FolderItem (Page 1927)

Accès à la bibliothèque des modules avec VBA (Page 1640)

Propriété PdlProtection

Description

Permet d'attribuer un mot de passe à la vue (vue de process ou type de bloc d'affichage) ou de supprimer le mot de passe. Accès en écriture.

Remarque

Signification de la protection par mot de passe

Notez que la propriété PdlProtection ne concerne que l'ouverture d'une vue.

Exemples

Dans l'exemple qui suit, on définit un mot de passe pour la vue active :

```
Sub ProtectPicture()
'VBAxxx
ActiveDocument.PdlProtection = "Test123"
End Sub
```

Dans l'exemple qui suit, on supprime le mot de passe de la vue active protégée par mot de passe :

```
Sub UnprotectPicture()
'VBAxxx
ActiveDocument.PdlProtection = ""
End Sub
```

Remarque

Accès en écriture uniquement.

Notez que, pour des raisons de sécurité, la lecture du mot de passe n'est pas autorisée.

Propriété PicDeactReferenced

Description

TRUE si la vue affectée pour l'état "Désactivé" doit être enregistrée dans l'objet RoundButton. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, la vue affectée à l'état "Désactivé" est référencée :

```
Sub RoundButtonConfiguration()
'VBA665
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicDeactReferenced = False
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Propriété PicDeactTransparent

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à l'état "Désactivé" doit être configurée sur "transparent", ou restitue la couleur. LONG Accès en écriture et en lecture.

La couleur n'est définie sur "Transparent" que si la propriété PicDeactUseTransColor a la valeur "True".

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, La couleur "Rouge" de l'objet Bitmap affecté à l'état "Désactivé" doit être définie valeur de "Transparent" :

```
Sub RoundButtonConfiguration()
'VBA666
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicDeactTransparent = RGB(255, 0, 0)
.PicDeactUseTransColor = True
End With
```

```
End Sub
```

Voir aussi

Propriété PicDeactUseTransColor (Page 2309)

Objet RoundButton (Page 2003)

Propriété PicDeactUseTransColor

Description

TRUE, lorsque la couleur transparente déterminée avec la propriété "PicDeactTransparent" doit être utilisée pour l'état "Désactivé". BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, La couleur "Rouge" de l'objet Bitmap affecté à l'état "Désactivé" doit être définie valeur de "Transparent" :

```
Sub RoundButtonConfiguration()
'VBA667
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicDeactTransparent = RGB(255, 0, 0)
.PicDeactUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété PicDeactTransparent (Page 2308)

Objet RoundButton (Page 2003)

Propriété PicDownReferenced

Description

TRUE si la vue affectée pour l'état "Activé" doit être enregistrée dans l'objet RoundButton. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, la vue affectée à l'état "Activé" est référencée :

```
Sub RoundButtonConfiguration()
'VBA668
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicDownReferenced = False
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Propriété PicDownTransparent

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à l'état "Marche" doit être configurée sur "transparent", ou restitue la couleur. LONG Accès en écriture et en lecture.

La couleur n'est définie sur "Transparent" que si la propriété PicDownUseTransColor a la valeur "True".

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, la couleur "Jaune" de l'objet Bitmap affecté à l'état "Activé" doit être définie valeur de "Transparent" :

```
Sub RoundButtonConfiguration()
'VBA669
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicDownTransparent = RGB(255, 255, 0)
.PicDownUseTransColor = True
End With
```

```
End Sub
```

Voir aussi

Propriété PicDownUseTransColor (Page 2311)

Objet RoundButton (Page 2003)

Propriété PicDownUseTransColor

Description

TRUE, lorsque la couleur transparente déterminée avec la propriété "PicDownTransparent" doit être utilisée pour l'état "Marche". BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, la couleur "Jaune" de l'objet Bitmap affecté à l'état "Activé" doit être définie valeur de "Transparent" :

```
Sub RoundButtonConfiguration()
'VBA670
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicDownTransparent = RGB(255, 255, 0)
.PicDownUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété PicDownTransparent (Page 2310)

Objet RoundButton (Page 2003)

Propriété PicReferenced

Description

TRUE si la vue affectée à l'objet GraphicObject est référencée et non pas enregistrée dans l'objet. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GraphicObjectConfiguration()" permet d'accéder aux propriétés de l'objet graphique. Dans cet exemple, la vue affectée doit être référencée :

```
Sub GraphicObjectConfiguration()
'VBA671
Dim objGraphicObject As HMIGraphicObject
Set objGraphicObject = ActiveDocument.HMIObjects.AddHMIObject("GraphicObject1",
"HMIGraphicObject")
With objGraphicObject
.PicReferenced = True
End With
End Sub
```

Voir aussi

Objet GraphicObject (Page 1931)

Propriété PictAlignment

Description

Définit comme attribut "Position de la vue" la position et la mise à l'échelle de la vue placée sur le bouton ou le bouton rond.

centré	La vue est placée au centre dans les proportions d'origine.
justifiée à gauche	La vue est placée alignée à gauche dans les proportions d'origine sur la partie gauche du bouton.
alignée à droite	La vue est placée alignée à droite dans les proportions d'origine sur la partie droite du bouton.
étirée	La vue est mise à l'échelle en carré et adaptée à la taille du bouton.

Propriété PicTransColor

Description

Détermine quelle couleur de l'objet Bitmap assigné (.bmp, .dib) doit être configurée sur "transparent" ou restitue la couleur. LONG Accès en écriture et en lecture.

La couleur n'est définie sur "Transparent" que si la propriété "PicUseTransColor" a la valeur "True".

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "GraphicObjectConfiguration()" permet d'accéder aux propriétés de l'objet graphique. Dans cet exemple, la couleur "Bleu" de l'objet Bitmap affecté doit être définie valeur de "Transparent" :

```
Sub GraphicObjectConfiguration()
'VBA672
Dim objGraphicObject As HMIGraphicObject
Set objGraphicObject = ActiveDocument.HMIObjects.AddHMIObject("GraphicObject1",
"HMIGraphicObject")
With objGraphicObject
.PicTransColor = 16711680
.PicUseTransColor = True
End With
End Sub
```

Voir aussi

Objet GraphicObject (Page 1931)

Propriété PictureDeactivated

Description

Détermine la vue qui s'affiche à l'état "Désactivé" ou restitue le nom de la vue.

La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, des vues sont définies pour les états "Activé" et "Désactivé" :

```
Sub ButtonConfiguration()
'VBA673
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
'
'Toi use this example copy a Bitmap-Graphic
'to the "GraCS"-Folder of the actual project.
'Replace the picturename "TestPicture1.BMP" with the name of
'the picture you copied
```

```
.PictureDeactivated = "TestPicture1.BMP"
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Propriété PicReferenced (Page 2311)

Propriété PictureDown

Description

Détermine la vue qui s'affiche à l'état "Activé" ou restitue le nom de la vue.

La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, des vues sont définies pour les états "Activé" et "Désactivé" :

```
Sub ButtonConfiguration()
'VBA674
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
'
'To use this example copy two Bitmap-Graphics
'to the "GraCS"-Folder of the actual project.
'Replace the picturenames "TestPicture1.BMP" and "TestPicture2.BMP"
'with the names of the pictures you copied
.PictureDown = "TestPicture1.BMP"
.PictureUp = "TestPicture2.BMP"
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Propriété PictureName

Description

Détermine la vue qui s'affiche dans la fenêtre de vue en Runtime ou restitue le nom de la vue.

La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA675
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété PictureUp

Description

Détermine la vue qui s'affiche à l'état "Arrêt" ou restitue le nom de la vue.

La vue (*.BMP ou *.DIB) doit se trouver dans le répertoire "GraCS" du projet actuel afin qu'elle puisse être incorporée.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, des vues sont définies pour les états "Activé" et "Désactivé" :

```
Sub ButtonConfiguration()
'VBA676
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
'
'To use this example copy two Bitmap-Graphics
'to the "GraCS"-Folder of the actual project.
'Replace the picturenames "TestPicture1.BMP" and "TestPicture2.BMP"
'with the names of the pictures you copied
.PictureDown = "TestPicture1.BMP"
.PictureUp = "TestPicture2.BMP"
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Objet Button (Page 1886)

Propriété PicUpReferenced

Description

TRUE si la vue affectée pour l'état "Désactivé" doit être enregistrée dans l'objet RoundButton. Faute de quoi, seul le renvoi à l'objet respectif est enregistré. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, la vue affectée à l'état "Désactivé" est référencée :

```
Sub RoundButtonConfiguration()
'VBA677
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicUpReferenced = False
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Propriété PicUpTransparent

Description

Détermine quelle couleur de l'objet Bitmap (.bmp, .dib) assigné à l'état "Arrêt" doit être configurée sur "transparent", ou restitue la couleur. LONG Accès en écriture et en lecture.

La couleur n'est définie sur "Transparent" que si la propriété "PicUpUseTransColor" a la valeur "True".

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, la couleur "Bleu" de l'objet Bitmap affecté à l'état "Désactivé" doit être définie valeur de "Transparent" :

```
Sub RoundButtonConfiguration()
'VBA678
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicUpTransparent = RGB(0, 0, 255)
.PicUpUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété PicUpUseTransColor (Page 2317)

Objet RoundButton (Page 2003)

Propriété PicUpUseTransColor

Description

TRUE, lorsque la couleur transparente déterminée avec la propriété "PicUpTransparent" doit être utilisée pour l'état "Arrêt". BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, la couleur "Bleu" de l'objet Bitmap affecté à l'état "Désactivé" doit être définie valeur de "Transparent" :

```
Sub RoundButtonConfiguration()
'VBA679
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.PicUpTransparent = RGB(0, 0, 255)
.PicUpUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété PicUpTransparent (Page 2317)

Objet RoundButton (Page 2003)

Propriété PicUseTransColor

Description

TRUE si la couleur transparente définie avec la propriété "PicTransColor" doit être utilisée pour l'état "Désactivé". BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GraphicObjectConfiguration()" permet d'accéder aux propriétés de l'objet graphique. Dans cet exemple, la couleur "Bleu" de l'objet Bitmap affecté doit être définie valeur de "Transparent" :

```
Sub GraphicObjectConfiguration()
'VBA680
Dim objGraphicObject As HMIGraphicObject
Set objGraphicObject = ActiveDocument.HMIObjects.AddHMIObject("GraphicObject1",
"HMIGraphicObject")
With objGraphicObject
.PicTransColor = RGB(0, 0, 255)
.PicUseTransColor = True
End With
End Sub
```

Voir aussi

Propriété PicTransColor (Page 2312)

Objet GraphicObject (Page 1931)

Propriété PointCount

Description

Définit ou renvoie pour les objets Polygon et PolyLine le nombre d'angles. Chaque point d'angle a des coordonnées de position et est identifié par un indice.

Exemple

Pour que cet exemple fonctionne, insérez un trait polygonal "Trait polygonal1" dans la vue active. La procédure "PolyLineCoordsOutput" permet de sortir les coordonnées de tous les angles du trait polygonal :

```
Sub PolyLineCoordsOutput()
'VBA681
Dim iPcIndex As Integer
Dim iPosX As Integer
Dim iPosY As Integer
Dim iIndex As Integer
Dim objPolyLine As HMIPolyLine
Set objPolyLine = Application.ActiveDocument.HMIObjects.AddHMIObject("PolyLine1",
"HMIPolyLine")

'
'Determine number of corners from "PolyLine1":
iPcIndex = objPolyLine.PointCount
'
'Output of x/y-coordinates from every corner:
For iIndex = 1 To iPcIndex
With objPolyLine
.index = iIndex
iPosX = .ActualPointLeft
iPosY = .ActualPointTop
MsgBox iIndex & ". corner:" & vbCrLf & "x-coordinate: " & iPosX & vbCrLf & "y-coordinate:
" & iPosY
End With
Next iIndex
End Sub
```

Listage des liaisons

Voir aussi

Propriété Index (Page 2207)

Propriété ActualPointTop (Page 2057)

Propriété ActualPointLeft (Page 2056)

Objet PolyLine (Page 1989)

Objet Polygon (Page 1986)

Propriété Position

Description

La valeur de la position détermine l'ordre dans les options de menu et des icônes dans les menus définis par l'utilisateur et les barres d'outils définies par l'utilisateur ou les menus définis par l'utilisateur dans la barre de menu. Accès en écriture et en lecture.

Une valeur de "1" correspond à la position 1 (début).

Exemple

Dans l'exemple qui suit, la position de toutes les options du premier menu défini par l'utilisateur de la vue active est sortie : Pour que cet exemple fonctionne, reportez-vous d'abord à l'exemple de "InsertSubMenu".

```
Sub ShowPositionOfCustomMenuItems()
'VBA683
Dim objMenu As HMIMenu
Dim iMaxMenuItems As Integer
Dim iPosition As Integer
Dim iIndex As Integer
Set objMenu = ActiveDocument.CustomMenus(1)
iMaxMenuItems = objMenu.MenuItems.Count
For iIndex = 1 To iMaxMenuItems
iPosition = objMenu.MenuItems(iIndex).Position
MsgBox "Position of the " & iIndex & ". menuitem: " & iPosition
Next iIndex
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Objet Menu (Page 1964)

Méthode InsertSubmenu (Page 1829)

Propriété PositiveValue

Description

Définit ou renvoie la valeur pour la propriété dynamisée si la variable configurée renvoie une valeur qui n'est pas égale à Zéro.

Exemple

Utilisez la propriété BinaryResultInfo pour restaurer un objet BinaryResultInfo. Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on affecte une valeur de propriété à chaque plage de valeurs binaire :

```
Sub AddDynamicDialogToCircleRadiusTypeBool()
'VBA684
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_C", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBool
.BinaryResultInfo.NegativeValue = 20
.BinaryResultInfo.PositiveValue = 40
End With
End Sub
```

Voir aussi

Propriété NegativeValue (Page 2291)

Objet BinaryResultInfo (Page 1884)

Référence VBA (Page 1724)

Propriété PredefinedAngels

Description

Détermine la représentation de la profondeur pour l'objet tridimensionnel ou la restitue. Plage des valeurs de 0 à 3.

Effets	Valeur affectée
Cavalière	0
Isométrique	1
Axonométrique	2
Librement défini	3

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la représentation en profondeur est définie valeur de "Isométrique" :

```
Sub HMI3DBarGraphConfiguration()
'VBA685
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
'Depth-angle a = 15 degrees
.AngleAlpha = 15
.PredefinedAngles = 1
'Depth-angle b = 45 degrees
.AngleBeta = 45
End With
End Sub
```

Voir aussi

Objet 3DBarGraph (Page 1867)

Propriété Pressed

Description

TRUE, lorsque l'objet Button ou RoundButton est enfoncé. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, le bouton rond est défini sur l'état "enfoncé" :

```
Sub RoundButtonConfiguration()
'VBA686
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.Pressed = True
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Propriété Process

Description

Définit le préréglage pour la valeur à afficher ou le restitue .

En Runtime, cette valeur est utilisée lorsqu'au moment de l'activation de la vue la variable correspondante n'est ni raccordée, ni mise à jour.

Exemple

La procédure "HMI3DBarGraphConfiguration()" permet d'accéder aux propriétés de l'objet 3DBarGraph. Dans cet exemple, la valeur est prédéfinie sur "100" :

```
Sub HMI3DBarGraphConfiguration()
'VBA687
Dim obj3DBar As HMI3DBarGraph
Set obj3DBar = ActiveDocument.HMIObjects.AddHMIObject("3DBar1", "HMI3DBarGraph")
With obj3DBar
'Depth-angle a = 15 degrees
.AngleAlpha = 15
'Depth-angle b = 45 degrees
.AngleBeta = 45
.Process = 100
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Objet OptionGroup (Page 1977)

Objet CheckBox (Page 1889)

Objet BarGraph (Page 1881)

Objet 3DBarGraph (Page 1867)

Propriété ProfileName

Description

Renvoie le nom de l'application indiquée. Accès en lecture.

Exemple

Dans cet exemple, le nom de l'application "Graphics Designer" est sorti :

```
Sub ShowProfileName()
'VBA688
MsgBox Application.ProfileName
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété ProgID

Description

Renvoie la ProgID d'un ActiveX Control. Accès en lecture STRING.

Exemple

Dans l'exemple qui suit, le contrôle ActiveX "WinCC Gauge Control" est inséré dans la vue active : La ProgID est sortie :

```
Sub AddActiveXControl()
'VBA689
Dim objActiveXControl As HMIActiveXControl
Set objActiveXControl = ActiveDocument.HMIObjects.AddActiveXControl("WinCC_Gauge",
"XGAUGE.XGaugeCtrl.1")
With ActiveDocument
.HMIObjects("WinCC_Gauge").Top = 40
.HMIObjects("WinCC_Gauge").Left = 40
MsgBox "ProgID of ActiveX-control: " & .HMIObjects("WinCC_Gauge").ProgID
End With
End Sub
```

Voir aussi

Objet ActiveXControl (Page 1873)

Méthode AddActiveXControl (Page 1774)

Propriété ProjectName

Description

Renvoie le nom du projet. Accès en lecture.

Exemple

Dans cet exemple, le nom et le type du projet chargé sont sortis :

```
Sub ShowProjectInfo()
'VBA690
Dim iProjectType As Integer
Dim strPropertyName As String
Dim strProjectType As String
iProjectType = Application.ProjectType
strProjectName = Application.ProjectName
Select Case iProjectType
Case 0
strProjectType = "Single-User System"
Case 1
strProjectType = "Multi-User System"
Case 2
strProjectType = "Client System"
End Select
MsgBox "Projecttype: " & strProjectType & vbCrLf & "Projectname: " & strProjectName
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété ProjectType

Description

Renvoie le type de projet. Plage de valeurs de 0 à 2. Accès en lecture.

Type de projet	Valeur affectée
Projet monoposte	0
Projet multiposte	1
Projet client	2

Exemple

Dans cet exemple, le nom et le type du projet chargé sont sortis :

```
Sub ShowProjectInfo()
'VBA691
Dim iProjectType As Integer
Dim strPropertyName As String
Dim strProjectType As String
iProjectType = Application.ProjectType
strProjectName = Application.ProjectName
Select Case iProjectType
Case 0
strProjectType = "Single-User System"
Case 1
strProjectType = "Multi-User System"
Case 2
strProjectType = "Client System"
End Select
MsgBox "Projecttype: " & strProjectType & vbCrLf & "Projectname: " & strProjectName
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété Properties

Description

Renvoie un listage Properties contenant toutes les propriétés de l'objet indiqué. Accès en lecture.

Pour renvoyer un élément du listage Properties, vous pouvez utiliser le numéro d'index ou le nom de la propriété VBA.

Vous devez utiliser la propriété Properties par ex. si vous souhaitez accéder aux propriétés d'objets se trouvant dans un objet groupe.

Exemple

Pour des exemples d'application de la propriété Properties, reportez-vous aux rubriques suivantes dans cette documentation :

- "Editer les objets avec VBA"
- "Objets groupes"
- "Objets utilisateur"

Voir aussi

Objet HMIObject (Page 1943)
Objets utilisateur (Page 1676)
Objets Groupe (Page 1668)
Edition d'objets avec VBA (Page 1653)

Propriété Prototype

Description

Renvoie l'en-tête d'une fonction d'un script. L'en-tête de fonction est attribué par défaut si aucun code source n'est configuré.

Exemple

Dans cet exemple, on insère un bouton et un cercle dans la vue active. Dans Runtime, le rayon du cercle doit être agrandi à chaque clic sur le bouton. Dans ce cas, seul le prototype du script VB est sorti :

```
Sub ExampleForPrototype()
'VBA692
Dim objButton As HMIButton
Dim objCircleA As HMICircle
Dim objEvent As HMIEvent
Dim objVBScript As HMIScriptInfo
Set objCircleA = ActiveDocument.HMIObjects.AddHMIObject("CircleA", "HMICircle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircleA
.Top = 100
.Left = 100
End With
With objButton
.Top = 10
.Left = 10
.Width = 200
.Text = "Increase Radius"
End With
'On every mouseclick the radius have to increase:
Set objEvent = objButton.Events(1)
Set objVBScript = objButton.Events(1).Actions.AddAction(hmiActionCreationTypeVBScript)
MsgBox objVBScript.Prototype
End Sub
```

Voir aussi

Objet ScriptInfo (Page 2009)

Propriété QualityCodeStateChecked

Description

VRAI si le code qualité de la variable indiquée dans le dialogue dynamique est utilisé pour la dynamisation. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA816
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
```

```
End Sub
```

Voir aussi

Objet DynamicDialog (Page 1912)

Propriété QualityCodeStateValues

Description

Renvoie la liste des QualityCodeStateValues. Utilisez la propriété QualityCodeStateValues avec la propriété Item pour affecter une valeur à un état de code qualité qui est utilisé pour la dynamisation.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA817
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
```

```
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Objet DynamicDialog (Page 1912)

Objet QualityCodeStateValues (listage) (Page 1997)

R

Propriété Radius

Description

Définit ou renvoie le rayon pour les objets suivants :

- Circle: rayon en pixels (0 à 10000)
- CircularArc: rayon en pixels (0 à 10000)
- PieSegment: rayon en pixels (0 à 10000)
- RoundButton: rayon en pixels (0 à 10000)

Exemple

La procédure "PieSegmentConfiguration()" permet d'accéder aux propriétés du segment de cercle. Dans cet exemple, le rayon est défini sur "80" :

```
Sub PieSegmentConfiguration()
'VBA693
Dim objPieSegment As HMIPieSegment
Set objPieSegment = ActiveDocument.HMIObjects.AddHMIObject("PieSegment1", "HMIPieSegment")
With objPieSegment
.StartAngle = 40
.EndAngle = 180
.Rayon = 80
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Objet PieSegment (Page 1983)

Objet CircularArc (Page 1893)

Objet Circle (Page 1890)

Propriété RadiusHeight

Description

Définit ou renvoie pour les objets elliptiques (Ellipse, EllipseArc, EllipseSegment) le rayon vertical en pixels (0 à 5000).

Exemple

La procédure "EllipseConfiguration()" permet d'accéder aux propriétés de l'objet Ellipse. Dans cet exemple, le rayon vertical est défini sur "60" :

```
Sub EllipseConfiguration()
'VBA694
Dim objEllipse As HMIEllipse
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("Ellipse1", "HMIEllipse")
With objEllipse
.RadiusHeight = 60
.RadiusWidth = 40
End With
End Sub
```

Voir aussi

Propriété RadiusWidth (Page 2331)

Objet EllipseSegment (Page 1920)

Objet EllipseArc (Page 1917)

Objet Ellipse (Page 1914)

Propriété RadiusWidth

Description

Définit ou renvoie pour les objets elliptiques (Ellipse, EllipseArc, EllipseSegment) le rayon horizontal en pixels (0 à 5000).

Exemple

La procédure "EllipseConfiguration()" permet d'accéder aux propriétés de l'objet Ellipse. Dans cet exemple, le rayon vertical est défini sur "40" :

```
Sub EllipseConfiguration()
'VBA695
Dim objEllipse As HMIEllipse
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("Ellipse1", "HMIEllipse")
With objEllipse
.RadiusHeight = 60
.RadiusWidth = 40
End With
End Sub
```

Voir aussi

Propriété RadiusHeight (Page 2331)

Objet EllipseSegment (Page 1920)

Objet EllipseArc (Page 1917)

Objet Ellipse (Page 1914)

Propriété RangeTo

Description

Définit ou renvoie la plage de valeurs analogique.

Exemple

Un exemple d'application de la propriété RangeTo se trouve dans cette documentation, à la rubrique Objet AnalogResultInfos (listage).

Voir aussi

Propriété Value (Page 2395)

Objet AnalogResultInfos (liste) (Page 1875)

Objet AnalogResultInfo (Page 1874)

Propriété ReferenceRotationLeft

Description

Détermine la coordonnée X du point de référence autour duquel l'objet doit pivoter en Runtime ou la restitue.

La valeur de la coordonnée X est relative à la largeur de l'objet. Spécifiez en pour cent la valeur en partant du bord gauche du rectangle entourant l'objet.

Exemple

La procédure "PolyLineConfiguration()" permet d'accéder aux propriétés de l'objet PolyLine. Dans cet exemple, les coordonnées du point de référence sont définies à 50 % de la largeur de l'objet et à 50 % de la hauteur de l'objet.

```
Sub PolyLineConfiguration()
'VBA696
Dim objPolyLine As HMIPolyLine
Set objPolyLine = ActiveDocument.HMIObjects.AddHMIObject("PolyLine1", "HMIPolyLine")
With objPolyLine
.ReferenceRotationLeft = 50
.ReferenceRotationTop = 50
End With
End Sub
```

Voir aussi

Propriété RotationAngle (Page 2336)

Propriété ReferenceRotationTop (Page 2333)

Objet PolyLine (Page 1989)

Objet Polygon (Page 1986)

Objet Line (Page 1958)

Propriété ReferenceRotationTop

Description

Détermine la coordonnée Y du point de référence autour duquel l'objet doit pivoter en Runtime ou la restitue.

La valeur de la coordonnée Y est relative à la largeur de l'objet. Spécifiez en pour cent la valeur en partant du bord supérieur du rectangle entourant l'objet.

Exemple

La procédure "PolyLineConfiguration()" permet d'accéder aux propriétés de l'objet PolyLine. Dans cet exemple, les coordonnées du point de référence sont définies à 50 % de la largeur de l'objet et à 50 % de la hauteur de l'objet.

```
Sub PolyLineConfiguration()
'VBA697
```

```
Dim objPolyLine As HMIPolyLine
Set objPolyLine = ActiveDocument.HMIObjects.AddHMIObject("PolyLine1", "HMIPolyLine")
With objPolyLine
.ReferenceRotationLeft = 50
.ReferenceRotationTop = 50
End With
End Sub
```

Voir aussi

Propriété RotationAngle (Page 2336)

Propriété ReferenceRotationLeft (Page 2332)

Objet PolyLine (Page 1989)

Objet Polygon (Page 1986)

Objet Line (Page 1958)

Propriété Relevant

Description

TRUE, lorsque l'objet est pris en compte pour la constitution de la signalisation groupée. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, l'objet est pris en compte pour la formation de la signalisation groupée :

```
Sub GroupDisplayConfiguration()
'VBA698
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.Relevant = True
End With
End Sub
```

Voir aussi

Objet Group (Page 1934)

Propriété ResultType

Description

Définit ou renvoie le type d'analyse des plages de valeurs dans le Dialogue Dynamique.

Exemple

Dans l'exemple qui suit, on dynamise le rayon d'un cercle par le dialogue Dynamique, on attribue un nom de variable et on affecte une valeur de propriété à chaque plage de valeurs binaire :

```
Sub AddDynamicDialogToCircleRadiusTypeBinary()
'VBA699
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_C", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeBool
.BinaryResultInfo.NegativeValue = 20
.BinaryResultInfo.PositiveValue = 40
End With
End Sub
```

Voir aussi

Objet DynamicDialog (Page 1912)

Propriété RightComma

Description

Définit ou renvoie le nombre de chiffres après la virgule (0 à 20) pour l'objet BarGraph.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, le nombre de chiffres après la virgule est limité à 4.

```
Sub BarGraphConfiguration()
'VBA700
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.RightComma = 4
```

```
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété RotationAngle

Description

Line, Polygon et PolyLine

Définit l'angle de rotation des objets suivants en degrés, ou le retourne Line, Polygon, PolyLine.

L'objet est représenté en runtime seulement (à partir de la position initiale configurée) pivoté autour du point de référence dans le sens des aiguilles d'une montre à hauteur de la valeur entrée.

Raccord en T

Définit l'orientation d'un raccord en T en degrés ou la retourne. L'attribut ne peut accepter que quatre valeurs :

0	La position par défaut du raccord en T représentée par la lettre "T"
90	La "jambe" du "T" est dirigée vers la gauche
180	La "jambe" du "T" est dirigée vers le haut
270	La "jambe" du "T" est dirigée vers la droite

D'autres valeurs sont automatiquement converties en module 360 et arrondies à la valeur supérieure ou inférieure admissible la plus proche.

Le raccord en T est représenté dans le projet et en runtime pivotant autour de son centre.

Exemple

La procédure "PolyLineConfiguration()" permet d'accéder aux propriétés de l'objet PolyLine. Dans cet exemple, l'objet pivote de 45° dans Runtime :

```
Sub PolyLineConfiguration()
'VBA701
Dim objPolyLine As HMIPolyLine
Set objPolyLine = ActiveDocument.HMIObjects.AddHMIObject("PolyLine1", "HMIPolyLine")
With objPolyLine
.ReferenceRotationLeft = 50
.ReferenceRotationTop = 50
.RotationAngle = 45
End With
```

```
End Sub
```

Voir aussi

Propriété ReferenceRotationTop (Page 2333)

Propriété ReferenceRotationLeft (Page 2332)

Objet PolyLine (Page 1989)

Objet Polygon (Page 1986)

Objet Line (Page 1958)

Propriété RoundCornerHeight

Description

Définit ou renvoie le rayon de l'angle de l'objet RoundRectangle.

Tapez la valeur en pourcentage de la demi-hauteur de l'objet.

Exemple

La procédure "RoundRectangleConfiguration()" permet d'accéder aux propriétés de l'objet RoundRectangle. Dans cet exemple, le rayon de l'angle est défini sur 25 % (hauteur) et 50 % (largeur) :

```
Sub RoundRectangleConfiguration()
'VBA702
Dim objRoundRectangle As HMIRoundRectangle
Set objRoundRectangle = ActiveDocument.HMIObjects.AddHMIObject("RoundRectangle1",
"HMIRoundRectangle")
With objRoundRectangle
.RoundCornerHeight = 25
.RoundCornerWidth = 50
End With
End Sub
```

Voir aussi

Propriété RoundCornerWidth (Page 2338)

Objet RoundRectangle (Page 2006)

Propriété RoundCornerWidth

Description

Définit ou renvoie le rayon de l'angle de l'objet RoundRectangle.

Tapez la valeur en pourcentage de la demi-largeur de l'objet.

Exemple

La procédure "RoundRectangleConfiguration()" permet d'accéder aux propriétés de l'objet RoundRectangle. Dans cet exemple, le rayon de l'angle est défini sur 25 % (hauteur) et 50 % (largeur) :

```
Sub RoundRectangleConfiguration()
'VBA703
Dim objRoundRectangle As HMIRoundRectangle
Set objRoundRectangle = ActiveDocument.HMIObjects.AddHMIObject("RoundRectangle1",
"HMIRoundRectangle")
With objRoundRectangle
.RoundCornerHeight = 25
.RoundCornerWidth = 50
End With
End Sub
```

Voir aussi

Propriété RoundCornerHeight (Page 2337)

Objet RoundRectangle (Page 2006)

S

Propriété SameSize

Description

TRUE, lorsque les quatre boutons de l'objet GroupDisplay sont de même taille. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, les quatre boutons ont la même taille :

```
Sub GroupDisplayConfiguration()
```

```
'VBA704
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.SameSize = True
End With
End Sub
```

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété ScaleColor

Description

Détermine la couleur de l'échelle ou la restitue. LONG Accès en écriture et en lecture.

Il faut que la propriété "Scaling" soit positionnée sur TRUE pour que la couleur s'affiche.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, l'échelle est en plus affichée et la couleur d'échelle est fixée sur "Rouge" :

```
Sub BarGraphConfiguration()
'VBA705
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Scaling = True
.ScaleColor = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Propriété Scaling (Page 2340)

Objet BarGraph (Page 1881)

Propriété ScaleTicks

Description

Définit ou renvoie pour l'objet BarGraph le nombre de sections d'échelle.

La section d'échelle est la zone délimitée par deux grandes divisions d'échelle. Si vous affectez la valeur "0" à la propriété, la division d'échelle appropriée est automatiquement calculée.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, le nombre de sections d'échelle est défini sur "10" :

```
Sub BarGraphConfiguration()
'VBA706
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Scaling = True
.ScaleTicks = 10
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété Scaling

Description

TRUE si pour l'objet BarGraph une échelle est utilisée en plus pour la représentation des valeurs. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, l'échelle est en plus affichée et la couleur d'échelle est fixée sur "Rouge" :

```
Sub BarGraphConfiguration()
'VBA707
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Scaling = True
```

```
.ScaleColor = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété ScalingMode

Description

Définit la taille avec laquelle les objets de l'instance de bloc d'affichage sont représentés.

Par défaut	Comme le mode d'échelle "proportionnel"
1 : 1	Le type de bloc d'affichage est représenté dans sa taille d'origine dans l'instance de bloc d'affichage. Si l'instance de bloc d'affichage est trop petite, la taille de l'instance de bloc d'affichage s'adapte à la taille du type de bloc d'affichage.
Proportionnel	Le type de bloc d'affichage est mis à l'échelle proportionnellement à la taille de l'instance de bloc d'affichage.

Exemple

Propriété ScalingType

Description

Détermine le type d'échelle du bargraphe ou le restitue. Plage des valeurs de 0 à 2.

Il faut que la propriété "Scaling" soit positionnée sur TRUE pour que la couleur s'affiche.

Mise à l'échelle du bargraphe	Valeur affectée
Linéaire	0
Logarithmique	1
Automatique	2

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la mise à l'échelle du bargraphe est définie sur "Linéaire"

```
Sub BarGraphConfiguration()
'VBA708
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.ScalingType = 0
.Scaling = True
End With
End Sub
```

Voir aussi

Propriété Scaling (Page 2340)

Objet BarGraph (Page 1881)

Propriété ScriptType

Description

Renvoie le type de script (C ou VBS) avec lequel une propriété ou un événement a été dynamisé. Accès en lecture.

Exemple

Dans cet exemple, on insère un bouton et un cercle dans la vue active. Dans Runtime, le rayon du cercle doit être agrandi à chaque clic sur le bouton. Dans ce cas, le type de script est sorti :

```
Sub ExampleForPrototype()
'VBA709
Dim objButton As HMIButton
Dim objCircleA As HMICircle
Dim objEvent As HMIEvent
Dim objVBScript As HMIScriptInfo
Dim strScriptType As String
Set objCircleA = ActiveDocument.HMIObjects.AddHMIObject("CircleA", "HMICircle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircleA
.Top = 100
.Left = 100
End With
With objButton
.Top = 10
.Left = 10
```

```
.Width = 200
.Text = "Increase Radius"
End With
'On every mouseclick the radius have to increase:
Set objEvent = objButton.Events(1)
Set objVBScript = objButton.Events(1).Actions.AddAction(hmiActionCreationTypeVBScript)
Select Case objVBScript.ScriptType
Case 0
strScriptType = "VB-Script is used"
Case 1
strScriptType = "C-Script is used"
End Select
MsgBox strScriptType
End Sub
```

Voir aussi

Objet ScriptInfo (Page 2009)

Propriété ScrollBars

Description

TRUE si la fenêtre de vue possède dans Runtime des barres de défilement. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA710
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
```

```
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété ScrollPositionX

Description

Définit le déplacement horizontal de la barre de défilement dans une fenêtre de vue ou restitue la valeur.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA808
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ScrollPositionX = 50
.ScrollPositionY = 50
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété ScrollPositionY

Description

Définit le déplacement vertical de la barre de défilement dans une fenêtre de vue ou restitue la valeur.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA809
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ScrollPositionX = 50
.ScrollPositionY = 50
.ServerPrefix = ""
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété ScrollPosX

Description

Définit ou renvoie pour l'objet View la position X des barres de défilement.

Exemple

Dans l'exemple qui suit, la vue active est copiée et la copie est ensuite activée. La position des barres de défilement est définie sur 40 (x) et 10 (y) :

```
Sub CreateViewAndActivateView()
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
objView.ScrollPosX = 40
objView.ScrollPosY = 10
End Sub
```

Voir aussi

Propriété ScrollPosY (Page 2346)

Objet View (Page 2050)

Propriété ScrollPosY

Description

Définit ou renvoie pour l'objet View la position Y des barres de défilement.

Exemple

Dans l'exemple qui suit, la vue active est copiée et la copie est ensuite activée. La position des barres de défilement est définie sur 40 (x) et 10 (y) :

```
Sub CreateViewAndActivateView()
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
objView.ScrollPosX = 40
objView.ScrollPosY = 10
End Sub
```

Voir aussi

Propriété ScrollPosX (Page 2345)

Objet View (Page 2050)

Propriété SelBGColor

Description

Définit ou renvoie la couleur d'arrière-plan pour l'entrée sélectionnée pour l'objet TextList. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, la couleur d'arrière-plan de l'entrée sélectionnée est définie sur "Rouge" :

```
Sub TextListConfiguration()
Dim objTextList As HMITextList
'
'Insérer une nouvelle Textliste dans la vue présente :
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.SelBGColor = RGB (255, 0, 0)
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété Selected

Description

TRUE si un objet est sélectionné dans la vue. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple qui suit, on insère deux nouveaux objets dans la vue active, puis on les sélectionne.

```
Sub SelectObjects()
'VBA714
```

```
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objGroup As HMIGroup
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
With objCircle
.Top = 40
.Left = 40
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 80
.Selected = True
End With
MsgBox "Objects selected!"
End Sub
```

Voir aussi

Objet HMIObject (Page 1943)

Propriété Selection

Description

Renvoie une liste qui contient tous les objets sélectionnés dans la vue indiquée.

Pour renvoyer un élément de la liste Selection, vous pouvez utiliser soit le numéro d'index, soit le nom d'objet.

Vous pouvez p. ex. utiliser la propriété Selection pour sélectionner tous les objets dans la vue.

Exemple

Dans l'exemple qui suit, on sélectionne tous les objets de la vue active.

```
Sub SelectAllObjectsInActiveDocument()
'VBA715
ActiveDocument.Selection.SelectAll
End Sub
```

Voir aussi

Objet Selection (liste) (Page 2010)

Objet Document (Page 1908)

Propriété SelIndex

Description

Définit ou retourne l'index dont le texte assigné est affiché en surmarquage dans le champ de combinaison ou le champ de liste.

Propriété SelText

Description

Montre le texte défini avec la propriété SelIndex, qui est représenté en surmarquage dans l'objet Champ de combinaison ou Champ de liste. Vous ne pouvez pas modifier directement l'attribut "Texte sélectionné". Vous modifiez l'attribut "Texte sélectionné" uniquement de façon indirecte en modifiant l'attribut "Champ sélectionné" ou en modifiant le texte même dans le groupe de propriétés "Police".

Propriété SelTextColor

Description

Définit ou renvoie la couleur de texte de l'entrée sélectionnée de l'objet TextList. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple, la couleur du texte de l'entrée sélectionnée est définie sur "Jaune" :

```
Sub TextListConfiguration()
'VBA716
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.SelTextColor = RGB(255, 255, 0)
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété ServerName

Description

Renvoie la désignation de l'ActiveX Control indiqué. Accès en lecture.

Exemple

Dans l'exemple suivant, le contrôle ActiveX "WinCC Gauge Control" est inséré dans la vue active et le nom de l'ActiveX Control est sorti:

```
Sub AddActiveXControl()
'VBA717
Dim objActiveXControl As HMIActiveXControl
Set objActiveXControl = ActiveDocument.HMIObjects.AddActiveXControl("WinCC_Gauge",
"XGAUGE.XGaugeCtrl.1")
With objActiveXControl
.Top = 40
.Left = 60
MsgBox .Properties("ServerName").value
End With
End Sub
```

Voir aussi

Objet ActiveXControl (Page 1873)

Propriété ServerPrefix

Description

Définit sur quel serveur se trouve la vue qui est affichée dans la fenêtre de vue dans Runtime ou renvoie le nom du serveur.

Spécifiez le nom de serveur suivi de deux deux-points: "<Nom de serveur>::". Le système ne vérifie pas si le serveur existe effectivement.

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA718
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = "my_Server::"
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété ShortCut

Description

Définit ou renvoie un raccourci clavier pour une option de menu personnalisée ou une icône personnalisée.

Les touches suivantes sont autorisées en combinaison avec <Ctrl>, <Alt> et <Maj> :

- Touches de fonction <F1> à <F12>
- Touches alphabétiques et numériques de <A> à <Z> et <0> à <9>.

Ne sont pas prises en compte les touches du pavé alphanumérique, toutes les touches de curseur (p. ex. <Ouvrir vue>) ainsi que les autres touches de fonction comme <OK> et <Echapp>. Il n'est pas fait de distinction entre majuscules et minuscules. Les combinaisons avec deux lettres/chiffres ou plus, comme par ex. "CTRL+A+B", ne sont pas admises alors que celles avec deux touches complémentaires le sont, comme par ex. "CTRL+ALT+A".

Informations sur l'utilisation de la propriété ShortCut

Les raccourcis clavier utilisés doivent être sans équivoque au sein des menus et barres d'outils d'une vue personnalisés. Les raccourcis clavier que vous configurez avec VBA ont la priorité sur d'éventuels raccourcis clavier présents dans Graphics Designer. Au sein des menus et barres d'outils personnalisés, les raccourcis clavier spécifiques à la vue ont la priorité sur les raccourcis clavier spécifiques à l'application.

Remarque

Les raccourcis clavier ne sont exécutés que si l'option de menu ou l'icône est visible et active.

Exemple

Dans l'exemple qui suit, on crée dans la vue active un menu personnalisé à deux entrées et un sous-menu à deux entrées. Le sous-menu est visuellement caractérisé par une ligne de séparation. La première entrée de menu contient le raccourci clavier <Ctrl+Maj+M> pour l'appel :

```
Sub CreateDocumentMenus()
'VBA719
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objSubMenu As HMIMenuItem
'
'Add menu to menubar:
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
'
'Add menuitems to the new menu:
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "&My first Menuitem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "My second Menuitem")
'
'Add seperator to menu:
Set objMenuItem = objDocMenu.MenuItems.InsertSeparator(3, "dSeparator1_3")
'
'Add submenu to the menu:
Set objSubMenu = objDocMenu.MenuItems.InsertSubMenu(4, "dSubMenu1_4", "My first submenu")
'
'Add menuitems to the submenu:
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(5, "dmItem1_5", "My first submenuitem")
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(6, "dmItem1_6", "My second
submenuitem")
'
ActiveDocument.CustomMenus("DocMenu1").MenuItems(1).ShortCut = "STRG+SHIFT+M"
End Sub
```

Voir aussi

Configuration des menus et des barres d'outils (Page 1619)

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Propriété SignificantMask

Description

Est nécessaire dans Runtime pour représenter la classe d'alarmes active avec la plus haute priorité de l'objet GroupDisplay.

La valeur de la propriété SignificantMask représente une valeur de sortie interne au système qui ne requiert aucune configuration spécifique de la part de l'utilisateur. Pour la mettre à jour, cliquez dans Runtime sur l'objet.

Exemple

--

Voir aussi

Objet GroupDisplay (Page 1935)

Propriété Size

Description

Définit ou renvoie la taille en points d'une police en fonction de la langue.

Exemple

Dans l'exemple suivant, on configure les attributs de texte d'un bouton pour le français et pour l'anglais :

```
Sub ExampleForLanguageFonts()
'VBA721
Dim colLangFonts As HMILanguageFonts
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
objButton.Text = "DefText"
Set colLangFonts = objButton.LDFonts
'
'Set font-properties for french:
With colLangFonts.ItemByLCID(1036)
.Family = "Courier New"
```

```
.Bold = True
.Italic = False
.Underlined = True
.Size = 12
End With
'
'Set font-properties for english:
With colLangFonts.ItemByLCID(1033)
.Family = "Times New Roman"
.Bold = False
.Italic = True
.Underlined = False
.Size = 14
End With
End Sub
```

Voir aussi

Propriété Underlined (Page 2388)

Propriété Parent (Page 2303)

Propriété LanguageID (Page 2219)

Propriété Italic (Page 2212)

Propriété FontFamily (Page 2189)

Propriété Bold (Page 2094)

Propriété Application (Page 2067)

Objet LanguageFont (Page 1950)

Propriété Sizeable

Description

TRUE si dans Runtime la taille des objets ApplicationWindow et PictureWindow peut être modifiée. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ApplicationWindowConfig" permet d'accéder aux propriétés de la fenêtre d'application. Dans cet exemple, la taille de la fenêtre d'application peut être modifiée dans Runtime :

```
Sub ApplicationWindowConfig()
'VBA722
Dim objAppWindow As HMIApplicationWindow
Set objAppWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow1",
"HMIApplicationWindow")
```

```
With objAppWindow
.Sizeable = True
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Objet ApplicationWindow (Page 1879)

Propriété SmallChange

Description

Détermine de combien de pas le régulateur doit être déplacé avec un clic de souris, ou restitue cette valeur.

Exemple

La procédure "SliderConfiguration()" permet d'accéder aux propriétés du défileur. Dans cet exemple, le nombre de pas est défini sur "4" :

```
Sub SliderConfiguration()
'VBA723
Dim objSlider As HMISlider
Set objSlider = ActiveDocument.HMIObjects.AddHMIObject("SliderObject1", "HMISlider")
With objSlider
.SmallChange = 4
End With
End Sub
```

Voir aussi

Objet Slider (Page 2013)

Propriété SnapToGrid

Description

TRUE si des objets dans la vue sont alignés sur le quadrillage (invisible). BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple suivant, l'alignement des objets de la vue active dans le quadrillage est activé :

```
Sub ActivateSnapToGrid()
'VBA724
ActiveDocument.SnapToGrid = True
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété SourceLink

Description

Renvoie l'objet Source. Utilisez la propriété SourceLink pour configurer en cas de liaison directe l'objet source.

Exemple

Dans l'exemple qui suit, dans Runtime, la position X du "Rectangle _A" est copiée à la position Y du "Rectangle_B" par un clic de souris sur le bouton :

```
Sub DirectConnection()
'VBA725
Dim objButton As HMIButton
Dim objRectangleA As HMIRectangle
Dim objRectangleB As HMIRectangle
Dim objEvent As HMIEvent
Dim objDirConnection As HMIDirectConnection
Set objRectangleA = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_A", "HMIRectangle")
Set objRectangleB = ActiveDocument.HMIObjects.AddHMIObject("Rectangle_B", "HMIRectangle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objRectangleA
.Top = 100
.Left = 100
End With
With objRectangleB
.Top = 250
.Left = 400
.BackColor = RGB(255, 0, 0)
End With
With objButton
.Top = 10
.Left = 10
.Width = 100
.Text = "SetPosition"
```

```
End With
'
'Directconnection is initiated by mouseclick:
Set objDirConnection =
objButton.Events(1).Actions.AddAction(hmiActionCreationTypeDirectConnection)
With objDirConnection
'Sourceobject: Property "Top" of Rectangle_A
.SourceLink.Type = hmiSourceTypeProperty
.SourceLink.ObjectName = "Rectangle_A"
.SourceLink.AutomationName = "Top"
'
'Targetobject: Property "Left" of Rectangle_B
.DestinationLink.Type = hmiDestTypeProperty
.DestinationLink.ObjectName = "Rectangle_B"
.DestinationLink.AutomationName = "Left"
End With
End Sub
```

Voir aussi

Propriété Type (Page 2378)

Propriété ObjectName (Page 2293)

Propriété AutomationName (Page 2070)

Objet SourceLink (Page 2016)

Objet DirectConnection (Page 1906)

Propriété SourceCode

Description

Définit ou renvoie le code source d'un script en C ou VB.

Si vous affectez à une propriété SourceCode un script en C, vous devez uniquement saisir le code du programme qui se trouve entre les deux accolades ("{}").

Si vous affectez à une propriété SourceCode un script VB, vous devez uniquement saisir le code du programme qui se trouve entre les mots clés Sub et EndSub.

Remarque

Si vous utilisez dans le code programme des guillemets simples (') ou des guillemets doubles (") vous devez placer devant chaque guillemet un guillemet supplémentaire pour que le code du programme puisse être correctement interprété dans l'éditeur VBA.

La propriété Compiled renvoie un TRUE si le code source a été compilé avec succès.

Exemple

Dans cet exemple, on insère un bouton et un cercle dans la vue active. Dans Runtime, le rayon du cercle doit être agrandi à chaque clic sur le bouton. On utilise pour cela un script VB :

```
Sub IncreaseCircleRadiusWithVBScript()
'VBA726
Dim objButton As HMIButton
Dim objCircleA As HMICircle
Dim objEvent As HMIEvent
Dim objVBScript As HMIScriptInfo
Dim strCode As String
strCode = "Dim objCircle" & vbCrLf & "Set objCircle = "
strCode = strCode & "hmiRuntime.ActiveScreen.ScreenItems(""CircleVB"")"
strCode = strCode & vbCrLf & "objCircle.Radius = objCircle.Radius + 5"
Set objCircleA = ActiveDocument.HMIObjects.AddHMIObject("CircleVB", "HMICircle")
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
With objCircleA
.Top = 100
.Left = 100
End With
With objButton
.Top = 10
.Left = 10
.Width = 200
.Text = "Increase Radius"
End With
'
'On every mouseclick the radius have to increase:
Set objEvent = objButton.Events(1)
Set objVBScript = objButton.Events(1).Actions.AddAction(hmiActionCreationTypeVBScript)
objVBScript.SourceCode = strCode
Select Case objVBScript.Compiled
Case True
MsgBox "Compilation ok!"
Case False
MsgBox "Error on compilation!"
End Select
End Sub
```

Voir aussi

Propriété ActualPointTop (Page 2140)

Objet ScriptInfo (Page 2009)

Propriété StartAngle

Description

Définit ou renvoie pour les objets CircularArc, EllipseArc, EllipseSegment et PieSegment le début de l'objet. L'indication se fait dans le sens horaire, en degrés, en commençant à 12:00 h.

Exemple

La procédure "PieSegmentConfiguration()" permet d'accéder aux propriétés du segment de cercle. Dans cet exemple, le segment de cercle commence à 40° et finit à 180°:

```
Sub PieSegmentConfiguration()
'VBA727
Dim PieSegment As HMIPieSegment
Set PieSegment = ActiveDocument.HMIObjects.AddHMIObject("PieSegment1", "HMIPieSegment")
With PieSegment
.StartAngle = 40
.EndAngle = 180
End With
End Sub
```

Voir aussi

Propriété EndAngle (Page 2158)

Objet PieSegment (Page 1983)

Objet EllipseSegment (Page 1920)

Objet EllipseArc (Page 1917)

Objet CircularArc (Page 1893)

Propriété StatusText

Description

Définit ou renvoie le texte qui est affiché dans la barre d´état lorsque vous pointez avec la souris sur une option de menu personnalisée ou une icône personnalisée.

Exemple

Dans l'exemple qui suit, on crée dans la vue active un menu personnalisé à deux entrées et un sous-menu à deux entrées. Le sous-menu est visuellement caractérisé par une ligne de séparation. Une entrée sur la barre d'état est définie pour chaque option de menu :

```
Sub CreateDocumentMenus()
'VBA728
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objSubMenu As HMIMenuItem
'
'Add menu:
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
'
'Add menuitems to custom-menu:
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "My first menuitem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "My second menuitem")
'
'Add seperator to custom-menu:
Set objMenuItem = objDocMenu.MenuItems.InsertSeparator(3, "dSeparator1_3")
'
'Add submenu to custom-menu:
Set objSubMenu = objDocMenu.MenuItems.InsertSubMenu(4, "dSubMenu1_4", "My first submenu")
'
'Add menuitems to submenu:
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(5, "dmItem1_5", "My first submenuitem")
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(6, "dmItem1_6", "My second
submenuitem")
'
'Assign statustexts to every menuitem
With objDocMenu
.MenuItems(1).StatusText = "My first menuitem"
.MenuItems(2).StatusText = "My second menuitem"
.MenuItems(4).SubMenu.Item(1).StatusText = "My first submenuitem"
.MenuItems(4).SubMenu.Item(2).StatusText = "My second submenuitem"
End With
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Propriété SubMenu

Description

Renvoie une liste MenuItems lorsque l'objet indiqué est du type "SubMenu".

Utilisez la liste SubMenu quand vous voulez créer un sous-menu dans un menu personnalisé.

Exemple

Dans l'exemple qui suit, on crée dans la vue active un menu personnalisé à deux entrées et un sous-menu à deux entrées. Le sous-menu est visuellement caractérisé par une ligne de séparation :

```
Sub CreateDocumentMenus()
'VBA730
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objSubMenu As HMIMenuItem
'
'Add menu:
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
'
'Add menuitems to custom-menu:
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "My first menuitem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "My second menuitem")
'
'Add seperator to custom-menu:
Set objMenuItem = objDocMenu.MenuItems.InsertSeparator(3, "dSeparator1_3")
'
'Add submenu to custom-menu:
Set objSubMenu = objDocMenu.MenuItems.InsertSubMenu(4, "dSubMenu1_4", "My first submenu")
'
'Add menuitems to submenu:
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(5, "dmItem1_5", "My first submenuitem")
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(6, "dmItem1_6", "My second
submenuitem")
End Sub
```

Voir aussi

Objet MenuItem (Page 1967)

Propriété SymbolLibraries

Description

Renvoie une liste SymbolLibraries qui contient les objets du type "SymbolLibrary".

Utilisez SymbolLibraries(1), pour renvoyer la "bibliothèque globale". Utilisez SymbolLibraries(2), pour renvoyer la "bibliothèque du projet".

Exemple

Dans l'exemple qui suit, on obtient le nom des bibliothèques :

```
Sub ShowSymbolLibraries()
```

```
'VBA731
Dim colSymbolLibraries As HMISymbolLibraries
Dim objSymbolLibrary As HMISymbolLibrary
Set colSymbolLibraries = Application.SymbolLibraries
For Each objSymbolLibrary In colSymbolLibraries
MsgBox objSymbolLibrary.Name
Next objSymbolLibrary
End Sub
```

Voir aussi

Objet Application (Page 1876)

T

Propriété TabOrderSwitch

Description

Définit ou renvoie la position de l'objet dans la séquence TAB.

Exemple

Dans cet exemple, deux champs d'E/S sont insérés dans la vue active et la séquence TAB est définie:

```
Sub IOFieldConfig()
'VBA732
Dim objIOField1 As HMIIOField
Dim objIOField2 As HMIIOField
Set objIOField1 = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
Set objIOField2 = ActiveDocument.HMIObjects.AddHMIObject("IOField2", "HMIIOField")
With objIOField1
.Top = 10
.Left = 10
.TabOrderSwitch = 1
End With
With objIOField2
.Top = 100
.Left = 10
.TabOrderSwitch = 2
End With
End Sub
```

Voir aussi

Objet HMIObject (Page 1943)

Propriété TabOrderAllHMIObjects

Description

TRUE si tous les objets dans une vue doivent être insérés dans la séquence TAB configurée. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ConfigureTabOrder()" permet de définir quels objets de la vue active doivent être intégrés dans la séquence TAB configurée. Dans cet exemple, tous les objets sont insérés dans la séquence TAB:

```
Sub ConfigureTabOrder()
'VBA733
With ActiveDocument
.TABOrderAllHMIObjects = True
.TABOrderKeyboard = False
.TABOrderMouse = False
.TABOrderOtherAction = False
End With
End Sub
```

Voir aussi

Propriété TabOrderOtherAction (Page 2365)

Propriété TabOrderMouse (Page 2365)

Propriété TabOrderKeyboard (Page 2364)

Objet Document (Page 1908)

Propriété TabOrderAlpha

Description

Définit ou restitue la position de l'objet dans la séquence TAB pour le pointeur Alpha/Maj.

Exemple

Dans cet exemple, deux champs d'E/S sont insérés dans la vue active et la séquence TAB est définie:

```
Sub IOFieldConfig()
'VBA734
Dim objIOField1 As HMIIOField
Dim objIOField2 As HMIIOField
```

```
Set objIOField1 = ActiveDocument.HMIObjects.AddHMIObject("IOField1", "HMIIOField")
Set objIOField2 = ActiveDocument.HMIObjects.AddHMIObject("IOField2", "HMIIOField")
With objIOField1
.Top = 10
.Left = 10
.TabOrderAlpha = 1
End With
With objIOField2
.Top = 100
.Left = 10
.TabOrderAlpha = 2
End With
End Sub
```

Voir aussi

Objet Document (Page 1908)

Propriété TabOrderKeyboard

Description

TRUE si des objets doivent être intégrés dans la séquence TAB configurée, dans lesquels un événement de commande au clavier a été configuré. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ConfigureTabOrder()" permet de définir quels objets de la vue active doivent être intégrés dans la séquence TAB configurée. Dans cet exemple, les objets avec commande au clavier sont insérés dans la séquence TAB:

```
Sub ConfigureTabOrder()
'VBA735
With ActiveDocument
.TABOrderAllHMIObjects = True
.TABOrderKeyboard = False
.TABOrderMouse = False
.TABOrderOtherAction = False
End With
End Sub
```

Voir aussi

Propriété TabOrderOtherAction (Page 2365)

Propriété TabOrderMouse (Page 2365)

Propriété TabOrderAllHMIObjects (Page 2363)

Objet Document (Page 1908)

Propriété TabOrderMouse

Description

TRUE si des objets doivent être intégrés dans la séquence TAB configurée, dans lesquels un événement de commande par la souris a été configuré. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ConfigureTabOrder()" permet de définir quels objets de la vue active doivent être intégrés dans la séquence TAB configurée. Dans cet exemple, les objets avec événement de commande par la souris sont insérés dans la séquence TAB:

```
Sub ConfigureTabOrder()
'VBA736
With ActiveDocument
.TABOrderAllHMIObjects = True
.TABOrderKeyboard = False
.TABOrderMouse = False
.TABOrderOtherAction = False
End With
End Sub
```

Voir aussi

Propriété TabOrderOtherAction (Page 2365)

Propriété TabOrderKeyboard (Page 2364)

Propriété TabOrderAllHMIObjects (Page 2363)

Objet Document (Page 1908)

Propriété TabOrderOtherAction

Description

TRUE si des objets doivent être intégrés dans la séquence TAB configurée, dans lesquels un autre événement a été configuré comme commande par la souris ou commande au clavier. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ConfigureTabOrder()" permet de définir quels objets de la vue active doivent être intégrés dans la séquence TAB configurée. Dans cet exemple, des objets avec d'autres événements sont intégrés comme commande par la souris ou commande au clavier dans la séquence TAB:

```
Sub ConfigureTabOrder()
'VBA737
With ActiveDocument
.TABOrderAllHMIObjects = True
.TABOrderKeyboard = False
.TABOrderMouse = False
.TABOrderOtherAction = False
End With
End Sub
```

Voir aussi

Propriété TabOrderMouse (Page 2365)

Propriété TabOrderKeyboard (Page 2364)

Propriété TabOrderAllHMIObjects (Page 2363)

Objet Document (Page 1908)

Propriété Tag

Description

Définit ou renvoie un texte d'information pour une option de menu personnalisée ou une icône personnalisée. La propriété Tag vous permet p. ex. de décrire brièvement quel est l'effet de l'option de menu.

Exemple

Dans l'exemple qui suit, on crée dans la vue active un menu personnalisé à deux entrées et un sous-menu à deux entrées. Le sous-menu est visuellement caractérisé par une ligne de séparation :

```
Sub CreateDocumentMenus()
'VBA738
Dim objDocMenu As HMIMenu
Dim objMenuItem As HMIMenuItem
Dim objSubMenu As HMIMenuItem
'
'Add menu:
Set objDocMenu = ActiveDocument.CustomMenus.InsertMenu(1, "DocMenu1", "Doc_Menu_1")
'
```

```
'Add menuitems to custom-menu:
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(1, "dmItem1_1", "My first menuitem")
Set objMenuItem = objDocMenu.MenuItems.InsertMenuItem(2, "dmItem1_2", "My second menuitem")
'
'Add seperator to custom-menu:
Set objMenuItem = objDocMenu.MenuItems.InsertSeparator(3, "dSeparator1_3")
'
'Add submenu to custom-menu:
Set objSubMenu = objDocMenu.MenuItems.InsertSubMenu(4, "dSubMenu1_4", "My first submenu")
'
'Add menuitems to submenu:
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(5, "dmItem1_5", "My first submenuitem")
Set objMenuItem = objSubMenu.SubMenu.InsertMenuItem(6, "dmItem1_6", "My second
submenuitem")
'
'To place an additional information:
With objDocMenu
.MenuItems(1).Tag = "This is the first menuitem"
End With
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Propriété TagPrefix

Description

Définit ou renvoie le préfixe de variable qui est placé devant toutes les variables qui sont contenues dans l'objet de la fenêtre de vue.

Exemple :

Dans la fenêtre de vue doit s'afficher la vue "EntréeSortie". La vue "EntréeSortie" contient 3 champs d'E/S qui sont liés à une variable structure. La variable structure se compose des éléments ES1, ES2, ES3; un élément pour chaque champ d'E/S.

Dans le projet sont p. ex. définies trois variables structure avec les noms de structure Struct1, Struct2 et Struct3.

Le préfixe de variable est dans ce cas le nom de structure suivi d'un point. Indiquez comme préfixe de variable p. ex. Struct2. (le point est nécessaire pour repérer les éléments des variables structure avec une syntaxe correcte comme éléments de structure), puis les champs d'E/S dans la vue "EntréeSortie" sont reliés aux éléments des variables structure Struct2:

Préfixe de variable : "Struct2."

- Valeur de sortie (premier champ d'E/S) : EA1
- Valeur de sortie (deuxième champ d'E/S) : EA2
- Valeur de sortie (troisième champ d'E/S) : EA3

La liaison de variable actuelle dans la fenêtre de vue devient donc

- Valeur de sortie (premier champ d'E/S) : Struct2.EA1
- Valeur de sortie (deuxième champ d'E/S) : Struct2.EA2
- Valeur de sortie (troisième champ d'E/S) : Struct2.EA3

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est configurée :

```
Sub PictureWindowConfig()
'VBA739
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.AdaptPicture = False
.AdaptSize = False
.Caption = True
.CaptionText = "Picturewindow in runtime"
.OffsetLeft = 5
.OffsetTop = 10
'Replace the picturename "Test.PDL" with the name of
'an existing document from your "GraCS"-Folder of your active project
.PictureName = "Test.PDL"
.ScrollBars = True
.ServerPrefix = "my_Server::"
.TagPrefix = "Struct."
.UpdateCycle = 5
.Zoom = 100
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété TagScaleParam1

Description

Définit la valeur1 pour la plage de valeurs process.

Propriété TagScaleParam2

Description

Définit la valeur2 pour la plage de valeurs process.

Propriété TagScaleParam3

Description

Définit la valeur3 pour la plage de valeurs process.

Propriété TagScaleParam42

Description

Définit la valeur4 pour la plage de valeurs process.

Propriété TagStartvaluePersistence

Description

Définit si une variable interne est fixée de manière persistante. Seules les variables internes peuvent être définies comme persistantes.

Propriété Text

Description

Détermine le texte pour un objet ou le restitue.

Exemple

La procédure "ButtonConfiguration()" permet d'accéder aux propriétés du bouton. Dans cet exemple, le texte est défini :

```
Sub ButtonConfiguration()
'VBA740
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("Button1", "HMIButton")
With objButton
.Text = "Button1"
End With
End Sub
```

Voir aussi

Objet Button (Page 1886)

Objet StaticText (Page 2017)

Objet OptionGroup (Page 1977)

Objet CheckBox (Page 1889)

Propriété Toggle

Description

TRUE, lorsque le bouton ou bouton rond doit basculer en Runtime après actionnement. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "RoundButtonConfiguration()" permet d'accéder aux propriétés du bouton rond. Dans cet exemple, le bouton rond doit être basculé dans Runtime après avoir été enfoncé :

```
Sub RoundButtonConfiguration()
'VBA741
Dim objRoundButton As HMIRoundButton
Set objRoundButton = ActiveDocument.HMIObjects.AddHMIObject("RButton1", "HMIRoundButton")
With objRoundButton
.Toggle = True
End With
End Sub
```

Voir aussi

Objet RoundButton (Page 2003)

Propriété ToleranceHigh

Description

Détermine le seuil pour "Tolerance high" ou le restitue.

Déterminez avec la propriété "TypeToleranceHigh" le type d'exploitation (en pour cent ou absolu).

La surveillance du seuil n'est valable que si la propriété CheckToleranceHigh est placée sur "True".

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés des seuils. Dans cet exemple, le seuil est configuré pour "Toleranz high" :

```
Sub BarGraphLimitConfiguration()
'VBA742
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeToleranceHigh = False
'Activate monitoring
.CheckToleranceHigh = True
'Set barcolor = "yellow"
.ColorToleranceHigh = RGB(255, 255, 0)
'Set upper limit to "40"
.ToleranceHigh = 40
End With
End Sub
```

Voir aussi

Propriété TypeToleranceHigh (Page 2383)

Propriété CheckToleranceHigh (Page 2120)

Objet BarGraph (Page 1881)

Propriété ToleranceLow

Description

Détermine le seuil pour "Tolerance low" ou le restitue.

Déterminez avec la propriété "TypeToleranceLow " le type d'exploitation (en pour cent ou absolu).

La surveillance du seuil n'est valable que si la propriété CheckToleranceLow est placée sur "True".

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés des seuils. Dans cet exemple, le seuil est configuré pour "Toleranz low" :

```
Sub BarGraphLimitConfiguration()
'VBA743
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
```

```
With objBarGraph
'Set analysis = absolute
.TypeToleranceLow = False
'Activate monitoring
.CheckToleranceLow = True
'Set barcolor = "red"
.ColorToleranceLow = RGB(255, 0, 0)
'Set lower limit to "40"
.ToleranceLow = 40
End With
End Sub
```

Voir aussi

Propriété TypeToleranceLow (Page 2384)

Propriété CheckToleranceLow (Page 2121)

Objet BarGraph (Page 1881)

Propriété ToolbarItems

Description

Renvoie une liste qui contient tous les éléments (icônes et lignes de séparation) d'une barre d'outils personnalisée.

Exemple

Dans l'exemple qui suit, on crée dans la vue active une barre d'outils personnalisée à deux icônes séparées par une ligne de séparation :

```
Sub AddDocumentSpecificCustomToolbar()
'VBA744
Dim objToolbar As HMIToolbar
Dim objToolbarItem As HMIToolbarItem
Set objToolbar = ActiveDocument.CustomToolbars.Add("DocToolbar")
'
'Add symbol-icon to userdefined toolbar
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(1, "tItem1_1", "My first
symbol-icon")
Set objToolbarItem = objToolbar.ToolbarItems.InsertToolbarItem(3, "tItem1_3", "My second
symbol-icon")
Set objToolbarItem = objToolbar.ToolbarItems.InsertSeparator(2, "tSeparator1_2")
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet Toolbar (Page 2028)

Propriété TooltipText

Description

Définit ou renvoie le texte qui est affiché comme info-bulle lorsque vous passez sur l'objet (HMIObject, icône) avec la souris.

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés de l'objet rectangle. Dans cet exemple, un texte d'étiquette est affecté au rectangle :

```
Sub RectangleConfiguration()
'VBA745
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.ToolTipText = "This is a rectangle"
End With
End Sub
```

L'exemple suivant montre comment initialiser la propriété avant dynamisation :

```
Sub Dyn()
'VBA823
Dim objCircle As HMICircle
Dim doc As Document
Dim objDynDialog As HMIDynamicDialog
Set doc = ActiveDocument
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle","HMICircle")
objCircle.ObjectName = "Circle1"
objCircle.BorderColor = RGB(255, 0, 0)
objCircle.BackColor = RGB(0, 255, 0)
objCircle.ToolTipText = "Text"
Set objDynDialog =
objCircle.ToolTipText.CreateDynamic(hmiDynamicCreationTypeDynamicDialog, "'Var'")
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet HMIObject (Page 1943)

Pour dynamiser une propriété par le dialogue Dynamique (Page 1685)

Propriété Top

Description

Détermine en Pixel la coordonnée Y d'un objet (mesurée du bord gauche supérieur de la vue) ou la restitue. La coordonnée Y se réfère à l'angle gauche supérieur du rectangle entourant l'objet.

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés de l'objet rectangle. Dans cet exemple, le rectangle est placé sur la position 10/40:

```
Sub RectangleConfiguration()
'VBA746
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
.Left = 10
.Top = 40
End With
End Sub
```

Voir aussi

Objet View (Page 2050)

Objet HMIObject (Page 1943)

Propriété TopConnectedObjectName

Description

Renvoie le nom de l'objet de fin qui relie le connecteur. Accès en lecture.

Exemple :

Un exemple pour l'application de la propriété BottomConnectedObjectName se trouve dans cette documentation, à la rubrique "Objet objConnection".

Voir aussi

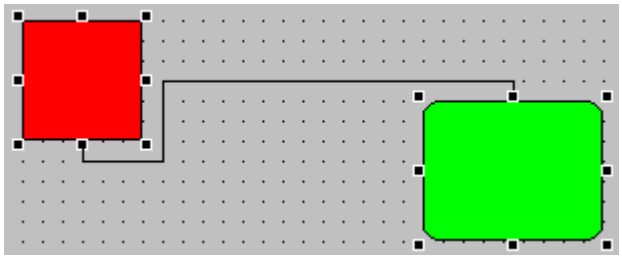
Objet objConnection (Page 1973)

Propriété TopConnectedConnectionPointIndex

Description

Renvoie le point de raccordement de l'objet auquel est relié le connecteur.

Point de raccordement	Valeur affectée
haut	0
droite	1
bas	2
gauche	3



Exemple

Un exemple pour l'application de la propriété BottomConnectedObjectName se trouve dans cette documentation, à la rubrique "Objet objConnection".

Voir aussi

Objet objConnection (Page 1973)

Propriété Transparency

Description

Définit dans quelle mesure un objet est représenté en transparence. La valeur entre 0 et 100 indique le pourcentage de transparence. Si un objet est à moitié transparent, d'autres objets sont visibles au travers. Un objet 100% transparent est invisible. Même un objet invisible reste commandable au Runtime.

Exemple

```
Sub addTransparentObject()
    'VBA849
    Dim objHMICircle As HMICircle
    Set objHMICircle = ActiveDocument.HMIObjects.AddHMIObject("Circle", "HMICircle")
    objHMICircle.Transparency = 40
End Sub
```

Propriété Trend

Description

TRUE si la tendance (croissante ou descendante) de la valeur de mesure à surveiller doit être affichée avec une petite flèche. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple est affichée la tendance de la valeur de mesure :

```
Sub BarGraphConfiguration()
'VBA747
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.trend = True
End With
End Sub
```

Voir aussi

Objet BarGraph (Page 1881)

Propriété TrendColor

Description

Détermine la couleur de la tendance ou la restitue.

La tendance (croissante ou décroissante) de la valeur à surveiller est visualisée par une petite flèche. La propriété Trend doit être positionnée sur "True" pour activer son affichage. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple est affichée la tendance de la valeur de mesure ; l'affichage de la tendance est défini sur "Rouge" :

```
Sub BarGraphConfiguration()
'VBA748
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.trend = True
.TrendColor = RGB(255, 0, 0)
End With
End Sub
```

Voir aussi

Propriété Trend (Page 2376)

Objet BarGraph (Page 1881)

Propriété Trigger

Description

Renvoie un objet Trigger. Utilisez la propriété Trigger, si vous dynamiser une propriété avec un script.

Exemple

Dans cet exemple, la propriété "Radius" d'un cercle est dynamisée avec un script en C (la valeur de retour active le rayon):

```
Sub AddDynamicAsCSkriptToProperty()
'VBA749
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle

Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("myCircle", "HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
```

```
With objVBScript
.Trigger.Type = hmiTriggerTypeStandardCycle
.Trigger.CycleType = hmiCycleType_2s
.Trigger.Name = "Trigger1"
End With
End Sub
```

Voir aussi

Objet Trigger (Page 2035)

Objet ScriptInfo (Page 2009)

Propriété Type

Description

Renvoie le type d'un objet ou le définit.

Le type d'objet est renvoyé soit comme chaîne soit comme entier.

Exemple

La procédure "RectangleConfiguration()" permet d'accéder aux propriétés de l'objet rectangle. Dans cet exemple, le type d'objet est sorti :

```
Sub RectangleConfiguration()
'VBA750
Dim objRectangle As HMIRectangle
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("Rectangle1", "HMIRectangle")
With objRectangle
MsgBox "Objecttype: " & .Type
End With
End Sub
```

Voir aussi

Objet Trigger (Page 2035)

Objet ToolbarItem (Page 2031)

Objet SourceLink (Page 2016)

Objet Property (Page 1993)

Objet HMIObject (Page 1943)

Objet FolderItem (Page 1927)

Objet DestLink (Page 1904)

Propriété TypeAlarmHigh

Description

TRUE, lorsque le seuil supérieur déclenchant l'alarme est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "50".

```
Sub BarGraphLimitConfiguration()
'VBA751
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeAlarmHigh = False
'Activate monitoring
.CheckAlarmHigh = True
'Set barcolor = "yellow"
.ColorAlarmHigh = RGB(255, 255, 0)
'Set upper limit = "50"
.AlarmHigh = 50
End With
End Sub
```

Voir aussi

Propriété ColorAlarmHigh (Page 2126)

Propriété CheckAlarmHigh (Page 2115)

Propriété AlarmHigh (Page 2061)

Objet BarGraph (Page 1881)

Propriété TypeAlarmLow

Description

TRUE, lorsque le seuil inférieur déclenchant l'alarme est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit être déclenchée à la valeur "10".

```
Sub BarGraphLimitConfiguration()
'VBA752
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeAlarmLow = False
'Activate monitoring
.CheckAlarmLow = True
'Set barcolor = "yellow"
.ColorAlarmLow = RGB(255, 255, 0)
'Set lower limit = "10"
.AlarmLow = 10
End With
End Sub
```

Voir aussi

Propriété ColorAlarmLow (Page 2127)

Propriété CheckAlarmLow (Page 2115)

Propriété AlarmLow (Page 2062)

Objet BarGraph (Page 1881)

Propriété TypeLimitHigh4

Description

TRUE, lorsque le seuil supérieur "Reserve 4" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "70".

```
Sub BarGraphLimitConfiguration()
'VBA753
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
```

```
'Set analysis = absolute
.TypeLimitHigh4 = False
'Activate monitoring
.CheckLimitHigh4 = True
'Set barcolor = "red"
.ColorLimitHigh4 = RGB(255, 0, 0)
'Set upper limit = "70"
.LimitHigh4 = 70
End With
End Sub
```

Voir aussi

Propriété LimitHigh4 (Page 2255)

Propriété CheckLimitHigh4 (Page 2117)

Objet BarGraph (Page 1881)

Propriété TypeLimitHigh5

Description

TRUE, lorsque le seuil supérieur "Reserve 5" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "80".

```
Sub BarGraphLimitConfiguration()
'VBA754
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeLimitHigh5 = False
'Activate monitoring
.CheckLimitHigh5 = True
'Set barcolor = "black"
.ColorLimitHigh5 = RGB(0, 0, 0)
'Set upper limit = "70"
.LimitHigh5 = 70
End With
End Sub
```

Voir aussi

Propriété LimitHigh5 (Page 2256)

Propriété CheckLimitHigh5 (Page 2118)

Objet BarGraph (Page 1881)

Propriété TypeLimitLow4

Description

TRUE, lorsque le seuil inférieur "Reserve 4" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "5".

```
Sub BarGraphLimitConfiguration()
'VBA755
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeLimitLow4 = False
'Activate monitoring
.CheckLimitLow4 = True
'Set barcolor = "green"
.ColorLimitLow4 = RGB(0, 255, 0)
'Set lower limit = "5"
.LimitLow4 = 5
End With
End Sub
```

Voir aussi

Propriété LimitLow4 (Page 2257)

Propriété ColorLimitLow4 (Page 2131)

Propriété CheckLimitLow4 (Page 2119)

Objet BarGraph (Page 1881)

Propriété TypeLimitLow5

Description

TRUE, lorsque le seuil inférieur "Reserve 5" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "0".

```
Sub BarGraphLimitConfiguration()
'VBA756
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeLimitLow5 = False
'Activate monitoring
.CheckLimitLow5 = True
'Set barcolor = "white"
.ColorLimitLow5 = RGB(255, 255, 255)
'Set lower limit = "0"
.LimitLow5 = 0
End With
End Sub
```

Voir aussi

Propriété LimitLow5 (Page 2257)

Propriété ColorLimitLow5 (Page 2132)

Propriété CheckLimitLow5 (Page 2119)

Objet BarGraph (Page 1881)

Propriété TypeToleranceHigh

Description

TRUE, lorsque le seuil inférieur "ToleranceHigh" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés des seuils. Dans cet exemple, le seuil est configuré pour "Toleranz high" :

```
Sub BarGraphLimitConfiguration()
'VBA757
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeToleranceHigh = False
'Activate monitoring
.CheckToleranceHigh = True
'Set barcolor = "yellow"
.ColorToleranceHigh = RGB(255, 255, 0)
'Set upper limit = "40"
.ToleranceHigh = 40
End With
End Sub
```

Voir aussi

Propriété ColorToleranceHigh (Page 2133)

Propriété CheckToleranceHigh (Page 2120)

Objet BarGraph (Page 1881)

Propriété TypeToleranceLow

Description

TRUE, lorsque le seuil inférieur "ToleranceLow" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés des seuils. Dans cet exemple, le seuil est configuré pour "Toleranz low" :

```
Sub BarGraphLimitConfiguration()
'VBA758
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeToleranceLow = False
'Activate monitoring
.CheckToleranceLow = True
```

```
'Set barcolor = "red"
.ColorToleranceLow = RGB(255, 0, 0)
'Set lower limit = "10"
.ToleranceLow = 10
End With
End Sub
```

Voir aussi

Propriété ToleranceLow (Page 2371)

Propriété ColorToleranceLow (Page 2134)

Propriété CheckToleranceLow (Page 2121)

Objet BarGraph (Page 1881)

Propriété TypeWarningHigh

Description

TRUE, lorsque le seuil inférieur "WarningHigh" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "75".

```
Sub BarGraphLimitConfiguration()
'VBA759
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeWarningHigh = False
'Activate monitoring
.CheckWarningHigh = True
'Set barcolor = "red"
.ColorWarningHigh = RGB(255, 0, 0)
'Set upper limit = "75"
.WarningHigh = 75
End With
End Sub
```

Voir aussi

Propriété WarningHigh (Page 2470)

Propriété ColorWarningHigh (Page 2135)

Propriété CheckWarningHigh (Page 2122)

Objet BarGraph (Page 1881)

Propriété TypeWarningLow

Description

TRUE, lorsque le seuil inférieur "WarningLow" est interprété en pour cent. FALSE, lorsque l'interprétation doit se faire en valeur absolue. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "12".

```
Sub BarGraphLimitConfiguration()
'VBA760
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeWarningLow = False
'Activate monitoring
.CheckWarningLow = True
'Set barcolor = "magenta"
.ColorWarningLow = RGB(255, 0, 255)
'Set lower limit = "12"
.WarningLow = 12
End With
End Sub
```

Voir aussi

Propriété WarningLow (Page 2471)

Propriété ColorWarningLow (Page 2136)

Propriété CheckWarningLow (Page 2123)

Objet BarGraph (Page 1881)

U

Propriété UsedLanguage

Description

La propriété UsedLanguage permet de paramétrer l'identificateur de langue LanguageID, ce qui sélectionne la page de code appropriée pour le jeu de caractères utilisé.

Avec la propriété UsedLanguage, vous paramétrez la page de code correspondant au jeu de caractères utilisé.

LONG Accès en écriture et en lecture.

Exemple

Dans l'exemple qui suit, on définit la page de code à Anglais Etats-Unis au moyen de la propriété "UsedLanguage" et du code de langue "1033".

```
Sub Create_Cycle_with_Dynamic()
'VBA61

Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
Dim strCode As String

strCode = "long lHeight;" & vbCrLf & "int check;" & vbCrLf
strCode = strCode & "GetHeight( ""events.PDL"", ""myCircle"" );" & vbCrLf
strCode = strCode & "lHeight = lHeight+5;" & vbCrLf
strCode = strCode & "check = SetHeight(""events.PDL"", ""myCircle"", lHeight );"
strCode = strCode & vbCrLf & "//Return-Type: BOOL" & vbCrLf
strCode = strCode & "return check;"
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("myCircle", "HMICircle")
'Create dynamic for Property "Height":
Set objCScript = objCircle.Height.CreateDynamic(hmiDynamicCreationTypeCScript)
'
'set Sourcecode and cycletime:
With objCScript
.SourceCode = strCode
.Trigger.Type = hmiTriggerTypeStandardCycle
.Trigger.CycleType = hmiCycleType_2s
.Trigger.Name = "Trigger1"
'Set language English-US
.UsedLanguage = 1033
End With

End Sub
```

Propriété Underlined

Description

TRUE si l'attribut de police souligné est réglé pour le texte en fonction de la langue dans l'objet. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple suivant, on configure les attributs de texte d'un bouton pour le français et pour l'anglais :

```
Sub ExampleForLanguageFonts()
'VBA761
Dim colLangFonts As HMILanguageFonts
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton", "HMIButton")
objButton.Text = "DefText"
Set colLangFonts = objButton.LDFonts
'
'Set font-properties for french:
With colLangFonts.ItemByLCID(1036)
.Family = "Courier New"
.Bold = True
.Italic = False
.Underlined = True
.Size = 12
End With
'
'Set font-properties for english:
With colLangFonts.ItemByLCID(1033)
.Family = "Times New Roman"
.Bold = False
.Italic = True
.Underlined = False
.Size = 14
End With
End Sub
```

Voir aussi

Propriété Size (Page 2353)

Propriété Parent (Page 2303)

Propriété LanguageID (Page 2219)

Propriété Italic (Page 2212)

Propriété FontFamily (Page 2189)

Propriété Bold (Page 2094)

Propriété Application (Page 2067)

Objet LanguageFont (Page 1950)

Propriété UnselBGColor

Description

Détermine pour l'objet TextList la couleur d'arrière-plan des inscriptions non sélectionnées sur la liste de sélection, ou la restitue. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple sont définies les couleurs pour les entrées non sélectionnées dans la liste de sélection :

```
Sub TextListConfiguration()
'VBA762
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.UnselBGColor = RGB(255, 0, 0)
.UnselTextColor = RGB(0, 0, 0)
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété UnselTextColor

Description

Définit ou renvoie pour l'objet TextList la couleur du texte de la liste de sélection pour les entrées non sélectionnées. LONG Accès en écriture et en lecture.

Calcul de la valeur chromatique

La couleur est représentée en format RVB (Rouge, Vert, Bleu). Indiquez pour chacune des trois valeurs RVB, la valeur décimale correspondante (plage de valeurs : 0 à 255).

Utilisez la fonction VBA "RGB" pour affecter une couleur à une propriété. La couleur "Rouge" est par exemple représentée comme suit : RVB(255, 0, 0)

Exemple

La procédure "TextListConfiguration()" permet d'accéder aux propriétés de l'objet TextList. Dans cet exemple sont définies les couleurs pour les entrées non sélectionnées dans la liste de sélection :

```
Sub TextListConfiguration()
'VBA763
Dim objTextList As HMITextList
Set objTextList = ActiveDocument.HMIObjects.AddHMIObject("myTextList", "HMITextList")
With objTextList
.UnselBGColor = RGB(255, 0, 0)
.UnselTextColor = RGB(0, 0, 0)
End With
End Sub
```

Voir aussi

Objet TextList (Page 2025)

Propriété UpdateCycle

Description

Définit ou renvoie le type et la fréquence de rafraîchissement de la fenêtre de vue dans Runtime.

Cycle de rafraîchissement	Valeur affectée
En cas de modification	0
250 ms	1
500 ms	2
1 s	3
2 s	4

Cycle de rafraîchissement	Valeur affectée
5 s	5
10 s	6
1 min	7
5 min	8
10 min	9
1 h	10
Cycle utilisateur 1	11
Cycle utilisateur 2	12
Cycle utilisateur 3	13
Cycle utilisateur 4	14
Cycle utilisateur 5	15
Cycle vue	255

Exemple

La procédure "PictureWindowConfig" permet d'accéder aux propriétés de la fenêtre de vue. Dans cet exemple, la fenêtre de vue est rafraîchie dans Runtime toutes les cinq secondes:

```
Sub PictureWindowConfig()
'VBA764
Dim objPicWindow As HMIPictureWindow
Set objPicWindow = ActiveDocument.HMIObjects.AddHMIObject("PicWindow1", "HMIPictureWindow")
With objPicWindow
.UpdateCycle = 5
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Propriété UserValue1

Description

Définit ou renvoie pour l'objet GroupDisplay une valeur quelconque.

Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, quatre valeurs d'utilisateurs différentes sont affectées :

```
Sub GroupDisplayConfiguration()
'VBA765
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.UserValue1 = 0
.UserValue2 = 25
.UserValue3 = 50
.UserValue4 = 75
End With
End Sub
```

Voir aussi

Propriété UserValue4 (Page 2394)

Propriété UserValue3 (Page 2393)

Propriété UserValue2 (Page 2392)

Objet GroupDisplay (Page 1935)

Propriété UserValue2

Description

Définit ou renvoie pour l'objet GroupDisplay une valeur quelconque.

Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, quatre valeurs d'utilisateurs différentes sont affectées :

```
Sub GroupDisplayConfiguration()
'VBA766
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
```

```
.UserValue1 = 0
.UserValue2 = 25
.UserValue3 = 50
.UserValue4 = 75
End With
End Sub
```

Voir aussi

Propriété UserValue4 (Page 2394)

Propriété UserValue3 (Page 2393)

Propriété UserValue1 (Page 2391)

Objet GroupDisplay (Page 1935)

Propriété UserValue3

Description

Définit ou renvoie pour l'objet GroupDisplay une valeur quelconque.

Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, quatre valeurs d'utilisateurs différentes sont affectées :

```
Sub GroupDisplayConfiguration()
'VBA767
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.UserValue1 = 0
.UserValue2 = 25
.UserValue3 = 50
.UserValue4 = 75
End With
End Sub
```

Voir aussi

Propriété UserValue4 (Page 2394)

Propriété UserValue2 (Page 2392)

Propriété UserValue1 (Page 2391)

Objet GroupDisplay (Page 1935)

Propriété UserValue4

Description

Définit ou renvoie pour l'objet GroupDisplay une valeur quelconque.

Vous pouvez par exemple faire analyser la valeur par un Script. En Runtime, cette valeur n'est ni exploitée, ni affichée.

Exemple

La procédure "GroupDisplayConfiguration()" permet d'accéder aux propriétés de la signalisation groupée. Dans cet exemple, quatre valeurs d'utilisateurs différentes sont affectées :

```
Sub GroupDisplayConfiguration()
'VBA768
Dim objGroupDisplay As HMIGroupDisplay
Set objGroupDisplay = ActiveDocument.HMIObjects.AddHMIObject("GroupDisplay1",
"HMIGroupDisplay")
With objGroupDisplay
.UserValue1 = 0
.UserValue2 = 25
.UserValue3 = 50
.UserValue4 = 75
End With
End Sub
```

Voir aussi

Propriété UserValue3 (Page 2393)

Propriété UserValue2 (Page 2392)

Propriété UserValue1 (Page 2391)

Objet GroupDisplay (Page 1935)

V

Value

Propriété Value

Description

Renvoie la valeur d'une propriété ou la définit.

Exemple

Utilisez la propriété Value si vous voulez renvoyer ou définir une valeur par la liste des propriétés. Dans cet exemple, la propriété Value permet d'accéder à la propriété d'un ActiveX Control :

```
Sub AddActiveXControl()
'VBA769
Dim objActiveXControl As HMIActiveXControl
Set objActiveXControl = ActiveDocument.HMIObjects.AddActiveXControl("WinCC_Gauge2",
"XGAUGE.XGaugeCtrl.1")
'
'Move ActiveX-Control:
objActiveXControl.Top = 40
objActiveXControl.Left = 60
'
'Modify individual properties:
objActiveXControl.Properties("BackColor").value = RGB(255, 0, 0)
End Sub
```

Voir aussi

Objet Property (Page 1993)

Propriété VALUE_ACCESS_FAULT

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Accès à la variable non autorisé !" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Objet VariableStateValue (Page 2045)

Propriété VALUE_ADDRESS_ERROR

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Erreur d'adressage" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA771
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_BAD_COMMLUV

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, no communication (last usable value)" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA818
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
```

```
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_BAD_DEVICE (Page 2404)

Propriété QualityCodeStateChecked (Page 2328)

Propriété VALUE_UNCERT_SUBSTSET (Page 2461)

Propriété VALUE_UNCERT_SIMVAL (Page 2459)

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)

Propriété VALUE_UNCERT_MISCSTATES (Page 2453)

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)

Propriété VALUE_UNCERT_LUV (Page 2449)

Propriété VALUE_UNCERT_INITVAL (Page 2447)

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)

Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)

Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)

Propriété VALUE_LOWLIMITED (Page 2427)

Propriété VALUE_HIGHLIMITED (Page 2423)

Propriété VALUE_BAD_PROCRELSUB (Page 2416)

Propriété VALUE_BAD_PROCRELNOM (Page 2414)

Propriété VALUE_BAD_OUTOFSERV (Page 2412)

Propriété VALUE_BAD_NOTCONNECTED (Page 2410)

Propriété VALUE_BAD_NONSPECIFIC (Page 2408)

Propriété VALUE_BAD_MISCSTATES (Page 2406)

Propriété VALUE_BAD_CONFERROR (Page 2402)

Propriété VALUE_BAD_COMMNUV (Page 2400)

Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_COMMNUV

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, no communication (no usable value)" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété QualityCodeStateChecked (Page 2328)

Propriété VALUE_UNCERT_SUBSTSET (Page 2461)

Propriété VALUE_UNCERT_SIMVAL (Page 2459)

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)

Propriété VALUE_UNCERT_MISCSTATES (Page 2453)

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)

Propriété VALUE_UNCERT_LUV (Page 2449)

Propriété VALUE_UNCERT_INITVAL (Page 2447)

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)

Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)

Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)

Propriété VALUE_LOWLIMITED (Page 2427)

Propriété VALUE_HIGHLIMITED (Page 2423)

Propriété VALUE_BAD_PROCRELSUB (Page 2416)

Propriété VALUE_BAD_PROCRELNOM (Page 2414)

Propriété VALUE_BAD_OUTOFSERV (Page 2412)

Propriété VALUE_BAD_NOTCONNECTED (Page 2410)

Propriété VALUE_BAD_NONSPECIFIC (Page 2408)

Propriété VALUE_BAD_MISCSTATES (Page 2406)

Propriété VALUE_BAD_DEVICE (Page 2404)

Propriété VALUE_BAD_CONFERROR (Page 2402)

Propriété VALUE_BAD_COMMLUV (Page 2399)

Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_CONFERROR

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, configuration error, value not accepted" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_MISCSTATES (Page 2453)

Propriété QualityCodeStateChecked (Page 2328)

Propriété VALUE_UNCERT_SUBSTSET (Page 2461)

Propriété VALUE_UNCERT_SIMVAL (Page 2459)

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)

Propriété VALUE_UNCERT_LUV (Page 2449)

Propriété VALUE_UNCERT_INITVAL (Page 2447)

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)

Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)

Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)

Propriété VALUE_LOWLIMITED (Page 2427)

Propriété VALUE_HIGHLIMITED (Page 2423)

Propriété VALUE_BAD_PROCRELSUB (Page 2416)

Propriété VALUE_BAD_PROCRELNOM (Page 2414)

Propriété VALUE_BAD_OUTOFSERV (Page 2412)

Propriété VALUE_BAD_NOTCONNECTED (Page 2410)

Propriété VALUE_BAD_NONSPECIFIC (Page 2408)

Propriété VALUE_BAD_MISCSTATES (Page 2406)

Propriété VALUE_BAD_DEVICE (Page 2404)

Propriété VALUE_BAD_COMMNUV (Page 2400)

Propriété VALUE_BAD_COMMLUV (Page 2399)

Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_DEVICE

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, device failure" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)

Propriété QualityCodeStateChecked (Page 2328)

Propriété VALUE_UNCERT_SUBSTSET (Page 2461)

Propriété VALUE_UNCERT_SIMVAL (Page 2459)

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)

Propriété VALUE_UNCERT_MISCSTATES (Page 2453)

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)

Propriété VALUE_UNCERT_LUV (Page 2449)

Propriété VALUE_UNCERT_INITVAL (Page 2447)

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)

Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)

Propriété VALUE_LOWLIMITED (Page 2427)

Propriété VALUE_HIGHLIMITED (Page 2423)

Propriété VALUE_BAD_PROCRELSUB (Page 2416)

Propriété VALUE_BAD_PROCRELNOM (Page 2414)

Propriété VALUE_BAD_OUTOFSERV (Page 2412)

Propriété VALUE_BAD_NOTCONNECTED (Page 2410)

Propriété VALUE_BAD_NONSPECIFIC (Page 2408)

Propriété VALUE_BAD_MISCSTATES (Page 2406)

Propriété VALUE_BAD_CONFERROR (Page 2402)

Propriété VALUE_BAD_COMMNUV (Page 2400)

Propriété VALUE_BAD_COMMLUV (Page 2399)

Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_MISCSTATES

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad miscellanous states" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_NONSPECIFIC

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, non-specific" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_NOTCONNECTED

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, not connected" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_HIGHLIMITED (Page 2423)

Propriété QualityCodeStateChecked (Page 2328)

Propriété VALUE_UNCERT_SUBSTSET (Page 2461)

Propriété VALUE_UNCERT_SIMVAL (Page 2459)

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)

Propriété VALUE_UNCERT_MISCSTATES (Page 2453)

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)

Propriété VALUE_UNCERT_LUV (Page 2449)

Propriété VALUE_UNCERT_INITVAL (Page 2447)

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)

Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)

Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)

Propriété VALUE_LOWLIMITED (Page 2427)

Propriété VALUE_BAD_PROCRELSUB (Page 2416)

Propriété VALUE_BAD_PROCRELNOM (Page 2414)

Propriété VALUE_BAD_OUTOFSERV (Page 2412)

Propriété VALUE_BAD_NONSPECIFIC (Page 2408)

Propriété VALUE_BAD_MISCSTATES (Page 2406)

Propriété VALUE_BAD_DEVICE (Page 2404)

Propriété VALUE_BAD_CONFERROR (Page 2402)

Propriété VALUE_BAD_COMMNUV (Page 2400)

Propriété VALUE_BAD_COMMLUV (Page 2399)

Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_OUTOFSERV

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, out of service" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_BAD_CONFERROR (Page 2402)

Propriété QualityCodeStateChecked (Page 2328)

Propriété VALUE_UNCERT_SUBSTSET (Page 2461)

Propriété VALUE_UNCERT_SIMVAL (Page 2459)

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)

Propriété VALUE_UNCERT_MISCSTATES (Page 2453)

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)

Propriété VALUE_UNCERT_LUV (Page 2449)

Propriété VALUE_UNCERT_INITVAL (Page 2447)

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)

Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)

Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)

Propriété VALUE_LOWLIMITED (Page 2427)

Propriété VALUE_HIGHLIMITED (Page 2423)

Propriété VALUE_BAD_PROCRELSUB (Page 2416)

Propriété VALUE_BAD_PROCRELNOM (Page 2414)

Propriété VALUE_BAD_NOTCONNECTED (Page 2410)

Propriété VALUE_BAD_NONSPECIFIC (Page 2408)

Propriété VALUE_BAD_MISCSTATES (Page 2406)

Propriété VALUE_BAD_DEVICE (Page 2404)

Propriété VALUE_BAD_COMMNUV (Page 2400)

Propriété VALUE_BAD_COMMLUV (Page 2399)

Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_PROCRELNOM

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, process related, no maintenance" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_BAD_PROCRELSUB

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "bad, process related, substitute value" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_CONVERSION_ERROR

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Erreur de transtypage" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA772
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)

Propriété VALUE_TIMEOUT (Page 2440)

Propriété VALUE_STARTUP_VALUE (Page 2439)

Propriété VALUE_SERVERDOWN (Page 2437)

Propriété VALUE_NOT_ESTABLISHED (Page 2436)

Propriété VALUE_MIN_RANGE (Page 2434)

Propriété VALUE_MIN_LIMIT (Page 2433)

Propriété VALUE_MAX_RANGE (Page 2431)

Propriété VALUE_MAX_LIMIT (Page 2429)

Propriété VALUE_INVALID_KEY (Page 2425)

Propriété VALUE_HARDWARE_ERROR (Page 2422)

Propriété VALUE_HANDSHAKE_ERROR (Page 2420)

Propriété VALUE_ADDRESS_ERROR (Page 2397)

Propriété VALUE_ACCESS_FAULT (Page 2395)

Objet VariableStateValue (Page 2045)

Propriété VALUE_HANDSHAKE_ERROR

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Erreur de protocole" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA773
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_HARDWARE_ERROR

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Pas de carte réseau" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA774
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_HIGHLIMITED

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "high limited" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_INVALID_KEY

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Variable introuvable" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA775
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_LOWLIMITED

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "low limited" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_MAX_LIMIT

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Limite supérieure dépassée" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA776
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_MAX_RANGE

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Limite supérieure de format dépassée" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA777
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_MIN_LIMIT

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Limite inférieure dépassée" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA778
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)

Propriété VALUE_TIMEOUT (Page 2440)

Propriété VALUE_STARTUP_VALUE (Page 2439)

Propriété VALUE_SERVERDOWN (Page 2437)

Propriété VALUE_NOT_ESTABLISHED (Page 2436)

Propriété VALUE_MIN_RANGE (Page 2434)

Propriété VALUE_MAX_RANGE (Page 2431)

Propriété VALUE_MAX_LIMIT (Page 2429)

Propriété VALUE_INVALID_KEY (Page 2425)

Propriété VALUE_HARDWARE_ERROR (Page 2422)

Propriété VALUE_HANDSHAKE_ERROR (Page 2420)

Propriété VALUE_CONVERSION_ERROR (Page 2418)

Propriété VALUE_ADDRESS_ERROR (Page 2397)

Propriété VALUE_ACCESS_FAULT (Page 2395)

Objet VariableStateValue (Page 2045)

Propriété VALUE_MIN_RANGE

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Limite inférieure de format dépassée" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA779
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_NOT_ESTABLISHED

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Pas de réponse du canal" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA780
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_SERVERDOWN

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Serveur non disponible" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA781
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_TIMEOUT (Page 2440)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_STARTUP_VALUE

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Valeur de démarrage" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA782
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)

Propriété VALUE_TIMEOUT (Page 2440)

Propriété VALUE_SERVERDOWN (Page 2437)

Propriété VALUE_NOT_ESTABLISHED (Page 2436)

Propriété VALUE_MIN_RANGE (Page 2434)

Propriété VALUE_MIN_LIMIT (Page 2433)

Propriété VALUE_MAX_RANGE (Page 2431)

Propriété VALUE_MAX_LIMIT (Page 2429)

Propriété VALUE_INVALID_KEY (Page 2425)

Propriété VALUE_HARDWARE_ERROR (Page 2422)

Propriété VALUE_HANDSHAKE_ERROR (Page 2420)

Propriété VALUE_CONVERSION_ERROR (Page 2418)

Propriété VALUE_ADDRESS_ERROR (Page 2397)

Propriété VALUE_ACCESS_FAULT (Page 2395)

Objet VariableStateValue (Page 2045)

Propriété VALUE_TIMEOUT

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si l'état de la variable "Pas de liaison" est présent.

Pour que l'état puisse être analysé, la propriété VariableStateChecked doit avoir la valeur TRUE.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie pas d'état, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA783
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
```

```
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Propriété VariableStateChecked (Page 2464)
Propriété VALUE_STARTUP_VALUE (Page 2439)
Propriété VALUE_SERVERDOWN (Page 2437)
Propriété VALUE_NOT_ESTABLISHED (Page 2436)
Propriété VALUE_MIN_RANGE (Page 2434)
Propriété VALUE_MIN_LIMIT (Page 2433)
Propriété VALUE_MAX_RANGE (Page 2431)
Propriété VALUE_MAX_LIMIT (Page 2429)
Propriété VALUE_INVALID_KEY (Page 2425)
Propriété VALUE_HARDWARE_ERROR (Page 2422)
Propriété VALUE_HANDSHAKE_ERROR (Page 2420)
Propriété VALUE_CONVERSION_ERROR (Page 2418)
Propriété VALUE_ADDRESS_ERROR (Page 2397)
Propriété VALUE_ACCESS_FAULT (Page 2395)
Objet VariableStateValue (Page 2045)

Propriété VALUE_UNCERT_ENGVHIGHLIM

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, engineering unit range violation, high limit set" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
```

```
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_ENGVLOWLIM

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, engineering unit range violation, low limit set" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_ENGVONLIM

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, engineering unit range violation, on limits set" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_INITVAL

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, initial value" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_LUV

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, last usable value" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_MAINTDEM

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, maintenance demanded" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_MISCSTATES

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain miscellanous states" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_LOWLIMITED (Page 2427)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_NONSPECIFIC

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, non-specific" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_PROCRELNOM

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, process related, no maintenance" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_SIMVAL (Page 2459)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_SIMVAL

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, simulated value" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_BAD_PROCRELNOM (Page 2414)
Propriété QualityCodeStateChecked (Page 2328)
Propriété VALUE_UNCERT_SUBSTSET (Page 2461)
Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)
Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)
Propriété VALUE_UNCERT_MISCSTATES (Page 2453)
Propriété VALUE_UNCERT_MAINTDEM (Page 2451)
Propriété VALUE_UNCERT_LUV (Page 2449)
Propriété VALUE_UNCERT_INITVAL (Page 2447)
Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)
Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)
Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)
Propriété VALUE_LOWLIMITED (Page 2427)
Propriété VALUE_HIGHLIMITED (Page 2423)
Propriété VALUE_BAD_PROCRELSUB (Page 2416)
Propriété VALUE_BAD_OUTOFSERV (Page 2412)
Propriété VALUE_BAD_NOTCONNECTED (Page 2410)
Propriété VALUE_BAD_NONSPECIFIC (Page 2408)
Propriété VALUE_BAD_MISCSTATES (Page 2406)
Propriété VALUE_BAD_DEVICE (Page 2404)
Propriété VALUE_BAD_CONFERROR (Page 2402)
Propriété VALUE_BAD_COMMNUV (Page 2400)
Propriété VALUE_BAD_COMMLUV (Page 2399)
Objets QualityCodeStateValue (Page 1995)

Propriété VALUE_UNCERT_SUBSTSET

Description

Définit ou renvoie la valeur qui est affectée à la propriété dynamisée si le code qualité "uncertain, substitute set" est présent.

Pour que le code qualité puisse être analysé, la propriété QualityCodeStateChecked doit avoir la valeur VRAI.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse du code qualité d'une variable. Si la variable ne renvoie pas de code qualité, une valeur de substitution (propriété ElseCase) sera définie :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA770
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of qualitycodestate
.QualityCodeStateChecked = True
End With
With objDynDialog.QualityCodeStateValues(1)
'
'define a value for every state:
.VALUE_BAD_COMMLUV = 20
.VALUE_BAD_COMMNUV = 30
.VALUE_BAD_CONFERROR = 40
.VALUE_BAD_DEVICE = 60
.VALUE_BAD_MISCSTATES = 70
.VALUE_BAD_NONSPECIFIC = 80
.VALUE_BAD_NOTCONNECTED = 90
.VALUE_BAD_OUTOFSERV = 100
.VALUE_BAD_PROCRELNOM = 110
.VALUE_BAD_PROCRELSUB = 120
.VALUE_HIGHLIMITED = 130
.VALUE_LOWLIMITED = 140
.VALUE_UNCERT_ENGVHIGHLIM = 150
.VALUE_UNCERT_ENGVLOWLIM = 160
.VALUE_UNCERT_INITVAL = 170
.VALUE_UNCERT_LUV = 180
.VALUE_UNCERT_MAINTDEM = 190
.VALUE_UNCERT_MISCSTATES = 200
.VALUE_UNCERT_NONSPECIFIC = 210
.VALUE_UNCERT_PROCRELNOM = 220
.VALUE_UNCERT_SIMVAL = 230
.VALUE_UNCERT_SUBSTSET = 240
End With
End Sub
```

Voir aussi

Propriété VALUE_UNCERT_MISCSTATES (Page 2453)

Propriété QualityCodeStateChecked (Page 2328)

Propriété VALUE_UNCERT_SIMVAL (Page 2459)

Propriété VALUE_UNCERT_PROCRELNOM (Page 2457)

Propriété VALUE_UNCERT_NONSPECIFIC (Page 2455)

Propriété VALUE_UNCERT_MAINTDEM (Page 2451)

Propriété VALUE_UNCERT_LUV (Page 2449)

Propriété VALUE_UNCERT_INITVAL (Page 2447)

Propriété VALUE_UNCERT_ENGVONLIM (Page 2445)

Propriété VALUE_UNCERT_ENGVLOWLIM (Page 2443)

Propriété VALUE_UNCERT_ENGVHIGHLIM (Page 2442)

Propriété VALUE_LOWLIMITED (Page 2427)

Propriété VALUE_HIGHLIMITED (Page 2423)

Propriété VALUE_BAD_PROCRELSUB (Page 2416)

Propriété VALUE_BAD_PROCRELNOM (Page 2414)

Propriété VALUE_BAD_OUTOFSERV (Page 2412)

Propriété VALUE_BAD_NOTCONNECTED (Page 2410)

Propriété VALUE_BAD_NONSPECIFIC (Page 2408)

Propriété VALUE_BAD_MISCSTATES (Page 2406)

Propriété VALUE_BAD_DEVICE (Page 2404)

Propriété VALUE_BAD_CONFERROR (Page 2402)

Propriété VALUE_BAD_COMMNUV (Page 2400)

Propriété VALUE_BAD_COMMLUV (Page 2399)

Objets QualityCodeStateValue (Page 1995)

Propriété VariablesExist

Description

TRUE si toutes les variables utilisées dans le code source d'un dialogue dynamique sont définies. Accès en lecture.

Cette propriété vous permet de contrôler si toutes les variables que vous avez définies dans le code source du dialogue dynamique sont créées dans WinCC.

Exemple

--

Voir aussi

Objet DynamicDialog (Page 1912)

Propriété VariableStateChecked

Description

VRAI si l'état de la variable indiquée dans le dialogue dynamique est utilisé pour la dynamisation. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie aucun état, une valeur de substitution (propriété ElseCase) est définie, un nom de variable est donné et trois plages de valeurs analogiques sont créées :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA785
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Objet DynamicDialog (Page 1912)

Propriété VariableStateType

Description

Renvoie le type de surveillance de variable par lequel une propriété ou un événement ont été dynamisés : aucune surveillance, Code qualité ou Etat de la variable. Accès en lecture.

Indice	VariableStateType
0	hmiNoVariableState
1	hmiVariableQCState
2	hmiVariableState

Exemple

La procédure "GetVariableStateType()" lit le type de surveillance dans le document actuel. Dans cet exemple, le type de surveillance est affiché dans un message :

```
Sub GetVariableStateType()
'VBA819
Dim objDyn As HMIDynamicDialog
Set objDyn =
ActiveDocument.Properties("Width").CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'TestVal'")
MsgBox objDyn.VariableStateType
objDyn.Delete
End Sub
```

Voir aussi

Objet DynamicDialog (Page 1912)

Propriété VariableStateValues

Description

Renvoie la liste des VariableStateValues. Utilisez la propriété VariableStateValues avec la propriété Item pour affecter une valeur à un état de variable qui est utilisé pour la dynamisation.

Exemple

Dans l'exemple qui suit, le rayon d'un cercle est dynamisé dans le dialogue Dynamique. La dynamisation a lieu par analyse de l'état d'une variable. Si la variable ne renvoie aucun état,

une valeur de substitution (propriété ElseCase) est définie, un nom de variable est donné et trois plages de valeurs analogiques sont créées :

```
Sub AddDynamicDialogToCircleRadiusTypeAnalog()
'VBA786
Dim objDynDialog As HMIDynamicDialog
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_A", "HMICircle")
Set objDynDialog = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeDynamicDialog,
"'NewDynamic1'")
With objDynDialog
.ResultType = hmiResultTypeAnalog
.AnalogResultInfos.ElseCase = 200
'
'Activate analysis of variablestate
.VariableStateChecked = True
End With
With objDynDialog.VariableStateValues(1)
'
'define a value for every state:
.VALUE_ACCESS_FAULT = 20
.VALUE_ADDRESS_ERROR = 30
.VALUE_CONVERSION_ERROR = 40
.VALUE_HANDSHAKE_ERROR = 60
.VALUE_HARDWARE_ERROR = 70
.VALUE_INVALID_KEY = 80
.VALUE_MAX_LIMIT = 90
.VALUE_MAX_RANGE = 100
.VALUE_MIN_LIMIT = 110
.VALUE_MIN_RANGE = 120
.VALUE_NOT_ESTABLISHED = 130
.VALUE_SERVERDOWN = 140
.VALUE_STARTUP_VALUE = 150
.VALUE_TIMEOUT = 160
End With
End Sub
```

Voir aussi

Objet VariableStateValues (liste) (Page 2046)

Objet DynamicDialog (Page 1912)

Propriété VariableTriggers

Description

Renvoie la liste des VariableTriggers. Utilisez la propriété VariableTriggers pour ajouter un déclenchement en fonction d'une variable à une action VB ou C.

Exemple

Dans l'exemple suivant, le rayon du cercle est dynamisé avec un script VB. Le déclenchement utilisé est du type en fonction d'une variable :

```
Sub DynamicWithVariableTrigger()
'VBA787
Dim objVBScript As HMIScriptInfo
Dim objVarTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_VariableTrigger",
"HMICircle")
Set objVBScript = objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVBScript)
With objVBScript
'Triggername and cycletime are defined by add-methode
Set objVarTrigger = .Trigger.VariableTriggers.Add("VarTrigger", hmiVariableCycleType_10s)
.SourceCode = ""
End With
End Sub
```

Voir aussi

Objet VariableTriggers (liste) (Page 2049)

Propriété VarName

Description

Définit la variable dont l'état doit être utilisé dans le dialogue dynamique pour la dynamisation ou renvoie le nom.

Exemple

Dans cet exemple, le nom des variables de déclenchement qui sont utilisées pour la dynamisation du rayon du cercle sont sorties :

```
Sub GetVarName()
'VBA788
Dim objVBScript As HMIScriptInfo
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.Item("Circle_VariableTrigger")
Set objVBScript = objCircle.Radius.Dynamic
With objVBScript
'Reading out of variablename
MsgBox "The radius is dynamicabled with: " & .Trigger.VariableTriggers.Item(1).VarName
End With
End Sub
```

Voir aussi

Objet VariableStateValue (Page 2045)

Propriété VBAVersion

Description

Renvoie le numéro de version VBA. Accès en lecture.

Exemple

Dans l'exemple qui suit, le numéro de version VBA actuel est sorti :

```
Sub ShowVBAVersion()
'VBA789
MsgBox Application.VBAVersion
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété VBE

Description

Renvoie l'objet VB-Extensibility. Accès en lecture.

Exemple

--

Voir aussi

Objet Application (Page 1876)

Propriété Version

Description

Renvoie le numéro de version de l'application indiquée. Accès en lecture.

Exemple

Dans l'exemple qui suit, le numéro de version du Graphics Designer est sorti :

```
Sub ShowVersionOfGraphicsDesigner()
'VBA791
MsgBox Application.Version
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété Views

Description

Renvoie la liste des vues. Utiliser cette liste pour créer p. ex. une nouvelle copie d'une vue.

Exemple

Dans l'exemple qui suit, la vue active est copiée et la copie est ensuite activée :

```
Sub AddView()
'VBA792
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Activate
End Sub
```

Voir aussi

Objet Views (liste) (Page 2052)

Propriété Visible

Description

TRUE si l'objet indiqué doit être visible. BOOLEAN Accès en écriture et en lecture.

Exemple

Dans l'exemple qui suit, on insère un cercle dans la vue active. Ce cercle ne doit pas être visible dans Runtime:

```
Sub HideCircleInRuntime()
'VBA793
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("myCircle", "HMICircle")
objCircle.Visible = False
End Sub
```

Voir aussi

Objet ToolbarItem (Page 2031)

Objet MenuItem (Page 1967)

Objet HMIObject (Page 1943)

Objet Document (Page 1908)

Objet Toolbar (Page 2028)

Objet Menu (Page 1964)

Objet Application (Page 1876)

W - Z

Propriété WarningHigh

Description

Définit ou renvoie pour l'objet BARGRAPH le seuil supérieur "Warning High".

Pour que le seuil soit surveillé, la propriété "CheckWarningHigh" doit être définie valeur de "True".

La représentation lorsque le seuil est atteint et le type d'analyse sont définis par les propriétés ColorWarningHigh et TypeWarningHigh.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil supérieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "75".

```
Sub BarGraphLimitConfiguration()
'VBA794
```

```
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeWarningHigh = False
'Activate monitoring
.CheckWarningHigh = True
'Set barcolor = "red"
.ColorWarningHigh = RGB(255, 0, 0)
'Set upper limit = "75"
.WarningHigh = 75
End With
End Sub
```

Voir aussi

Propriété TypeWarningHigh (Page 2385)

Propriété ColorWarningHigh (Page 2135)

Propriété CheckWarningHigh (Page 2122)

Objet BarGraph (Page 1881)

Propriété WarningLow

Description

Définit ou renvoie pour l'objet BARGRAPH le seuil inférieur "Warning Low".

Pour que le seuil soit surveillé, la propriété "CheckWarningLow" doit être définie valeur de "True".

La représentation lorsque le seuil est atteint et le type d'analyse sont définis par les propriétés ColorWarningLow et TypeWarningLow.

Exemple

La procédure "BarGraphLimitConfiguration()" permet de configurer les propriétés du seuil inférieur pour une alarme. Dans cet exemple, le type d'analyse est défini sur "absolu". L'alarme doit se déclencher à la valeur "12".

```
Sub BarGraphLimitConfiguration()
'VBA795
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
'Set analysis = absolute
.TypeWarningLow = False
'Activate monitoring
.CheckWarningLow = True
```

```
'Set barcolor = "magenta"
.ColorWarningLow = RGB(255, 0, 255)
'Set lower limit = "12"
.WarningLow = 75
End With
End Sub
```

Voir aussi

Propriété TypeWarningLow (Page 2386)

Propriété ColorWarningLow (Page 2136)

Propriété CheckWarningLow (Page 2123)

Objet BarGraph (Page 1881)

Propriété Width

Description

Définit ou renvoie la largeur d'un objet en pixels.

Exemple

Dans l'exemple qui suit, on insère trois objets de taille différente dans la vue active. On sélectionne ensuite tous les objets et on les ramène tous à la même largeur :

```
Sub ApplySameWidthToSelectedObjects()
'VBA796
Dim objCircle As HMICircle
Dim objRectangle As HMIRectangle
Dim objEllipse As HMIEllipse
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("sCircle", "HMICircle")
Set objRectangle = ActiveDocument.HMIObjects.AddHMIObject("sRectangle", "HMIRectangle")
Set objEllipse = ActiveDocument.HMIObjects.AddHMIObject("sEllipse", "HMIEllipse")
With objCircle
.Top = 30
.Left = 0
.Width = 15
.Selected = True
End With
With objRectangle
.Top = 80
.Left = 42
.Width = 40
.Selected = True
End With
With objEllipse
.Top = 48
.Left = 162
```

```
.Width = 120
.BackColor = RGB(255, 0, 0)
.Selected = True
End With
MsgBox "Objects selected!"
ActiveDocument.Selection.SameWidth
End Sub
```

Voir aussi

Objet HMIObject (Page 1943)

Propriété WinCCStyle

Description

Définit dans quel style un objet est représenté.

personnalisé	Représente l'objet en fonction des paramétrages individuels.
global	Représente l'objet dans le design des paramètres globaux.
Style Windows	Représente l'objet dans le style Windows.

Exemple

Propriété WindowBorder

Description

TRUE si la fenêtre de l'application ou la fenêtre de vue doit être représentée dans Runtime avec une bordure. BOOLEAN Accès en écriture et en lecture.

Exemple

La procédure "ApplicationWindowConfig" permet d'accéder aux propriétés de la fenêtre d'application. Dans cet exemple, la fenêtre d'application est configurée:

```
Sub ApplicationWindowConfig()
'VBA797
Dim objAppWindow As HMIApplicationWindow
```

```
Set objAppWindow = ActiveDocument.HMIObjects.AddHMIObject("AppWindow",
"HMIApplicationWindow")
With objAppWindow
.Caption = True
.CloseButton = False
.Height = 200
.Left = 10
.MaximizeButton = True
.Moveable = False
.OnTop = True
.Sizeable = True
.Top = 20
.Visible = True
.Width = 250
.WindowBorder = True
End With
End Sub
```

Voir aussi

Objet PictureWindow (Page 1980)

Objet ApplicationWindow (Page 1879)

Propriété WindowMonitorNumber

Description

Définit sur quel moniteur la fenêtre de vue est affichée. La condition est que le système prenne en charge plusieurs moniteurs. Cet attribut n'agit que si l'attribut "Fenêtre indépendante" est positionné sur "oui".

1-n Le numéro du moniteur dans le système d'exploitation sur lequel s'affiche la fenêtre de vue.

Exemple

Propriété WindowPositionMode

Description

Définit la position et la mise à l'échelle de la fenêtre de vue sur l'écran. La propriété n'a d'effet que si l'attribut "Fenêtre indépendante" est réglé sur "oui".

Par défaut	La fenêtre de vue est positionnée dans sa taille d'origine sur l'endroit configuré sur l'écran.
Centrer	La fenêtre de vue est positionnée dans sa taille d'origine au centre de l'écran.
Maximiser	La fenêtre de vue est mise à l'échelle de la taille de l'écran.

Exemple

Propriété WindowsStyle

Description

Définit si l'objet doit être représenté dans le style Windows de WinCC Version 6.2. Il est uniquement sélectionnable si le design actuellement choisi est "WinCC Classic".

oui	Représente l'objet dans le style Windows de WinCC, version 6.2.
non	Ne représente pas l'objet dans le style Windows de WinCC, version 6.2.

Exemple

Propriété WindowState

Description

Renvoie l'état de la fenêtre qui contient l'application indiquée. Accès en lecture.

WindowState	Valeur affectée
Maximized	0
Minimized	1
CustomSized	2

Exemple

Dans l'exemple qui suit, l'état de la fenêtre du Graphics Designer est sorti :

```
Sub ShowWindowState()
'VBA798
Dim strState As String
Select Case Application.WindowState
Case 0
strState = "The application-window is maximized"
Case 1
strState = "The applicationwindow is minimized"
Case 2
strState = "The application-window has a userdefined size"
End Select
MsgBox strState
End Sub
```

Voir aussi

Objet Application (Page 1876)

Propriété ZeroPoint

Description

Définit ou renvoie pour l'objet BarGraph la position de l'origine du bargraphe.

Spécifiez la valeur en % par rapport à la hauteur totale du bargraphe. L'origine peut aussi se trouver en dehors de la zone représentée.

La propriété ScalingType doit être définie sur "2" et "Scaling" valeur de "True".

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple l'origine se trouve à la moitié de la hauteur du bargraphe :

```
Sub BarGraphConfiguration()
'VBA799
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Scaling = True
.ScalingType = 2
.ZeroPoint = 50
.ZeroPointValue = 0
End With
End Sub
```

Voir aussi

Propriété ZeroPointValue (Page 2477)

Propriété ScalingType (Page 2341)

Propriété Scaling (Page 2340)

Objet BarGraph (Page 1881)

Propriété ZeroPointValue

Description

Détermine la valeur absolue pour l'origine ou la restitue.

Exemple

La procédure "BarGraphConfiguration()" permet de configurer les propriétés de l'objet BarGraph. Dans cet exemple, la valeur absolue de l'origine est définie sur "0" :

```
Sub BarGraphConfiguration()
'VBA800
Dim objBarGraph As HMIBarGraph
Set objBarGraph = ActiveDocument.HMIObjects.AddHMIObject("Bar1", "HMIBarGraph")
With objBarGraph
.Scaling = True
.ScalingType = 2
.ZeroPointValue = 0
End With
End Sub
```

Voir aussi

Propriété ZeroPoint (Page 2476)

Propriété ScalingType (Page 2341)

Propriété Scaling (Page 2340)

Objet BarGraph (Page 1881)

Objet 3DBarGraph (Page 1867)

Propriété Zoom

Description

Définit ou renvoie le facteur de zoom.

Exemple

Dans cet exemple, une copie de la vue active est générée et le facteur de zoom est défini sur 50 % :

```
Sub CreateViewFromActiveDocument()
'VBA801
Dim objView As HMIView
Set objView = ActiveDocument.Views.Add
objView.Zoom = 50
End Sub
```

Voir aussi

Objet View (Page 2050)

Objet PictureWindow (Page 1980)

3.5.2 VBA dans d'autres éditeurs WinCC

3.5.2.1 VBA dans d'autres éditeurs WinCC

Introduction

VBA vous offre la possibilité d'accéder à d'autres éditeur WinCC comme par exemple Tag Logging. Les éditeurs suivants peuvent être automatisés en plus de Graphics Designer avec VBA :

- Stock de variables
- Tag Logging
- Text Library
- Alarm Logging

Les fonctions d'accès aux éditeurs sont compris dans la classe HMIGO.

Condition

Le fichier "HMIGenObjects.dll" est référencé. Ceci est effectué automatiquement lors de l'installation de WinCC.

Principe

Pour que vous puissiez avoir accès à la classe HMIGO, vous devez référencer la "HMI GeneralObjects 1.0 Type Library" dans l'éditeur VBA ("Project" > "References"). Vous devez générer une nouvelle instance de cette classe dans le code, p. ex. :

```
'Dim HMIGOObject As New HMIGO
```

Générez plusieurs objets différents de cette classe quand vous accédez à plusieurs objets au même moment. Deux instances de la classe HMIGO sont par exemple requises dans Tag Logging: Vous avez besoin de la première instance pour accéder aux variables d'archive et de la deuxième pour accéder aux archives mêmes des valeurs processus.

Utilisation

Les fonctions disponibles avec la classe HMIGO pour permettent d'avoir accès à la gestion des variables, Tag Logging, Text Library et Alarm Logging. Pour pouvoir utiliser les fonction dans VBA, vous devez avoir un projet ouvert dans WinCC. Vous avez en outre la possibilité d'accéder directement aux propriétés de la classe.

Vous pouvez ainsi, directement à partir du code, p. ex. générer plusieurs variables et changer leurs valeurs, éditer des entrées de texte dans TextLibrary ou adapter des alarmes.

Interroger l'état de l'objet

La classe HMIGO dispose de l'énumération HMIGO_OBJECT_STATE qui renvoie l'état de l'objet spécifié. L'énumération peut renvoyer les valeurs suivantes:

- OBJECT_EMPTY (2): Aucune liaison à l'objet.
- OBJECT_OPENED (3): Il existe une liaison à l'objet. Vous pouvez modifier ou lire ses paramètres.
- OBJECT_MODIFIED (4): Les paramètres d'un objet ont été modifiés. Les modifications sont perdues si la fonction Commit correspondante n'est pas sélectionnée.
- WINCC_CONNECTED (1): Le projet est relié au projet WinCC. Cette liaison s'établit de façon standard la première fois que cette fonction est sélectionnée. Utilisez par exemple l'instruction HMIGO = nothing pour annuler cette liaison.

Traitement des erreurs

Lors de l'utilisation de la classe HMIGO des erreurs peuvent apparaître. Utilisez l'instruction OnError pour réagir à ces messages d'erreur. l'instruction OnError doit se trouver avant l'appel d'une fonction hors de la classe HMIGeneralObjects :

```
Sub CreateTag()
'HMIGO_000
Dim hmiGOTag as New HMIGO
On Error GoTo ErrorHandlerHMIGO
hmiGOTag.CreateTag "NewTag", TAG_BINARY_TAG, "ExistingConnection", "DB1,DD0,QC",
"NewOrExistingGroupName"

'...
    Exit Sub
ErrorHandlerHMIGO:
MsgBox ("Error: " & Err.Number & " " & Err.Description & " " & Err.Source)
Resume Next
End Sub
```

Ceci entraîne le renvoi d'un texte d'erreur de l'interface.

Voir aussi

VBA dans Alarm Logging (Page 2532)

VBA dans Text Library (Page 2519)

VBA dans Tag Logging (Page 2492)

VBA dans la gestion des variables (Page 2481)

3.5.2.2 VBA dans la gestion des variables

VBA dans la gestion des variables

Introduction

VBA vous permet de générer, modifier ou supprimer des variables ainsi que de lire et modifier leurs propriétés, type et valeurs directement à partir du code.

Principe

Si vous avez généré l'instance de la classe HMIGO, les fonctions suivantes sont disponibles pour l'accès à la gestion des variables :

- CloseTag
- CommitTag
- CreateTag
- DeleteTag
- GetTag
- ListTag

Les énumérations "HMIGO_TAG_TYPE" et "HMIGO_TAG_LIST_TYPE" vous permettent d'alimenter les paramètres de ces fonctions.

Remarque

Au cours de l'édition avec VBA, la variable ne doit pas être ouverte dans le stock de variables. Si vous voulez modifier le type de données d'une variable, il vous faut d'abord supprimer puis régénérer cette variable. Vous devez auparavant sauvegarder les paramètres pour transmettre ces derniers après génération de la variable.

Remarque

Si vous définissez la valeur initiale d'une variable binaire, utilisez les valeurs "0" ou "1". N'utilisez pas les valeurs "False" ou "True". Ces valeurs ne sont pas valables pour la programmation VBA dans WinCC et provoquent un message d'erreur.

Remplacez dans les codes VBA existants les valeurs "False" et "True" par "0" et "1".

Accès aux propriétés de l'objet

Vous pouvez aussi accéder directement dans VBA aux paramètres des fonctions citées ci-dessus par les propriétés de l'objet :

Propriété de l'objet	Description	Lecture/écriture
ObjectStateTag	Renvoie l'état de l'objet par le biais de l'énumération HMIGO_OBJECT_STATE. Vous trouverez sous "VBA dans d'autres éditeurs WinCC".	oui/non
Nom de variable	Nom des variables	oui/non
TagGroupName	Nom d'un groupe dans lequel la variable est insérée. Si le groupe est encore inexistant, celui-ci est créé. Si aucun nom de groupe est indiqué, la variable est créé de dehors de tous les groupes.	oui/non
TagConnection	Nom d'une liaison sous laquelle la variable ou / et le groupe doit être créé(e)(s). Il doit s'agir d'une liaison déjà existante, sinon aucune variable ne peut être créée. Une variable interne est crée au cas où le nom serait omis.	oui/non
TagMaximum	Détermine la valeur du seuil supérieur	oui/oui
TagMinimum	Détermine la valeur du seuil inférieur	oui/oui
TagStart	Détermine la nouvelle valeur de départ.	oui/oui
TagS5S7Address	Adresse de l'automate S7 ou S5 auquel le connecteur est relié. Si l'adresse est omise, une entrée vide est transmise.	oui/oui
Type de variable (Enum)	Type de données de la variable. Les types possibles sont les suivants : • TAG_BINARY_TAG (1) • TAG_SIGNED_8BIT_VALUE (2) • TAG_UNSIGNED_8BIT_VALUE (3) • TAG_SIGNED_16BIT_VALUE (4) • TAG_UNSIGNED_16BIT_VALUE (5) • TAG_SIGNED_32BIT_VALUE (6) • TAG_UNSIGNED_32BIT_VALUE (7) • TAG_FLOATINGPOINT_NUMBER_32BIT_IEEE_754 (8) • TAG_FLOATINGPOINT_NUMBER_64BIT_IEEE_754 (9) • TAG_TEXT_TAG_8BIT_CHARACTER_SET (10) • TAG_TEXT_TAG_16BIT_CHARACTER_SET (11) • TAG_RAW_DATA_TYPE (12) • TAG_STRUCT (14) • TAG_TEXT_REFERENCE (18)	oui/non
TagUpdate (Enum)	Définit si la variable est uniquement mise à jour au niveau de l'ordinateur ou également dans l'ensemble du projet. (pour variable interne uniquement) • TAG_COMPUTER_LOCAL (1) • TAG_PROJECT_WIDE (2)	oui/oui

Propriété de l'objet	Description	Lecture/écriture
LengthText	Longueur d'une variable de texte (0...255) "LengthText" est également utilisable pour la longueur des variables de données brutes. La longueur n'est pas contrôlée. Prenez en considération les paramètres par défaut des canaux de communication.	ja/ja (seulement pour les variables externes de type Text)
TagScaleValid	Définit une mise à l'échelle linéaire.	oui/oui
TagScaleParam1	Définit la valeur1 pour zone de valeur process.	oui/oui
TagScaleParam2	Définit la valeur2 pour zone de valeur process.	oui/oui
TagScaleParam3	Définit la valeur1 pour zone de valeur variable.	oui/oui
TagScaleParam4	Définit la valeur2 pour zone de valeur variable.	oui/oui
TagStartvaluePersistence	Définit si une variable interne est fixée de manière persistante.	oui/oui
TagSubst	Valeur de remplacement (seulement en cas de variables externes)	oui/oui
UseSubstValueOnCommonError	Définit la valeur de remplacement en cas d'erreurs de liaison.	oui/oui
UseSubstValueOnMaxLimit	Définit la valeur de remplacement de la limite supérieure.	oui/oui
UseSubstValueOnMinLimit	Définit la valeur de remplacement de la limite inférieure.	oui/oui
UseSubstValueOnStartValue	Définit la valeur de remplacement de la valeur initiale.	oui/oui

Vous trouverez une description des propriétés dans la section Description des paramètres des fonctions correspondantes.

Remarque

L'option "Synchronisation de variables" située dans le dialogue Propriétés de variable ne peut pas être activée avec VBA. La synchronisation de variables n'est disponible que pour les variables internes.

Pour les variables externes, l'option "Transtypage" ne peut pas être activée avec VBA.

Voir aussi

Fonction ListTag (Page 2490)

Fonction GetTag (Page 2489)

Fonction DeleteTag (Page 2488)

Fonction CreateTag (Page 2486)

Fonction CommitTag (Page 2485)

Fonction CloseTag (Page 2484)

VBA dans d'autres éditeurs WinCC (Page 2479)

Fonction CloseTag

Description

Ferme la variable ouverte.

Remarque

Les paramètres modifiés ne sont pas enregistrés.

Syntaxe

```
Expression CloseTag()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CloseTag()
' HMIGO_001
' procedure to close a variable
' tag need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strVariableName As String
  Set objHMIGO = New HMIGO
  strVariableName = "NewVariable"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
 'open a tag
  objHMIGO.GetTag strVariableName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
 'close a tag
  objHMIGO.CloseTag
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"

  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTag (Page 2490)
Fonction GetTag (Page 2489)
Fonction DeleteTag (Page 2488)
Fonction CreateTag (Page 2486)
Fonction CommitTag (Page 2485)
VBA dans la gestion des variables (Page 2481)

Fonction CommitTag

Description

Ecrit les paramètres modifiés de la Variable ouverte dans WinCC.

Remarque

Si vous modifiez d'autres paramètres après avoir appelé CommitTag, vous devez écrire ces modifications dans WinCC avec un nouvel appel de cette fonction.

Syntaxe

```
Expression CommitTag()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CommitTag()
' HMIGO_002
' procedure to change a property of a variable
' tag need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strVariableName As String
  Set objHMIGO = New HMIGO
  strVariableName = "NewVariable"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
 'open a tag
```

```
  objHMIGO.GetTag strVariableName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
 'change a property
  objHMIGO.TagStart = 10
 'current status is "MODIFIED"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
 'commit a tag
  objHMIGO.CommitTag
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"

  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTag (Page 2490)

Fonction GetTag (Page 2489)

Fonction DeleteTag (Page 2488)

Fonction CreateTag (Page 2486)

Fonction CloseTag (Page 2484)

VBA dans la gestion des variables (Page 2481)

Fonction CreateTag

Description

Génère une nouvelle variable.

Syntaxe

```
Expression.CreateTag(TagName,TagType,[Connection],[S7S5Address],
[GroupName])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
TagName (chaîne de caractères)	Nom de la variable à générer.
TagType (HMIGO_TAG_TYPE)	Type de données de la variable. Les types possibles sont les suivants : TAG_BINARY_TAG TAG_SIGNED_8BIT_VALUE TAG_UNSIGNED_8BIT_VALUE TAG_SIGNED_16BIT_VALUE TAG_UNSIGNED_16BIT_VALUE TAG_SIGNED_32BIT_VALUE TAG_UNSIGNED_32BIT_VALUE TAG_FLOATINGPOINT_NUMBER_32BIT_IEEE_754 TAG_FLOATINGPOINT_NUMBER_64BIT_IEEE_754 TAG_TEXT_TAG_8BIT_CHARACTER_SET TAG_TEXT_TAG_16BIT_CHARACTER_SET TAG_RAW_DATA_TYPE TAG_TEXT_REFERENCE
Connection (chaîne de caractères, optionnel)	Nom d'une liaison sous laquelle la variable ou / et le groupe doit être créé(e)(s). Il doit s'agir d'une connexion existant déjà, faute de quoi il est impossible de créer des variables. Si le nom est omis, une variable et/ou groupe interne sera créé(e).
S7S5Address (chaîne de caractères, optionnel)	Adresse de l'automate S7 ou S5 auquel le connecteur est relié. Si vous ne spécifiez pas d'adresse, un paramètre vide est transmis. Pour la configuration du code qualité, ajoutez au paramètre "S7S5Address" la chaîne de caractères ",QC", comme par ex. : "DB1,DD0,QC". S'il n'est plus nécessaire de contrôler le code qualité de la variable, effacez la chaîne de caractères ",QC".
GroupName (chaîne de caractères, optionnel)	Nom d'un groupe dans lequel la variable est insérée. Si le groupe n'existe pas encore, il est créé. Si le nom du groupe n'est pas spécifié, la variable est créée hors de tous les groupes.

Exemple

```
Sub CreateTag()
' HMIGO_003
' procedure to create a variable
' tag must not be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strVariableName As String
```

```
  Set objHMIGO = New HMIGO
  strVariableName = "NewVariable"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
 'create a tag
  objHMIGO.CreateTag strVariableName, TAG_SIGNED_32BIT_VALUE
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTag (Page 2490)

Fonction GetTag (Page 2489)

Fonction DeleteTag (Page 2488)

Fonction CommitTag (Page 2485)

Fonction CloseTag (Page 2484)

VBA dans la gestion des variables (Page 2481)

Fonction DeleteTag

Description

Supprime la variable indiquée.

Syntaxe

`Expression.DeleteTag(TagName)`

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
TagName (chaîne de caractères)	Nom de la variable à supprimer.

Exemple

```
Sub DeleteTag()
' HMIGO_004
```

```
' procedure to delete a variable
' tag need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strVariableName As String
  Set objHMIGO = New HMIGO
  strVariableName = "NewVariable"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"

 'delete a tag
  objHMIGO.DeleteTag strVariableName
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTag (Page 2490)

Fonction GetTag (Page 2489)

Fonction CreateTag (Page 2486)

Fonction CommitTag (Page 2485)

Fonction CloseTag (Page 2484)

VBA dans la gestion des variables (Page 2481)

Fonction GetTag

Description

Lit les paramètres de la Variable indiquée.

Vous pouvez modifier ou lire les paramètres via les propriétés de l'objet. Vous trouverez une liste des propriétés d'objets disponibles dans cette documentation à la rubrique "VBA dans la gestion des variables".

Syntaxe

```
Expression.GetTag(TagName)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
TagName (chaîne de caractères)	Nom de la variable dont les valeurs doivent être lues.

Exemple

```
Sub GetTag()
' HMIGO_005
' procedure to open a variable
' tag need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strVariableName As String
  Set objHMIGO = New HMIGO
  strVariableName = "NewVariable"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
 'open/ get a tag
  objHMIGO.GetTag strVariableName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTag, vbOKOnly, "Status Variable"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTag (Page 2490)

Fonction DeleteTag (Page 2488)

Fonction CreateTag (Page 2486)

Fonction CommitTag (Page 2485)

Fonction CloseTag (Page 2484)

VBA dans la gestion des variables (Page 2481)

Fonction ListTag

Description

En alternative, la fonction ListTag renvoie sous forme de liste le contenu suivant du stock de variables:

- toutes les unités de canal créées
- tous les canaux créés
- toutes les liaisons établies

- tous les groupes de variables créés
- toutes les variables créées

Syntaxe

```
Expression.ListTag(ListType,pListArray,[Filter])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ListType (HMIGO_TAG_LIST_TYPE)	Détermine quel sera le contenu renvoyé sous forme de liste. Solutions possibles: • TAG_CHANNEL_UNITS (0) toutes les unités de canal créées • TAG_CHANNELS (2) tous les canaux créés • TAG_CONNECTIONS (3) toutes les liaisons établies • TAG_GROUPS (4) tous les groupes de variables créés • TAG_NAMES (5) toutes les variables créées
pListArray (Variant)	Liste du contenu exigé.
Filtre (String)	Des filtres peuvent également être placés en option. Les Wildcards "*" et "?" sont également acceptés.

Exemple

Etant donné qu'aucune liaison n'est créée, il est vérifié dans l'exemple suivant si la liste des liaisons établies est vide:

```
Sub ReadTagByConnection()
'HMIGO_027
'read content in data manager by connections
'no conections are implemented
  Dim objHMIGO As New HMIGO
  Dim arrContent As Variant
'read all connections
  objHMIGO.ListTag TAG_CONNECTIONS, arrContent
'check result
  If (UBound(arrContent) - LBound(arrContent) + 1) <= 0 Then
    MsgBox "no entries because no connections are implemented"
    End If
End Sub
```

Voir aussi

Fonction GetTag (Page 2489)
Fonction DeleteTag (Page 2488)
Fonction CreateTag (Page 2486)
Fonction CommitTag (Page 2485)
Fonction CloseTag (Page 2484)
VBA dans la gestion des variables (Page 2481)

3.5.2.3 VBA dans Tag Logging

VBA dans Tag Logging

Introduction

VBA vous permet de générer, modifier ou supprimer des archives de valeurs de process et des variables d'archive directement à partir du code.

Principe

Si vous avez généré l'instance de la classe HMIGO, les fonctions suivantes sont disponibles pour l'accès à TagLogging :

- CloseTlgArchive
- CloseTlgTag
- CommitTlgArchive
- CommitTlgTag
- CreateTlgArchive
- CreateTlgTag
- DeleteTlgArchive
- DeleteTlgTag
- GetTlgArchive
- GetTlgTag
- ListTlgArchive
- ListTlgTag

Remarque

Au cours de l'édition avec VBA, Tag Logging ne doit pas être ouvert.
Avant d'utiliser le déclenchement du Tag Logging dans VBA pour un projet WinCC venant d'être créé, il vous faut d'abord démarrer l'éditeur "Tag Logging", créer une archive puis refermer l'éditeur. Le déclenchement est créé par l'éditeur.

Accès direct aux propriétés de l'objet

Vous pouvez aussi accéder directement dans VBA aux paramètres des fonctions citées ci-dessus par les propriétés de l'objet : Dans la colonne "est utilisé dans" vous voyez si vous pouvez accéder à la propriété de l'objet dans l'archive de valeurs de process (P) et/ou dans l'archive agrégée (V).

Propriété de l'objet	Description	Lecture/ écriture	est utilisé dans
ObjectStateTlgArchive	Renvoie l'état de l'objet par le biais de l'énumération HMIGO_OBJECT_STATE. Vous trouverez sous "VBA dans d'autres éditeurs WinCC".	oui/non	P, V
TlgArchiveName	Nom des archives de la valeur du process ou condensées.	oui/non	P, V
TlgArchiveType	Détermine s'il s'agit d'une archive de valeurs de process ou d'une archive condensée.	oui/non	P, V
TlgArchiveAccessLevel Read	Le niveau d'autorisation lecture.	oui/non	P, V
TlgArchiveAccessLevel Write	Le niveau d'autorisation écriture.	oui/non	P, V
TlgArchiveArchiveState	Définit si l'archivage doit être bloqué ou activé lors du lancement du système. Valeurs possibles pour Enum HMIGO_TLG_ARCHIVE_STATE: • TLG_ARCHIVE_STATE_LOCKED (1) • TLG_ARCHIVE_STATE_ACTIVATED (0)	oui/oui	P, V
TlgArchiveBufferSize	Définit le nombre d'enregistrements pour une archive cyclique.	oui/oui	P
TlgArchiveBufferType	Détermine le lieu de l'enregistrement des variables. Types possibles pour Enum HMIGO_TLG_ARCHIVE_BUFFER_TYPE: • TLG_ARCHIVE_BUFFER_TYPE_DISK (2) • TLG_ARCHIVE_BUFFER_TYPE_RAM (1)	oui/oui	P
TlgArchiveCompressRa nge	Détermine la période d'agrégation des variables. Il s'agit du nom du temporisateur définit sous "Temporisations" dans l'éditeur TagLogging et qui est supérieur ou égal à 1 minute. Etant donné que le format est une chaîne de caractères, il dépend de la langue. Egalement définissable par la fonction ListTlgArchive(TLG_ARCHIVE_TRIGGER_ NAMES, arrTrigger)	oui/oui	V

Propriété de l'objet	Description	Lecture/ écriture	est utilisé dans
TlgArchiveCompressType	Détermine avec quel algorithme les valeurs sont agrégées. Types possibles pour Enum HMIGO_TLG_ARCHIVE_COMPRESS_TYPE: • TLG_COMPRESS_TYPE_CALC (1) • TLG_COMPRESS_TYPE_CALC_COPY (2) • TLG_COMPRESS_TYPE_CALC_DEL (3) • TLG_COMPRESS_TYPE_CALC_COPY_DEL (4)	oui/oui	V
ObjectStateTlgTag	Renvoie l'état de l'objet par le biais de l'énumération HMIGO_OBJECT_STATE. Vous trouverez sous "VBA dans d'autres éditeurs WinCC".	oui/non	P, V
TlgTagArchiveName	Nom des variables d'archive.	oui/non	P, V
TlgTagName	Nom de l'archive.	oui/non	P, V
TlgTagType	Indique le type de variable. Types possibles pour Enum HMIGO_TLG_TAG_TYPE: • TLG_TAG_TYP_ANALOG (65537) • TLG_TAG_TYP_BINARY (65538) • TLG_TAG_TYP_PROCESS (65544) • TLG_TAG_TYP_COMPRESS (65540)	oui/oui	P, V
TlgTagArchiving	Indique le mode d'acquisition. Valeurs possibles pour Enum HMIGO_TLG_TAG_ARCHIVING: • TLG_TAG_ACYCLIC (8388609) • TLG_TAG_CYCLIC_CONTINUOUS (8388610) • TLG_TAG_CYCLIC_SELECTIVE (8388612) • TLG_TAG_ON_EVERY_CHANGE (8388616)	oui/oui	P
TlgTagArchivingState	Détermine si l'archivage être bloqué ou activé lors du lancement du système. Valeurs possibles pour Enum HMIGO_TLG_TAG_ARCHIVING_STATE: • TLG_TAG_LOCKED (1) • TLG_TAG_ACTIVATED (0)	oui/oui	P, V
TlgTagOnError	Détermine si c'est la dernière valeur saisie ou la valeur de substitution qui doit être enregistrée en cas de défaut. Valeurs possibles pour Enum HMIGO_TLG_TAG_ON_ERROR: • TLG_TAG_LAST_VALUE (1) • TLG_TAG_SUBSTITUTE (2)	oui/oui	P

Propriété de l'objet	Description	Lecture/ écriture	est utilisé dans
TlgTagTriggerType	Détermine comment l'archivage doit se comporter lors d'un changement d'état. Valeurs possibles pour Enum HMIGO_TLG_TAG_TRIGGER_TYPE: • TLG_TAG_FROM_0_TO_1 (2) • TLG_TAG_FROM_1_TO_0 (3) • TLG_TAG_ALWAYS (4) • TLG_TAG_EVERY_CHANGE (1)	oui/oui	P
TlgTagOnChange	Détermine si l'archivage doit avoir lieu si une modification apparaît. Valeurs possibles pour Enum HMIGO_TLG_TAG_ON_CHANGE: • TLG_TAG_EVERY_VALUE (0) • TLG_TAG_RELATIVE_HYSTERESE (1) • TLG_TAG_ABSOLUTE_HYSTERESE (2)	oui/oui	P
TlgTagMethodType	Détermine la méthode à employer pour le traitement de la valeur avant l'archivage. Valeurs possibles pour Enum HMIGO_TLG_TAG_METHOD_TYPE: • TLG_TAG_ACTUAL (1) • TLG_TAG_SUM (3) • TLG_TAG_MaxValue (5) • TLG_TAG_MinValue (4) • TLG_TAG_AVERAGE (2)	oui/oui	P, V
TlgTagTriggerScan	Nom du temporisateur pour le cycle d'acquisition.	oui/oui	P
TlgTagTriggerArchiving	Nom du temporisateur pour le cycle d'archivage.	oui/oui	P
TlgTagStartTriggerFunction	Indique le nom d'une fonction de script avec laquelle un événement déclencheur doit être vérifié pour le démarrage de l'archivage.	oui/oui	P
TlgTagStartTriggerModule	Indique le nom d'une DLL depuis laquelle la fonction script pour la vérification d'un événement déclencheur doit être appelée.	oui/oui	P
TlgTagStopTriggerFunction	Indique le nom d'une fonction de script avec laquelle un événement d'arrêt doit être vérifié pour le démarrage de l'archivage.	oui/oui	P
TlgTagTriggerFunction	Indique le nom d'une fonction de script pour le basculement dynamique cycle d'acquisition/d'archivage.	oui/oui	P
TlgTagNameCompressArchive	Contient dans le cas d'archive agrégée le nom de l'archive source.	oui/oui	V
TlgTagNameCompressTag	Contient dans le cas d'archives agrégées le nom de la variable source.	oui/oui	V
TlgTagNameRawValue	Contient dans le cas d'archives pilotées par process le nom de la variable de données brutes.	oui/oui	P

Propriété de l'objet	Description	Lecture/ écriture	est utilisé dans
TlgTagConvertModule	Nom de la DLL de normalisation utilisée pour la conversion des données.	oui/oui	P
TlgTagNameProcTag	Nom des variables de process fournissant la valeur d'acquisition.	oui/oui	P
TlgTagAliasName	Le nom alternatif sous lequel la variable est accessible (alias).	oui/oui	P
TlgTagStartEvent	Nom de la variable par laquelle le démarrage de l'archivage est vérifié.	oui/oui	P
TlgTagStopEvent	Nom de la variable par laquelle l'arrêt de l'archivage est vérifié.	oui/oui	P
TlgTagTriggerFactor	Contient le facteur pour le cycle d'affichage comme multiple du cycle d'archivage.	oui/oui	P
TlgTagUpperLimit	Valeur pour la mise à l'échelle du seuil supérieur de la variable.	oui/oui	P
TlgTagLowerLimit	Valeur pour la mise à l'échelle du seuil inférieur de la variable.	oui/oui	P
TlgTagHysterese	Valeur pour l'hystérésis par laquelle la modification d'une valeur est vérifiée.	oui/oui	P

Voir aussi

Fonction ListTlgTag (Page 2518)

Fonction ListTlgArchive (Page 2516)

Fonction GetTlgArchive (Page 2513)

Fonction DeleteTlgTag (Page 2511)

Fonction DeleteTlgArchive (Page 2510)

Fonction CreateTlgTag (Page 2505)

Fonction CreateTlgArchive (Page 2503)

CommitTlgTag-Funktion (Page 2501)

CommitTlgArchive-Funktion (Page 2499)

Fonction CloseTlgTag (Page 2498)

Fonction CloseTlgArchive (Page 2497)

VBA dans d'autres éditeurs WinCC (Page 2479)

Fonction CloseTlgArchive

Description

Ferme l'archive de valeurs de process ouverte ou l'archive agrégée ouverte.

Remarque

Les paramètres modifiés ne sont pas enregistrés.

Syntaxe

```
Expression.CloseTlgArchive()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CloseTlgArchive()
' HMIGO_006
' procedure to close an archive
' the archive need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
 'open archive
  objHMIGO.GetTlgArchive strArchiveName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
 'close archive
  objHMIGO.CloseTlgArchive
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"

  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgTag (Page 2511)
Fonction DeleteTlgArchive (Page 2510)
Fonction CreateTlgTag (Page 2505)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
CommitTlgArchive-Funktion (Page 2499)
Fonction CloseTlgTag (Page 2498)
VBA dans Tag Logging (Page 2492)

Fonction CloseTlgTag

Description

Ferme la variable d'archive ouverte.

Remarque

Les paramètres modifiés ne sont pas enregistrés.

Syntaxe

```
Expression.CloseTlgTag()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CloseTlgTag()
' HMIGO_007
' procedure to close a tag logging tag
' the archive need to be created before
' the tag logging tag need to be created before
```

```
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Dim strTlgTagName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
  strTlgTagName = "NewTag"

 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Tag"
 'open/ get tag logging tag
  objHMIGO.GetTlgTag strArchiveName, strTlgTagName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Archive"
 'close tag logging tag
  objHMIGO.CloseTlgTag
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Archive"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTlgTag (Page 2518)

Fonction ListTlgArchive (Page 2516)

Fonction GetTlgArchive (Page 2513)

Fonction DeleteTlgTag (Page 2511)

Fonction DeleteTlgArchive (Page 2510)

Fonction CreateTlgTag (Page 2505)

Fonction CreateTlgArchive (Page 2503)

CommitTlgTag-Funktion (Page 2501)

CommitTlgArchive-Funktion (Page 2499)

Fonction CloseTlgArchive (Page 2497)

VBA dans Tag Logging (Page 2492)

CommitTlgArchive-Funktion

Description

Ecrit les paramètres modifiés de l'archive indiquée vers WinCC.

Remarque

Si vous modifiez d'autres paramètres après avoir appelé CommitTlgArchive, vous devez écrire ces modifications dans WinCC avec un nouvel appel de cette fonction.

Syntaxe

```
Expression.CommitTlgArchive()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CommitTlgArchive()
' HMIGO_008
' procedure to change a property of an archive
' the archive need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
 'open archive
  objHMIGO.GetTlgArchive strArchiveName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
 'change a property
  objHMIGO.TlgArchiveArchiveState = TLG_STATE_LOCKED
 'current status is "MODIFIED"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
 'commit archive
  objHMIGO.CommitTlgArchive
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"

  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction DeleteTlgArchive (Page 2510)
Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgTag (Page 2511)
Fonction CreateTlgTag (Page 2505)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
Fonction CloseTlgTag (Page 2498)
Fonction CloseTlgArchive (Page 2497)
VBA dans Tag Logging (Page 2492)

CommitTlgTag-Funktion

Description

Ecrit les paramètres modifiés de la variable d'archive indiquée vers WinCC.

Remarque

Si vous modifiez d'autres paramètres après avoir appelé CommitTlgTag, vous devez écrire ces modifications dans WinCC avec un nouvel appel de cette fonction.

Syntaxe

```
Expression.CommitTlgTag()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CommitTlgTag()
' HMIGO_009
' procedure to change a property of a tag logging tag
```

```
 ' the archive need to be created before
 ' the tag logging tag need to be created before
 ' declarations
   Dim objHMIGO As HMIGO
   Dim strArchiveName As String
   Dim strTlgTagName As String
   Set objHMIGO = New HMIGO
   strArchiveName = "NewArchive"
   strTlgTagName = "NewTag"

  'current status is "EMPTY"
   MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Tag"
  'open/ get tag logging tag
   objHMIGO.GetTlgTag strArchiveName, strTlgTagName
  'current status is "OPENED"
   MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Archive"
  'change a property
   objHMIGO.TlgTagArchiving = TLG_TAG_ON_EVERY_CHANGE
  'current status is "MODIFIED"
   MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Archive"
  'commit tag logging tag
   objHMIGO.CommitTlgTag
  'current status is "OPENED"
   MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Archive"
   Set objHMIGO = Nothing
 End Sub
```

Voir aussi

Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgTag (Page 2511)
Fonction DeleteTlgArchive (Page 2510)
Fonction CreateTlgTag (Page 2505)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
CommitTlgArchive-Funktion (Page 2499)
Fonction CloseTlgTag (Page 2498)
Fonction CloseTlgArchive (Page 2497)
VBA dans Tag Logging (Page 2492)

Fonction CreateTlgArchive

Description

Crée une nouvelle archive de valeur du process ou agrégée.

Syntaxe

```
Expression.CreateTlgArchive(ArchiveName,ArchiveType)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ArchiveName (chaîne de caractères)	Nom de l'archive à créer.
ArchiveType (HMIGO_TLG_ARCHIVE_TYPE)	Type de l'archive. Les types possibles sont les suivants : • TLG_PROCESSARCHIVE (131073) pour une archive de valeurs de processus • TLG_COMPRESSARCHIVE (131074) pour une archive agrégée

Valeurs par défaut pour la création d'une archive de variable

Le tableau suivant montre les valeurs par défaut entrées lors de la création d'une archive de valeurs du process ou agrégée. Ces valeurs peuvent être modifiées ultérieurement et écrites avec la fonction CommitTlgArchive.

Propriété	Valeur par défaut (nom de l'Enum => valeur)	Remarque
TlgArchiveAccessLevelRead	0	Sans niveau d'autorisation
TlgArchiveAccessLevelWrite	0	Sans niveau d'autorisation
TlgArchiveArchiveState	TLG_ARCHIVE_STATE_ACTIVATED (0)	L'archivage est démarré au démarrage de Runtime.
TlgArchiveBufferSize	1000	Nombre d'enregistrements
TlgArchiveBufferType	TLG_ARCHIVE_BUFFER_TYPE_DISK (2)	Les valeurs sont enregistrées dans la banque de données sur le disque dur.

Propriété	Valeur par défaut (nom de l'Enum => valeur)	Remarque
TlgArchiveCompressRange	"1 jour". Cette chaîne de caractère doit être créée individuellement pour chaque langue (l'anglais par exemple: "1 jour")	Correspond à un jour exactement. Significatif seulement pour les variables d'agrégation. Particularité : l'utilisateur est responsable pour une valeur >= 1 minute
TlgArchiveCompressType	TLG_COMPRESS_TYPE_CALC (1)	Seulement calculer les valeurs d'agrégation. Pertinent seulement pour les variables d'agrégation.

Enum HMIGO_TLG_ARCHIVE_STATE

Paramètres	Description
TLG_ARCHIVE_STATE_LOCKED (1)	L'archivage est bloqué au démarrage du système.
TLG_ARCHIVE_STATE_ACTIVATED (0)	L'archivage est démarré au démarrage de Runtime.

Enum HMIGO_TLG_ARCHIVE_BUFFER_TYPE

Paramètres	Description
TLG_ARCHIVE_BUFFER_TYPE_DISK (2)	L'archivage est effectué sur le disque dur.
TLG_ARCHIVE_BUFFER_TYPE_RAM (1)	L'archivage des valeurs n'est effectué que dans la mémoire vive.

Enum HMIGO_TLG_ARCHIVE_COMPRESS_TYPE

Paramètres	Description
TLG_COMPRESS_TYPE_CALC (1)	Seulement les valeurs d'agrégation sont calculées.
TLG_COMPRESS_TYPE_CALC_COPY (2)	Les valeurs d'agrégation sont calculées et les valeurs d'origine sont copiées.
TLG_COMPRESS_TYPE_CALC_DEL (3)	Les valeurs d'agrégation sont calculées et les valeurs d'origine sont ensuite supprimées.
TLG_COMPRESS_TYPE_CALC_COPY_DEL (4)	Les valeurs d'agrégation sont calculées et les valeurs d'origine sont copiées et ensuite supprimées.

Exemple

```
Sub CreateTlgArchive()
' HMIGO_010
' procedure to create an archive
' the archive must not be created before
```

```
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
 'create tag logging archive
  objHMIGO.CreateTlgArchive strArchiveName, TLG_PROCESSARCHIVE
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction GetTlgArchive (Page 2513)
Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction DeleteTlgTag (Page 2511)
Fonction DeleteTlgArchive (Page 2510)
Fonction CreateTlgTag (Page 2505)
CommitTlgTag-Funktion (Page 2501)
CommitTlgArchive-Funktion (Page 2499)
Fonction CloseTlgTag (Page 2498)
Fonction CloseTlgArchive (Page 2497)
VBA dans Tag Logging (Page 2492)

Fonction CreateTlgTag

Description

Crée une nouvelle variable d'archive.

Syntaxe

```
Expression.CreateTlgTag(ArchiveName,TagName,[TagType])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ArchiveName (chaîne de caractères)	Nom d'une archive existante dans laquelle la variable doit être inscrite.
TagName (chaîne de caractères)	Nom de la variable à créer.
TagType (HMIGO_TLG_TAG_TYPE, optionnel)	Détermine le type de variable. Si le type n'est pas indiqué, la valeur par défaut TLG_VAR_TYP_ANALOG est entrée. Les types possibles sont les suivants : • TLG_VAR_TYP_ANALOG (65537) pour une variable analogique. • TLG_VAT_TYP_BINARY (65538) pour une variable binaire. • TLG_VAR_TYP_PROCESS (65544) pour une variable de processus. • TLG_VAT_TYP_COMPRESS (65540) pour une variable d'agrégation.

Valeurs par défaut pour la création d'une variable d'archive

Le tableau suivants montre les valeurs par défaut entrées lors de la création d'une variable d'archive. Ces valeurs peuvent être modifiées ultérieurement et écrites avec la fonction CommitTlgTag.

Propriété	Valeur par défaut (nom de l'Enum => valeur)	Remarques
TlgTagType	TLG_VAR_TYP_ANALOG (65537)	Saisie via une variable analogique du gestionnaire de données.
TlgTagArchiving	TLG_TAG_CYCLIC_CONTINUOUS (8388610)	Saisie cyclique continue.
TlgTagArchivingState	TLG_TAG_ACTIVATED (0)	L'archivage est démarré au démarrage de Runtime.
TlgTagTriggerScan	1 seconde	Veuillez tenir compte du fait que "1 seconde" est uniquement le nom du déclenchement. Vous devez vous-même faire en sorte que le déclenchement existe et que son cycle soit réellement d'1s.
TlgTagTriggerArchiving	1 seconde	Veuillez tenir compte du fait que "1 seconde" est uniquement le nom du déclenchement. Vous devez vous-même faire en sorte que le déclenchement existe et que son cycle soit réellement d'1s.
TlgTagTriggerFactor	1	Le cycle d'affichage et le cycle d'archivage sont identiques.

Propriété	Valeur par défaut (nom de l'Enum => valeur)	Remarques
TlgTagOnError	TLG_TAG_LAST_VALUE (1)	La dernière valeur valide est prise comme valeur de substitution.
TlgTagTriggerType	TLG_TAG_ALWAYS (4)	Chaque valeur est archivée.
TlgTagMethodType	TLG_TAG_ACTUAL (1)	Pas de traitement. La valeur est admise directement.
TlgTagStartTriggerFunction	Aucune fonction spécifiée	--
TlgTagStopTriggerFunction	Aucune fonction spécifiée	--
TlgTagTriggerFunction	Aucune fonction spécifiée	--
TlgTagUpperLimit	Aucune valeur spécifiée	--
TlgTagLowerLimit	Aucune valeur spécifiée	--
TlgTagNameCompressArchive	Aucun nom d'archive spécifié	--
TlgTagNameCompressTag	Aucun nom de variable spécifié	--
TlgTagNameRawValue	Aucune variable de données brutes spécifiée	--
TlgTagStartTriggerModule	Aucun nom de DLL spécifié	--
TlgTagNameProcTag	correspond au "TagName"	--
TlgTagOnChange	TLG_TAG_EVERY_VALUE (0)	Chaque valeur est archivée.
TlgTagHysterese	0	Aucune hystérésis n'est utilisée pour la vérification.
TlgTagAliasName	Aucune valeur spécifiée	--
TlgTagStartEvent	Aucune variable spécifiée	--
TlgTagStopEvent	Aucune variable spécifiée	--

Liste des énumérateurs pour Tag Logging

Types d'Enum	Description
TLG_TAG_TYPE	Les paramètres transmis indiquent le type de la variable. Les types possibles figurent dans le tableau Enum HMIGO_TLG_TAG_TYPE.
TLG_TAG_ARCHIVING	Les paramètres transmis indiquent le type de la saisie. Les valeurs possibles figurent dans le tableau Enum HMIGO_TLG_TAG_ARCHIVING.
TLG_TAG_ARCHIVING_STATE	Le paramètre transmis indique si l'archivage est autorisé ou bloqué au démarrage du système. Les valeurs possibles figurent dans le tableau Enum HMIGO_TLG_TAG_ARCHIVING_STATE.
TLG_TAG_ON_ERROR	Le paramètre transmis indique quelle valeur doit être enregistrée lors d'un défaut : la dernière à avoir être saisie ou la valeur de substitution. Les valeurs possibles figurent dans le tableau Enum HMIGO_TLG_TAG_ON_ERROR.

Types d'Enum	Description
TLG_TAG_TRIGGER_TYPE	Le paramètre transmis indique comment l'archivage se déroule lors d'un changement d'état. Les valeurs possibles figurent dans le tableau Enum HMIGO_TLG_TAG_TRIGGER_TYPE.
TLG_TAG_METHOD_TYPE	Le paramètre transmis indique quelle méthode est employée pour le traitement de la valeur avant l'archivage. Les valeurs possibles figurent dans le tableau Enum HMIGO_TLG_TAG_METHOD_TYPE.
TLG_TAG_ON_CHANGE	Le paramètre transmis indique si un archivage doit avoir lieu en cas de modification. Les valeurs possibles figurent dans le tableau Enum HMIGO_TLG_TAG_ON_CHANGE.

Enum HMIGO_TLG_TAG_TYPE

Valeurs	Description
TLG_TAG_TYP_ANALOG (65537)	Variable analogique
TLG_TAG_TYP_BINARY (65538)	Variable binaire
TLG_TAG_TYP_PROCESS (65544)	Variable de process
TLG_TAG_TYP_COMPRESS (65540)	Variable des archives agrégées

Enum HMIGO_TLG_TAG_ARCHIVING

Valeurs	Description
TLG_TAG_ACYCLIC (8388609)	Saisie acyclique
TLG_TAG_CYCLIC_CONTINUOUS (8388610)	Saisie cyclique continue
TLG_TAG_CYCLIC_SELECTIVE (8388612)	Saisie cyclique sélective
TLG_TAG_ON_EVERY_CHANGE (8388616)	Saisie seulement en cas de modification

Enum HMIGO_TLG_TAG_ARCHIVING_STATE

Valeurs	Description
TLG_TAG_LOCKED (1)	Saisie bloquée lors du démarrage du système.
TLG_TAG_ACTIVATED (0)	Saisie autorisée lors du démarrage du système.

Enum HMIGO_TLG_TAG_ON_ERROR

Valeurs	Description
TLG_TAG_LAST_VALUE (1)	La dernière valeur valide à avoir été saisie est utilisée.
TLG_TAG_SUBSTITUTE (2)	Une valeur de substitution est entrée.

Enum HMIGO_TLG_TAG_TRIGGER_TYPE

Valeurs	Description
TLG_TAG_FROM_0_TO_1 (2)	Changement d'état de signal de la valeur 0 à 1.
TLG_TAG_FROM_1_TO_0 (3)	Changement d'état de signal de la valeur 1 à 0.
TLG_TAG_ALWAYS (4)	Toujours archiver.
TLG_TAG_EVERY_CHANGE (1)	Archiver à chaque changement d'état.

Enum HMIGO_TLG_TAG_METHOD_TYPE

Valeurs	Description
TLG_TAG_ACTUAL (1)	La valeur actuelle est admise.
TLG_TAG_SUM (3)	La somme est constituée.
TLG_TAG_MaxValue (5)	La valeur la plus grande est enregistrée.
TLG_TAG_MinValue (4)	La valeur la plus petite est enregistrée.
TLG_TAG_AVERAGE (2)	La valeur moyenne est enregistrée.

Enum HMIGO_TLG_TAG_ON_CHANGE

Valeurs	Description
TLG_TAG_EVERY_VALUE (0)	La valeur actuelle est admise.
TLG_TAG_RELATIVE_HYSTERESE (1)	Une hystérésis en pourcentages est utilisée pour calculer si la valeur doit être archivée.
TLG_TAG_ABSOLUTE_HYSTERESE (2)	Une hystérésis en nombres absolus est utilisée pour calculer si la valeur doit être archivée.

Exemple

```
Sub CreateTlgTag()
' HMIGO_011
' procedure to create a tag logging tag
' the archive need to be created before
' the tag logging tag must not be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Dim strTlgTagName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
  strTlgTagName = "NewTag"

 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Tag"
 'create tag logging tag
```

```
  objHMIGO.CreateTlgTag strArchiveName, strTlgTagName, TLG_TAG_TYPE_ANALOG
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Archive"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgTag (Page 2511)
Fonction DeleteTlgArchive (Page 2510)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
Fonction CloseTlgArchive (Page 2497)
Fonction CloseTlgTag (Page 2498)
VBA dans Tag Logging (Page 2492)

Fonction DeleteTlgArchive

Description

Supprime l'archive indiquée.

Syntaxe

```
Expression.DeleteTlgArchive(ArchiveName)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ArchiveName (chaîne de caractères)	Nom de l'archive à supprimer. Les variables d'archive contenues dans l'archive sont également supprimées.

Exemple

```
Sub DeleteTlgArchive()
' HMIGO_012
' procedure to delete an archive
' the archive need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"

 'delete tag logging archive
  objHMIGO.DeleteTlgArchive strArchiveName
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgTag (Page 2511)
Fonction CreateTlgTag (Page 2505)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
CommitTlgArchive-Funktion (Page 2499)
Fonction CloseTlgTag (Page 2498)
Fonction CloseTlgArchive (Page 2497)
VBA dans Tag Logging (Page 2492)

Fonction DeleteTlgTag

Description

Supprime la variable d'archive spécifiée.

Syntaxe

```
Expression.DeleteTlgTag(ArchiveName,TagName)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ArchiveName (chaîne de caractères)	Nom de l'archive contenant la variable d'archive devant être supprimée.
TagName (chaîne de caractères)	Nom de la variable d'archive à supprimer.

Exemple

```
Sub DeleteTlgTag()
' HMIGO_013
' procedure to delete a tag logging tag
' the archive need to be created before
' the tag logging tag need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Dim strTlgTagName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
  strTlgTagName = "NewTag"

 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Tag"
 'delete tag logging tag
  objHMIGO.DeleteTlgTag strArchiveName, strTlgTagName
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgArchive (Page 2510)
Fonction CreateTlgTag (Page 2505)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
CommitTlgArchive-Funktion (Page 2499)
Fonction CloseTlgTag (Page 2498)
Fonction CloseTlgArchive (Page 2497)
VBA dans Tag Logging (Page 2492)

Fonction GetTlgArchive

Description

Lit les paramètres de la variable indiquée.

Vous pouvez modifier ou lire les paramètres via les propriétés de l'objet. Vous trouverez une liste des propriétés de l'objet disponibles dans cette documentation sous "VBA dans TagLogging".

Syntaxe

```
Expression.GetTlgArchive(ArchiveName)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ArchiveName (chaîne de caractères)	Nom de l'archive dont les valeurs doivent être lues.

Exemple

```
Sub GetTlgArchive()
' HMIGO_014
' procedure to open an archive
' the archive need to be created before
```

```
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
 'open/ get tag logging archive
  objHMIGO.GetTlgArchive strArchiveName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgArchive, vbOKOnly, "Status Tlg Archive"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction CreateTlgTag (Page 2505)

Fonction ListTlgTag (Page 2518)

Fonction ListTlgArchive (Page 2516)

Fonction GetTlgArchive (Page 2513)

Fonction DeleteTlgTag (Page 2511)

Fonction DeleteTlgArchive (Page 2510)

Fonction CreateTlgArchive (Page 2503)

CommitTlgTag-Funktion (Page 2501)

CommitTlgArchive-Funktion (Page 2499)

Fonction CloseTlgTag (Page 2498)

Fonction CloseTlgArchive (Page 2497)

VBA dans Tag Logging (Page 2492)

Fonction GetTlgTag

Description

Lit les paramètres de la variable d'archive indiquée.

Vous pouvez modifier ou lire les paramètres via les propriétés de l'objet. Vous trouverez une liste des propriétés de l'objet disponibles dans cette documentation sous "VBA dans TagLogging".

Syntaxe

```
Expression.GetTlgTag(ArchiveName,TagName)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ArchiveName (chaîne de caractères)	Nom de l'archive contenant la variable d'archive.
TagName	Nom des variables d'archive dont les paramètres doivent être lus.

Exemple

```
Sub GetTlgTag()
' HMIGO_015
' procedure to open a tag logging tag
' the archive need to be created before
' the tag logging need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim strArchiveName As String
  Dim strTlgTagName As String
  Set objHMIGO = New HMIGO
  strArchiveName = "NewArchive"
  strTlgTagName = "NewTag"

 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Tag"
 'open/ get tag logging tag
  objHMIGO.GetTlgTag strArchiveName, strTlgTagName
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateTlgTag, vbOKOnly, "Status Tlg Archive"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction CreateTlgTag (Page 2505)
Fonction ListTlgTag (Page 2518)
Fonction ListTlgArchive (Page 2516)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgTag (Page 2511)
Fonction DeleteTlgArchive (Page 2510)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
CommitTlgArchive-Funktion (Page 2499)
Fonction CloseTlgTag (Page 2498)
Fonction CloseTlgArchive (Page 2497)
VBA dans Tag Logging (Page 2492)

Fonction ListTlgArchive

Description

La fonction ListTlgArchive renvoie en alternative dans une liste les valeurs suivantes du Tag Logging:

- toutes les archives Tag Logging
- tous les cycles existant / Temporisateur

Syntaxe

```
Expression.ListTlgArchive(ListType,pListArray,[Filter])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ListType (HMIGO_TLG_ARCHIVE_LIST_TYPE)	Détermine quel sera le contenu renvoyé dans une liste. Solutions possibles: • TLG_ARCHIVE_NAMES (1) toutes les archives Tag Logging créées • TLG_ARCHIVE_TRIGGER_NAMES (2) toutes les temporisations / temporisateurs créés
pListArray (Variant)	Liste du contenu exigé.
Filtre (String)	Des filtres peuvent également être placés en option. Un nom de déclenchement peut respectivement être utilisé comme filtre. Les Wildcards "*" et "?" sont acceptés.

Exemple

Dans l'exemple suivant il est vérifié si des archives sont configurées:

```
Sub ReadTlgArchives()
'HMIGO_028
'read content in tag logging
'no archives are implemented
  Dim objHMIGO As New HMIGO
  Dim arrContent As Variant
'read all tlg archives
  objHMIGO.ListTlgArchive TLG_ARCHIVE_NAMES, arrContent
'check result
  If (UBound(arrContent) - LBound(arrContent) + 1) <= 0 Then
    MsgBox "no entries because no tag logging archives are implemented"
    End If
End Sub
```

Voir aussi

Fonction ListTlgTag (Page 2518)
Fonction GetTlgArchive (Page 2513)
Fonction DeleteTlgTag (Page 2511)
Fonction DeleteTlgArchive (Page 2510)
Fonction CreateTlgTag (Page 2505)
Fonction CreateTlgArchive (Page 2503)
CommitTlgTag-Funktion (Page 2501)
CommitTlgArchive-Funktion (Page 2499)
Fonction CloseTlgTag (Page 2498)
Fonction CloseTlgArchive (Page 2497)
VBA dans Tag Logging (Page 2492)

Fonction ListTlgTag

Description

La fonction ListTlgTag renvoie dans une liste toutes les variables créées pour une archive ou toutes les archives de Tag Logging.

Syntaxe

```
Expression.ListTlgTag(ListType,ListArray,[ArchiveName],[Filter])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ListType (HMIGO_TLG_TAG_LIST_TYPE)	Détermine quel sera le contenu renvoyé dans une liste. Solutions possibles: TLG_TG_NAMES (1) toutes les variables créées dans une archive de Tag Logging
ListArray (Variant)	Liste du contenu exigé.
ArchiveName (chaîne de caractères)	Nom de l'archive dans Tag Logging (optionnel). Les variables de toutes les archives sont renvoyées si le nom de l'archive n'est pas spécifié.
Filtre (String)	Des filtres peuvent également être placés en option. Les Wildcards "*" et "?" sont acceptés.

Exemple

Dans l'exemple suivant il est vérifié si des variables d'archive sont configurées dans l'archive "Archive processus":

```
Sub ReadTlgTag()
'HMIGO_029
'read content in tag logging
'no tags within archives are implemented
  Dim objHMIGO As New HMIGO
  Dim arrContent As Variant
  Dim strArchive as String
'set tlg archive name
  strArchive = "processarchive"
'read all tlg tags in specified archive
  objHMIGO.ListTlgTag TLG_TAG_NAMES, arrContent, strArchive
'check result
  If (UBound(arrContent) - LBound(arrContent) + 1) <= 0 Then
    MsgBox "no entries because no tag logging tags in specified archive are implemented"
    End If
End Sub
```

Voir aussi

Fonction ListTlgArchive (Page 2516)

Fonction GetTlgArchive (Page 2513)

Fonction DeleteTlgTag (Page 2511)

Fonction DeleteTlgArchive (Page 2510)

Fonction CreateTlgTag (Page 2505)

Fonction CreateTlgArchive (Page 2503)

CommitTlgTag-Funktion (Page 2501)

CommitTlgArchive-Funktion (Page 2499)

Fonction CloseTlgTag (Page 2498)

Fonction CloseTlgArchive (Page 2497)

VBA dans Tag Logging (Page 2492)

3.5.2.4 VBA dans Text Library

VBA dans Text Library

Introduction

VBA vous permet de générer, modifier ou supprimer des textes de Text Library ainsi que de lire des TextIDs et des textes directement à partir du code.

Principe

Si vous avez généré l'instance de la classe HMIGO, les fonctions suivantes sont disponibles pour l'accès à TextLibrary :

- CreateTextLanguage
- CreateText
- DeleteText
- DeleteTextLanguage
- GetText
- GetTextID
- ListText
- ModifyText

Les énumérations "HMIGO_TEXT_CREATE_MODE" et "HMIGO_TEXT_LIST_TYPE" vous permettent d'alimenter les paramètres de ces fonctions.

Remarque

Au cours de l'édition avec VBA, TextLibrary ne doit pas être ouverte.

Voir aussi

Fonction ModifyText (Page 2531)
Fonction ListText (Page 2529)
Fonction GetTextID (Page 2527)
Fonction GetText (Page 2526)
Fonction DeleteTextLanguage (Page 2525)
Fonction DeleteText (Page 2523)
Fonction CreateText (Page 2522)
Fonction CreateTextLanguage (Page 2520)
VBA dans d'autres éditeurs WinCC (Page 2479)

Fonction CreateTextLanguage

Description

Crée un langue dans la Text Library.

Syntaxe

```
Expression.CreateTextLanguage(LanguageID)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
LanguageID (Long)	ID de la langue devant être créée, z.B. 1031 pour le français, 1033 pour l'anglais, etc. Vous trouverez une table des codes de langue dans l'aide en ligne de WinCC en recherchant sous "identificateurs de langue".

Exemple

```
Sub CreateTextLanguage()
' HMIGO_016
' procedure to create a language in text library
' language must not be created before
' LanguageID german = 1031
' LanguageID english(US) = 1033
' LanguageID spanish = 1034
' LanguageID french = 1040
' LanguageID farsi= 1065
' declarations
  Dim objHMIGO As HMIGO
  Dim lngLangugeNumber As Long
  Set objHMIGO = New HMIGO
  lngLangugeNumber = 1065     'farsi
 'create new language
  objHMIGO.CreateTextLanguage lngLangugeNumber
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ModifyText (Page 2531)

Fonction ListText (Page 2529)

Fonction GetTextID (Page 2527)

Fonction GetText (Page 2526)

Fonction DeleteTextLanguage (Page 2525)

Fonction DeleteText (Page 2523)

Fonction CreateText (Page 2522)

VBA dans Text Library (Page 2519)

Fonction CreateText

Description

Crée un texte pour la langue indiquée. Les entrées de texte pour d'autres langues peuvent être complétés avec ModifyText.

Syntaxe

```
Expression.CreateText(LanguageID,Text,CreateMode,TextID)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
LanguageID (Long)	ID de la langue dans laquelle le texte est créé.
Text (chaîne de caractères)	Texte à créer.
CreateMode (HMIGO_TEXT_CREATE_MODE)	Mode de la création de texte : • TEXT_ADD_REFCOUNT (0) augmente uniquement le compteur de référence si un texte identique existe déjà. • TEXT_CREATE_ALWAYS (1) crée toujours une nouvelle ligne de texte et y insère le texte.
TextID (Long)	Renvoie le TextID reçu par le texte créé ou le TextID dont le compteur de références est augmenté. Cet ID est nécessaire pour l'édition du texte sous d'autre fonctions.

Exemple

```
Sub CreateText()
' HMIGO_017
' procedure to create a new text
' declarations
  Dim objHMIGO As HMIGO
  Dim lngLanguageID As Long
  Dim lngTextCreateMode As Long
  Dim lngTextID As Long         'return value of ".CreateText"
  Dim strText As String
  Set objHMIGO = New HMIGO
    strText = "new text"
  'LanguageID = english
  lngLanguageID = 1033
  '"TEXT_ADD_REFCOUNT" check if text exists, if not create new text
  lngTextCreateMode = 0
```

```
  '"TEXT_CREATE_ALWAYS" create always a new text (for messages)
'  lngTextCreateMode = 1

 'create new text
  objHMIGO.CreateText lngLanguageID, strText, lngTextCreateMode, lngTextID
 'show TextID of created text
  MsgBox "TextID: " & lngTextID, vbOKOnly, "Result CreateText"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ModifyText (Page 2531)

Fonction ListText (Page 2529)

Fonction GetTextID (Page 2527)

Fonction GetText (Page 2526)

Fonction DeleteTextLanguage (Page 2525)

Fonction DeleteText (Page 2523)

Fonction CreateTextLanguage (Page 2520)

VBA dans Text Library (Page 2519)

Fonction DeleteText

Description

Supprime une ligne de texte. Toutes les langues pour la ligne de texte concernée ainsi que la ligne de texte même sont supprimées.

Syntaxe

```
Expression.DeleteText(TextID)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
TextID (Long)	ID de la ligne de texte à supprimer.

Exemple

```
Sub DeleteText()
' HMIGO_018
' procedure to delete a text
' text will be searched and deleted
' declarations
  Dim objHMIGO As HMIGO
  Dim lngLanguageID As Long
  Dim lngTextID As Long         'return value of GetTextID
  Dim strText As String
On Error GoTo ErrorHandler
  Set objHMIGO = New HMIGO
    strText = "new text"
  lngLanguageID = 1033

 'first: find text in text library and return TextID
  objHMIGO.GetTextID 1033, strText, lngTextID

 'if searched text exists: delete this text
  If Not lngTextID = -1 Then
    objHMIGO.DeleteText lngTextID
    MsgBox "Text : """ & strText & """ found in TextID: " & lngTextID & vbNewLine & _
           "TextID is deleted!", vbOKOnly, "Result DeleteText"
    Else
    MsgBox "Text : """ & strText & """ not found." & vbNewLine & _
           "No Text deleted!", vbOKOnly, "Result DeleteText"
    End If
  Set objHMIGO = Nothing
    Exit Sub
ErrorHandler:
 'if lngText = (-1),searched text does not exit
  If lngTextID = -1 Then
    'reset errorhandler
    Err.Clear
Resume Next
    End If
  MsgBox "ErrNr. : " & Err.Number & vbNewLine & _
         "ErrDes.: " & Err.Description, vbOKOnly, "Error ocurred"
    'reset errorhandler
    Err.Clear
End Sub
```

Voir aussi

VBA dans Text Library (Page 2519)
Fonction ModifyText (Page 2531)
Fonction ListText (Page 2529)
Fonction GetTextID (Page 2527)
Fonction GetText (Page 2526)
Fonction DeleteTextLanguage (Page 2525)
Fonction CreateText (Page 2522)
Fonction CreateTextLanguage (Page 2520)

Fonction DeleteTextLanguage

Description

Permet de supprimer une langue dans TextLibrary. Tous les textes émis dans ces langues sont également supprimés.

Syntaxe

```
Expression.DeleteTextLanguage(LanguageID)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
LanguageID (Long)	ID de la langue à supprimer.

Exemple

Dans l'exemple suivant il est vérifié si la langue '1036' existe. Si c'est le cas, cette langue est supprimée.

```
Sub DeleteLanguage()
'HMIGO_030
' delete an existing languages in TextLibrary
' language '1036'/spanish has to exist
  Dim objHMIGO As New HMIGO
  Dim arrContent As Variant
  Dim intLanguage As Long
  Dim lngPointer As Long
```

```
' get all existing languages
  objHMIGO.ListText TEXT_LANGUAGE_IDS, arrContent
' check requested list for language '1036'/ spanish and delete
  For lngPointer = LBound(arrContent) To UBound(arrContent)
    intLanguage = arrContent(lngPointer) + Val("&H400")
    If intLanguage = 1036 Then
     'delete language
      objHMIGO.DeleteTextLanguage intLanguage
    End If
  Next lngPointer
End Sub
```

Voir aussi

Fonction GetText (Page 2526)
Fonction ModifyText (Page 2531)
Fonction ListText (Page 2529)
Fonction GetTextID (Page 2527)
Fonction DeleteText (Page 2523)
Fonction CreateText (Page 2522)
Fonction CreateTextLanguage (Page 2520)
VBA dans Text Library (Page 2519)

Fonction GetText

Description

Renvoie dans la langue sélectionnée le texte correspondant au TextID.

Syntaxe

```
Expression.GetText(LanguageID,TextID,Text)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGeneralObjects".

Paramètres

Paramètre (type de données)	Description
LanguageID (Long)	ID de la langue du texte devant être lu.
TextID (Long)	ID de la ligne de texte dont le texte doit être lu.
Text (chaîne de caractères)	Renvoie le texte du TextID et de la langue sélectionnés.

Exemple

```
Sub GetText()
' HMIGO_019
' procedure to get a text
' text with TextID = '69' need to be created
' declarations
  Dim objHMIGO As HMIGO
  Dim lngLanguageID As Long
  Dim lngTextID As Long
  Dim strText As String         'return value of GetText
  Set objHMIGO = New HMIGO
  lngTextID = 69
  lngLanguageID = 1033

 'find text text library
  objHMIGO.GetText lngLanguageID, lngTextID, strText

 'show found text
  MsgBox "Read Text in TextID : " & lngTextID & " is """ & strText & """ !", _
                              vbOKOnly, "Result GetText"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ModifyText (Page 2531)

Fonction ListText (Page 2529)

Fonction GetTextID (Page 2527)

Fonction DeleteTextLanguage (Page 2525)

Fonction DeleteText (Page 2523)

Fonction CreateText (Page 2522)

Fonction CreateTextLanguage (Page 2520)

VBA dans Text Library (Page 2519)

Fonction GetTextID

Description

Renvoie dans la langue sélectionnée l'ID correspondant au texte recherché.

Si plusieurs textes ont le même contenu, seulement ID le plus bas sera renvoyé. Le CreateMode de la fonction CreateText détermine plusieurs lignes de texte contiennent le même texte.

Syntaxe

```
Expression.GetTextID(LanguageID,Text,TextID)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
LanguageID (Long)	ID de la langue du texte recherché.
Text (chaîne de caractères)	Le texte recherché.
TextID (Long)	ID de la ligne de texte dans laquelle le texte recherché a été trouvé.

Exemple

```
Sub GetTextID()
' HMIGO_020
' procedure to search a TextID
' text will be searched and a TextID will be returned
' declarations
  Dim objHMIGO As HMIGO
  Dim lngLanguageID As Long
  Dim lngTextID As Long         'return value of GetTextID
  Dim strText As String
On Error GoTo ErrorHandler
  Set objHMIGO = New HMIGO
  strText = "old text"
  lngLanguageID = 1033

 'first: find text in text library and return TextID
  objHMIGO.GetTextID 1033, strText, lngTextID

 'if searched text exists: delete this text
  If Not lngTextID = -1 Then
    MsgBox "Text : """ & strText & """ found in TextID: " & lngTextID, _
                   vbOKOnly, "Result GetTextID"
    Else
    MsgBox "Text : """ & strText & """ not found!", vbOKOnly, "Result GetTextID"
    End If
  Set objHMIGO = Nothing
    Exit Sub
ErrorHandler:
 'if lngText = (-1),searched text does not exit
  If lngTextID = -1 Then
    'reset errorhandler
    Err.Clear
Resume Next
    End If
```

```
  MsgBox "ErrNr. : " & Err.Number & vbNewLine & _
         "ErrDes.: " & Err.Description, vbOKOnly, "Error ocurred"
    'reset errorhandler
    Err.Clear
End Sub
```

Voir aussi

Fonction ModifyText (Page 2531)

Fonction ListText (Page 2529)

Fonction GetText (Page 2526)

Fonction DeleteTextLanguage (Page 2525)

Fonction DeleteText (Page 2523)

Fonction CreateText (Page 2522)

Fonction CreateTextLanguage (Page 2520)

VBA dans Text Library (Page 2519)

Fonction ListText

Description

En alternative, la fonction ListText renvoie sous forme de liste le contenu suivant de TextLibrary:

- toutes les langues créées
- toutes les ID des textes
- le texte intégral d'une langue

Syntaxe

```
Expression.ListText(ListType,pListArray,[LanguageID],[Filter])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ListType (HMIGO_TEXT_LIST_TYPE)	Détermine quel sera le contenu renvoyé sous forme de liste. Solutions possibles: • TEXT_LANGUAGE_IDS (1) toutes les langues créées. Le résultat doit encore être converti en ajoutant 400hex. • TEXT_IDS (2) les ID de tous les textes. • TEXT_TEXTS (3) tous les textes d'une langue.
pListArray (Variant)	Liste du contenu exigé.
LanguageID (Long)	ID de la langue dont le texte est renvoyé.
Filtre (String)	Des filtres peuvent également être utilisés en option. Les Wildcard "*" et "?" sont acceptés.

Exemple

Dans l'exemple suivant, il est vérifié si la liste contenant le texte d'une langue est vide étant donné que celle-ci n'existe pas:

```
Sub ReadTextsByLanguage()
'HMIGO_031
'read content in textLibrary by language
  Dim objHMIGO As New HMIGO
  Dim arrContent As Variant
  Dim intLanguage As Integer
'set invalid language ID
  intLanguage = 1051 'language does not exist
'read all texts
  objHMIGO.ListText TEXT_TEXTS, arrContent, intLanguage
'check result
  If (UBound(arrContent) - LBound(arrContent) + 1) <= 0 Then
    MsgBox "no entries because wrong language selection"
    End If
End Sub
```

Voir aussi

Fonction ModifyText (Page 2531)

Fonction GetTextID (Page 2527)

Fonction GetText (Page 2526)

Fonction DeleteTextLanguage (Page 2525)

Fonction DeleteText (Page 2523)

Fonction CreateTextLanguage (Page 2520)

VBA dans Text Library (Page 2519)

Fonction ModifyText

Description

Modifie avec l'ID sélectionnée le texte correspondant à la langue sélectionnée.

Syntaxe

```
Expression.ModifyText(LanguageID,TextID,Text)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
LanguageID (Long)	ID de la langue du texte devant être modifié.
TextID (Long)	ID de la ligne de texte devant être modifié.
Text (chaîne de caractères)	Nouveau texte à insérer.

Exemple

```
Sub ModifyText()
' HMIGO_021
' procedure to modify a text
' text will be searched and replaced
' declarations
  Dim objHMIGO As HMIGO
  Dim lngLanguageID As Long
  Dim lngTextID As Long         'return value of GetTextID
  Dim strOldText As String
  Dim strNewText As String
On Error GoTo ErrorHandler
  Set objHMIGO = New HMIGO
  strOldText = "old text"
  strNewText = "new text"
  lngLanguageID = 1033

 'first: find text in text library and return TextID
  objHMIGO.GetTextID 1033, strOldText, lngTextID

 'if searched text exists: replace this text
  If Not lngTextID = -1 Then
    objHMIGO.ModifyText lngLanguageID, lngTextID, strNewText
    MsgBox "Text : """ & strOldText & """ found in TextID: " & lngTextID & vbNewLine & _
           "Text replaced with : """ & strNewText & """ !", vbOKOnly, "Result DeleteText"
    Else
    MsgBox "Text : """ & strOldText & """ not found." & vbNewLine & _
```

```
            "No Replacements done!", vbOKOnly, "Result DeleteText"
     End If
   Set objHMIGO = Nothing
 Exit Sub
 ErrorHandler:
  'if lngText = (-1),searched text does not exit
   If lngTextID = -1 Then
     'reset errorhandler
     Err.Clear
     Resume Next
     End If
   MsgBox "ErrNr. : " & Err.Number & vbNewLine & _
          "ErrDes.: " & Err.Description, vbOKOnly, "Error ocurred"
     'reset errorhandler
     Err.Clear
 End Sub
```

Voir aussi

Fonction ListText (Page 2529)
Fonction GetTextID (Page 2527)
Fonction GetText (Page 2526)
Fonction DeleteTextLanguage (Page 2525)
Fonction DeleteText (Page 2523)
Fonction CreateText (Page 2522)
Fonction CreateTextLanguage (Page 2520)
VBA dans Text Library (Page 2519)

3.5.2.5 VBA dans Alarm Logging

VBA dans Alarm Logging

Introduction

VBA vous permet de générer, modifier ou supprimer des alarmes directement à partir du code.

Principe

Si vous avez généré l'instance de la classe HMIGO, les fonctions suivantes sont disponibles pour l'accès à Alarm Logging :

- CloseSingleAlarm
- CommitSingleAlarm
- CreateSingleAlarm

- DeleteSingleAlarm
- GetSingleAlarm
- ListSingleAlarm

Les énumérations "HMIGO_SINGLE_ALARM_CLASS_IDS" et "HMIGO_SINGLE_ALARM_LIST_TYPE" vous permettent d'affecter les paramètres de ces fonctions.

Remarque

Au cours de l'édition avec VBA, Alarm Logging ne doit pas être ouvert.

Accès aux propriétés de l'objet

Vous pouvez aussi accéder directement dans VBA aux paramètres des fonctions citées ci-dessus par les propriétés de l'objet :

Propriété de l'objet	Description	Lecture/ écriture
ObjectStateSingleAlarm	Renvoie l'état de l'objet par le biais de l'énumération HMIGO_OBJECT_STATE. Vous trouverez sous "VBA dans d'autres éditeurs WinCC".	oui/non
SingleAlarmMessageNumber	Numéro de l'alarme	oui/non
SingleAlarmAGNumber	Numéro d'AS	oui/oui
SingleAlarmCPUNumber	Numéro de CPU de l'API	oui/oui

Propriété de l'objet	Description	Lecture/ écriture
SingleAlarmClassID	Classe des alarmes. Valeurs possibles pour Enum SINGLE_ALARM_CLASS_IDS : • SINGLE_ALARM_ERROR (1) • SINGLE_ALARM_CLASS_2 (2) • SINGLE_ALARM_CLASS_3 (3) • SINGLE_ALARM_CLASS_4 (4) • SINGLE_ALARM_CLASS_5 (5) • SINGLE_ALARM_CLASS_6 (6) • SINGLE_ALARM_CLASS_7 (7) • SINGLE_ALARM_CLASS_8 (8) • SINGLE_ALARM_CLASS_9 (9) • SINGLE_ALARM_CLASS_10 (10) • SINGLE_ALARM_CLASS_11 (11) • SINGLE_ALARM_CLASS_12 (12) • SINGLE_ALARM_CLASS_13 (13) • SINGLE_ALARM_CLASS_14 (14) • SINGLE_ALARM_CLASS_15 (15) • SINGLE_ALARM_CLASS_16 (16) • SINGLE_ALARM_CLASS_SYSTEM_REQUIRE_ACKNOWLEDGEMENT (17) • SINGLE_ALARM_CLASS_SYSTEM_WITHOUT_ACKNOWLEDGEMENT (18)	oui/oui
SingleAlarmMessageTypeID	ID de type de l'alarme. Les valeurs autorisées dépendent de la classe d'alarme : • Classe 1 : Valeurs de 1 à 16 • Classe 2 : Valeurs de 17 à 32 • Classe 3 : Valeurs de 33 à 48 • ... • Classe 18 : Valeurs de 263 à 288	oui/oui
SingleAlarmTextXXID XX = 1...10	Pour les textes utilisateurs de 1 à 10 les propriétés SingleAlarmText1ID à SingleAlarmText10ID sont disponibles	oui/oui
SingleAlarmTagNameProcessValueXX XX = 1...10	Pour les valeurs de process à archiver, il existe les propriétés SingleAlarmTagNameProcessValue1 à 10 Si vous voulez supprimer une valeur de process configurée, il vous faut décrire ce paramètre avec une variable de type "Long" dont la valeur est "0".[1)]	oui/oui
SingleAlarmTagName	Nom de variable pour événement	oui/oui
SingleAlarmMessageBit	Bits pour procédé de signalisation par bit	oui/oui
SingleAlarmQuitTag	Nom de variable pour état d'acquittement	oui/oui
SingleAlarmQuitBits	Bit pour procédé de signalisation par bit	oui/oui
SingleAlarmStateTag	Variable pour interrogation de l'état	oui/oui
SingleAlarmStateBits	Bit pour variable d'état	oui/oui
SingleAlarmNormDLL	Nom de la DLL de normalisation.	oui/oui
SingleAlarmQuitSingle	Acquittement de l'alarme, possibilités : VRAI ou FAUX	oui/oui

Propriété de l'objet	Description	Lecture/ écriture
SingleAlarmHornActivate	Activation du signalisateur, possibilités : VRAI ou FAUX	oui/oui
SingleAlarmArchiving	Archivage de l'alarme, possibilités : VRAI ou FAUX	oui/oui
SingleAlarmProtocol	Impression de l'alarme, possibilités : VRAI ou FAUX	oui/oui
SingleAlarmFlankInvert	Emission de l'alarme si front descendant, possibilités : VRAI ou FAUX	oui/oui
SingleAlarmLockedOnStart	Désactivation de l'alarme lors du démarrage du système, possibilités : VRAI ou FAUX	oui/oui
SingleAlarmGlobalAPFunction	Transmission de l'alarme à la fonction AP globale, possibilités : VRAI ou FAUX	oui/oui
SingleAlarmActionName	Nom de l'action	oui/oui
SingleAlarmActionParams	Paramètres de l'action	oui/oui
SingleAlarmInfoText	Texte d'info pour l'alarme	oui/oui
SingleAlarmGroup	Nom de l'alarme groupée définie par l'utilisateur à laquelle est assignée une alarme.	oui/oui

1)

```
Sub DeleteSingleAlarmTagNameProcessValue1()
     'HMIGO_033
     Dim objGO as HMIGO
     Dim var as Long
     var = 0
     Set objGO = new HMIGO
     'message 1 will be modified
       objGO.GetSingleAlarm 1
       objGO.SingleAlarmTagNameProcessValue1 = var
       objGO.CommitSingleAlarm
     Set objGO = nothing
End Sub
```

Voir aussi

Fonction ListSingleAlarm (Page 2544)
Fonction GetSingleAlarm (Page 2543)
Fonction DeleteSingleAlarm (Page 2542)
Fonction CreateSingleAlarm (Page 2538)
Fonction CommitSingleAlarm (Page 2537)
Fonction CloseSingleAlarm (Page 2536)
VBA dans d'autres éditeurs WinCC (Page 2479)

Fonction CloseSingleAlarm

Description

Ferme l'alarme ouverte.

Remarque

Les paramètres modifiés ne sont pas enregistrés. Si vous voulez enregistrer les valeurs actuelles, il vous faut à nouveau exécuter la fonction CommitSingleAlarm().

Syntaxe

```
Expression.CloseSingleAlarm()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CloseSingleAlarm()
' HMIGO_22
' procedure to close a singlealarm
' message #100 need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim lngMsgNumber As Long
  Set objHMIGO = New HMIGO
  lngMsgNumber = 100
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
 'open a singlealarm
  objHMIGO.GetSingleAlarm lngMsgNumber
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
 'close a singlealarm
  objHMIGO.CloseSingleAlarm
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"

  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListSingleAlarm (Page 2544)
Fonction GetSingleAlarm (Page 2543)
Fonction DeleteSingleAlarm (Page 2542)
Fonction CreateSingleAlarm (Page 2538)
Fonction CommitSingleAlarm (Page 2537)
VBA dans Alarm Logging (Page 2532)

Fonction CommitSingleAlarm

Description

Ecrit les paramètres modifiés de l'alarme ouverte dans WinCC.

Remarque

Si vous voulez modifier d'autres propriétés après un appel de CommitSingleAlarm, vous devez écrire ces modifications vers WinCC par un nouvel appel de la fonction.

Syntaxe

```
Expression.CommitSingleAlarm()
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

--

Exemple

```
Sub CommitSingleAlarm()
' HMIGO_023
' procedure to change a property of a singlealarm
' message #100 need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim lngMsgNumber As Long
  Dim lngMsgBitNumber As Long
  Set objHMIGO = New HMIGO
  lngMsgNumber = 100
  lngMsgBitNumber = 10
 'current status is "EMPTY"
```

```
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
 'open a singlealarm
  objHMIGO.GetSingleAlarm lngMsgNumber
 'current status is "OPENED" for changes
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
 'change a property
  objHMIGO.SingleAlarmMessageBit = lngMsgBitNumber
 'current status is "MODIFIED" for changes
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
 'commit a single alarm
  objHMIGO.CommitSingleAlarm
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"

  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListSingleAlarm (Page 2544)

Fonction GetSingleAlarm (Page 2543)

Fonction DeleteSingleAlarm (Page 2542)

Fonction CreateSingleAlarm (Page 2538)

Fonction CloseSingleAlarm (Page 2536)

VBA dans Alarm Logging (Page 2532)

Fonction CreateSingleAlarm

Description

Crée une alarme.

Syntaxe

```
Expression.CreateSingleAlarm(MessageNumber,ClassID,MessageTypeID,Tex
t1ID,MessageTagName,MessageBit)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
MessageNumber (Long)	Numéro de l'alarme. • Si vous spécifiez ici un numéro d'alarme non utilisé, il sera accepté. • Si la valeur "0" est entrée, le numéro d'alarme est attribué par le système. Dans ce cas, le numéro d'alarme est renvoyé.
ClassID (HMIGO_SINGLE_ALARM_CLASS_IDS)	Classe d'alarmes. Les valeurs possibles figurent dans le tableau Enum HMIGO_SINGLE_ALARM_CLASS_IDS.
MessageTypeID (nombre entier)	Les valeurs autorisées dépendent de la classe d'alarme : • Classe 1: Valeurs de 1 à 16 • Classe 2: Valeurs de 17 à 32 • Classe 3: Valeurs de 33 à 48 • ... • Classe 18: Valeurs de 263 à 288
Text1ID (Long)	ID du premier texte utilisateur. La fonction ModifySingleAlarm permet de définir de nouveaux textes utilisateur (1-10)
MessageTagName (chaîne de caractères)	Nom de variable pour l'événement.
MessageBit (nombre entier)	Bit en cas de procédé de signalisation par bit (0...31)

Valeurs par défaut pour la création d'une alarme

Le tableau suivant montre les valeurs par défaut entrées lors de la création d'une alarme. Ces propriétés peuvent être modifiées. La fonction ModifySingleAlarm permet d'enregistrer les modifications.

Paramètres	Valeur par défaut (nom de l'Enum => valeur)	Remarque
SingleAlarmAGNumber	0	--
SingleAlarmCPUNumber	0	--
SingleAlarmTextXXID	Aucun texte entré	--
SingleAlarmTagNameProcessValueXX	Aucune variable entrée	--
SingleAlarmQuitTag	Aucune variable entrée	--
SingleAlarmQuitBits	0	Pas de mise à 1 de bit.
SingleAlarmStateTag	Aucune variable entrée	Correspond à un jour exactement. Pertinent seulement pour les variables d'agrégation.
SingleAlarmStateBits	0	Pas de mise à 1 de bit.
SingleAlarmNormDLL	Aucun nom entré	--

Paramètres	Valeur par défaut (nom de l'Enum => valeur)	Remarque
SingleAlarmQuitSingle	FALSE	Acquittement individuel, pas d'acquittement groupé.
SingleAlarmHornActivate	FALSE	Pas d'activation de l'avertisseur
SingleAlarmArchiving	TRUE	L'alarme est archivée.
SingleAlarmProtocol	TRUE	L'alarme est imprimée.
SingleAlarmFlankInvert	FALSE	Non activé.
SingleAlarmLockedOnStart	FALSE	L'alarme est n'est pas bloqué.
Fonction SingleAlarmGlobalAPI	FALSE	L'alarme est n'est pas transmise.
SingleAlarmActionName	Aucun nom entré	--
SingleAlarmActionParams	Aucun paramètre entré pour l'action	--
SingleAlarmInfoText	Aucun texte entré	--
SingleAlarmGroup	Aucun texte entré	--

Enum HMIGO_SINGLE_ALARM_CLASS_IDS

On dispose des classes d'alarmes suivantes :

Valeurs	Description
SINGLE_ALARM_ERROR (1)	--
SINGLE_ALARM_CLASS_2 (2)	--
SINGLE_ALARM_CLASS_3 (3)	--
SINGLE_ALARM_CLASS_4 (4)	--
SINGLE_ALARM_CLASS_5 (5)	--
SINGLE_ALARM_CLASS_6 (6)	--
SINGLE_ALARM_CLASS_7 (7)	--
SINGLE_ALARM_CLASS_8 (8)	--
SINGLE_ALARM_CLASS_9 (9)	--
SINGLE_ALARM_CLASS_10 (10)	--
SINGLE_ALARM_CLASS_11 (11)	--
SINGLE_ALARM_CLASS_12 (12)	--
SINGLE_ALARM_CLASS_13 (13)	--
SINGLE_ALARM_CLASS_14 (14)	--
SINGLE_ALARM_CLASS_15 (15)	--
SINGLE_ALARM_CLASS_16 (16)	--
SINGLE_ALARM_CLASS_SYSTEM_REQUIRE_ACKNOWLEDGEMENT (17)	--
SINGLE_ALARM_CLASS_SYSTEM_WITHOUT_ACKNOWLEDGEMENT (18)	--

Exemple

```
Sub CreateSingleAlarm()
' HMIGO_024
' procedure to create a SingleAlarm
' message must not be created before
' message Text ID need to be created before in text library
' declarations
  Dim objHMIGO As HMIGO
  Dim strMsgText As String      'message text
  Dim strMsgTagName As String   'message variable
  Dim lngMsgNumber As Long      'message number
  Dim lngMsgBitNumber As Long   'bit number within the message variable
  Dim lngMsgTypeID As Long      'message type
  Dim lngMsgClassID             'SINGLE_ALARM_ERROR
  Dim lngMsgTextID As Long      'message text ID from textlibrary
  Set objHMIGO = New HMIGO
  strMsgText = "NewText"
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
 'preset reqiured parameter
  lngMsgNumber = 50
  lngMsgClassID = 1
  lngMsgTypeID = 2
  lngMsgTextID = 69
  strMsgText = "new text message"
  strMsgTagName = "NewVariable"
  lngMsgBitNumber = 5

 'create a tag
  objHMIGO.CreateSingleAlarm lngMsgNumber,SINGLE_ALARM_ERROR, lngMsgTypeID,lngMsgTextID,
strMsgTagName, lngMsgBitNumber

 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListSingleAlarm (Page 2544)

Fonction GetSingleAlarm (Page 2543)

Fonction DeleteSingleAlarm (Page 2542)

Fonction CommitSingleAlarm (Page 2537)

Fonction CloseSingleAlarm (Page 2536)

VBA dans Alarm Logging (Page 2532)

Fonction DeleteSingleAlarm

Description

Supprime le message d'alarme spécifié.

Syntaxe

```
Expression.DeleteSingleAlarm(MessageNumber)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
MessageNumber (Long)	Numéro de l'alarme à supprimer.

Exemple

```
Sub DeleteSingleAlarm()
' HMIGO_025
' procedure to delete a singlealarm
' message #100 need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim lngMsgNumber As Long

  Set objHMIGO = New HMIGO
  lngMsgNumber = 100
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"

 'delete a singlealarm
  objHMIGO.DeleteSingleAlarm lngMsgNumber
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

VBA dans Alarm Logging (Page 2532)
Fonction ListSingleAlarm (Page 2544)
Fonction GetSingleAlarm (Page 2543)
Fonction CreateSingleAlarm (Page 2538)
Fonction CommitSingleAlarm (Page 2537)
Fonction CloseSingleAlarm (Page 2536)

Fonction GetSingleAlarm

Description

Lit les paramètres de l'alarme indiquée.

Vous pouvez modifier ou lire les paramètres via les propriétés de l'objet. Vous trouverez une liste des propriétés de l'objet disponibles dans cette documentation sous "VBA dans Alarm Logging".

Syntaxe

```
Expression.GetSingleAlarm(MessageNumber)
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
MessageNumber (Long)	Le numéro de l'alarme à lire.

Exemple

```
Sub GetSingleAlarm()
' HMIGO_026
' procedure to open a singlealarm
' message #100 need to be created before
' declarations
  Dim objHMIGO As HMIGO
  Dim lngMsgNumber As Long
  Set objHMIGO = New HMIGO
  lngMsgNumber = 100
 'current status is "EMPTY"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
 'open/ get a tag
```

```
  objHMIGO.GetSingleAlarm lngMsgNumber
 'current status is "OPENED"
  MsgBox objHMIGO.ObjectStateSingleAlarm, vbOKOnly, "Status SingleAlarm"
  Set objHMIGO = Nothing
End Sub
```

Voir aussi

Fonction ListSingleAlarm (Page 2544)
Fonction DeleteSingleAlarm (Page 2542)
Fonction CreateSingleAlarm (Page 2538)
Fonction CommitSingleAlarm (Page 2537)
Fonction CloseSingleAlarm (Page 2536)
VBA dans Alarm Logging (Page 2532)

Fonction ListSingleAlarm

Description

La fonction ListSingleAlarm renvoie dans une liste le contenu suivant d'Alarm Logging:

- toutes les actions créées qui sont en corrélation avec des alarmes
- toutes les ID pour classes d'alarmes qui ont été créées
- tous les textes d'info
- tous les numéros d'alarme créés
- toutes les ID de types pour messages qui ont été créées
- toutes les classes d'alarmes créées
- toutes les alarmes groupées créées

Syntaxe

```
Expression.ListSingleAlarm(ListType,pListArray,[Filter])
```

Expression

Impératif. Une expression qui renvoie un objet du type "HMIGO".

Paramètres

Paramètre (type de données)	Description
ListType (HMIGO_SINGLE_ALARM_LIST_TYPE)	Détermine quel sera le contenu renvoyé dans une liste. Solutions possibles: • SINGLE_ALARM_ACTION_NAMES (1) toutes les actions pour Loop In Alarm créées, et ce lorsque le paramètre est utilisé comme chaîne de caractères dans la configuration • SINGLE_ALARM_CLASS_IDS (2) toutes les ID des classes d'alarmes créées • SINGLE_ALARM_INFO_TEXTS (3) tous les textes d'info créés • SINGLE_ALARM_MESSAGE_NUMBERS (4) tous les numéros d'alarme créés • SINGLE_ALARM_MESSAGE_TYPE_IDS (5) toutes les ID de types pour alarmes créées • SINGLE_ALARM_GROUP_MESSAGE_CLASSES (6) toutes les classes d'alarmes créées • SINGLE_ALARM_GROUP_MESSAGE_USER_DEFINED (7) toutes les alarmes groupées créées
pListArray (Variant)	Liste du contenu exigé.
Filtre (String)	Des filtres peuvent également être placés en option. Les Wildcards "*" et "?" sont acceptés.

Exemple

Dans l'exemple suivant il est vérifié si des textes d'info sont configurés:

```
Sub ReadSingleAlarm()
'HMIGO_032
'read content in alarm logging
'no info texts are implemented
  Dim objHMIGO As New HMIGO
  Dim arrContent As Variant
'read all info texts
  objHMIGO.ListSingleAlarm SINGLE_ALARM_INFO_TEXTS, arrContent
'check result
  If (UBound(arrContent) - LBound(arrContent) + 1) <= 0 Then
    MsgBox "no entries because no info texts are implemented"
    End If
End Sub
```

Voir aussi

Fonction CreateSingleAlarm (Page 2538)
Fonction CommitSingleAlarm (Page 2537)
Fonction CloseSingleAlarm (Page 2536)
VBA dans Alarm Logging (Page 2532)

Index

, 55

ajouter, 882, 883
DeactivateRTProject, 1506
InquireLanguage, 1508
TlgTrendWindowPressReportSaveButton , 955

A

abort, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
abs, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
AcknowledgeMessage, 900, 902
acos, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Action, 22, 38, 66, 68, 71, 76, 77, 1693
  Ajouter des informations, 66
  Cause de la non exécution d'une action, 870
  Classement, 828
  compiler, 877
  Comportement au Runtime, 890
  configurer sous VBA., 1693
  Création de fonctions et d'actions, 828, 870
  créer, 57
  créer une nouvelle action, 874
  CrossReference, 57
  Dialogue "Info", 66, 68
  Différences entre actions et fonctions, 870
  Distinction Fonction - Action, 828
  Editer, 57, 875, 895
  Edition, 870
  enregistrer, 877
  exporter, 886
  importer, 887
  Impression, 855
  Localisation, 870
  protéger contre les modifications et l'affichage, 876
  protéger par mot de passe, 68
  renommer, 887
  Renommer, 870
  supprimer, 851
  traduire, 870
  Utilisation de DLL, 840
  Utiliser des actions étrangères au projet, 888
Action en C, 1697
  configurer avec VBA à l'occurrence d'un évènement, 1697
action globale, 836
  Caractéristiques, 836
  Utilisation, 836
Action locale, 835
  Caractéristiques, 835
  Utilisation, 835
Action VB, 1699
  configurer avec VBA à l'occurrence d'un évènement, 1699
Action:Configurer un déclenchement du type Temporisateur, 76
Action:Configurer un déclenchement du type Variable, 77
Action:créer nouvelle, 62
Action:Editer, 63
Action:Elimination des défauts, 69
Action:Enregistrer, 69
Action:Renommer, 80
Action:Supprimer, 38

Action:Trigger, 19, 71
Action:utiliser plusieurs fois, 22
Actions d'édition, 846
    Exécuter les fonctions d'édition avec la souris, 846
    Exécuter les fonctions d'édition avec le clavier, 845
Activer, 82
Activer:des actions globales dans Runtime, 82
ActiveX Control, 1663
    insérer avec VBA, 1663
AddIn, 1714
    AddIn Manager, 1714
    Chargement automatique, 1714
    Chargement manuel, 1714
    Charger (instructions), 1716
    Comportement au chargement, 1714
    créer (exemple), 1718
    créer avec VB (exemple), 1718
    Décharger, 1714
    Utilisation dans le Graphics Designer, 1714
Affichage, 849
    paramétrer différentes vues, 849
ajouter un trigger, 883
AjouterDéclenchementTemporisateur , 882
Alarm Control, 282
Alarm Logging, 2532
    créer une alarme avec VBA, 2532
    éditer une alarme avec VBA, 2532
    modifier une alarme avec VBA, 2532
    supprimer une alarme avec VBA, 2532
AlarmControl, 248
    Exemple VBS, 818
Analog Clock, 251
Aperçu avant impression
    ouvrir, 856
Application externe
    Accès à MS Excel avec VBA, 1708
Applications externes, 1707
    Accès avec VBA, 1707
    ouverture de la session (log in), 1707
asctime, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
asin, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
atan, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
atan2, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
atof, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
atoi, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

atol, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
AttachDB, 686
Attributs:GSC Runtime, 89
Autorisation
    attribuer, 885
autorisations, 885
AXC_OnBtnAlarmHidingList, 910
AXC_OnBtnArcLong, 911
AXC_OnBtnArcShort, 912
AXC_OnBtnComment, 913
AXC_OnBtnEmergAckn, 914
AXC_OnBtnHideDlg, 915
AXC_OnBtnHideUnhideMsg, 915
AXC_OnBtnHit, 916
AXC_OnBtnHornAckn , 917
AXC_OnBtnInfo, 918
AXC_OnBtnLock, 919
AXC_OnBtnLockUnlock, 920
AXC_OnBtnLockWin, 920
AXC_OnBtnLoop, 921
AXC_OnBtnMsgFirst, 922
AXC_OnBtnMsgLast, 923
AXC_OnBtnMsgNext, 924
AXC_OnBtnMsgPrev, 925
AXC_OnBtnMsgWin, 925
AXC_OnBtnPrint, 926
AXC_OnBtnProtocol, 927
AXC_OnBtnScroll, 928
AXC_OnBtnSelect, 929
AXC_OnBtnSinglAckn, 929
AXC_OnBtnSortDlg, 930
AXC_OnBtnTimeBase, 931
AXC_OnBtnVisibleAckn, 932
AXC_SetFilter, 900

B

Barre des menus, 842
Barre d'état, 842
Barre d'outils, 1620, 1647
    Affecter un macro VBA, 1636
    Affecter un texte d'aide, 1633
    Affecter un texte d'état, 1633
    Ajouter un icône d'outil :, 1630
    configurer, 1619
    créer, 1629
    personnalisé, 1620
    Placement, 1620
    Propriétés, 1620
    spécifique à une application, 1620
    spécifique à une vue, 1620, 1647
Barre d'outils personnalisée, 1612, 1620
    Affecter un macro VBA, 1636
    Affecter un texte d'aide, 1633
    Affecter un texte d'état, 1633
    Ajouter un icône d'outil :, 1630
    configurer, 1619
    créer, 1629
    Placement, 1620
    Propriétés, 1620
Barre d'outils spécifique à une application, 1620
    Affecter un macro VBA, 1636
    Affecter un texte d'aide, 1633
    Affecter un texte d'état, 1633
    Ajouter un icône d'outil :, 1630
    configurer, 1619
    créer, 1629
Barre d'outils spécifique à une vue, 1648
    Affecter un macro VBA, 1636
    Affecter un texte d'état, 1633
    configurer, 1619
    créer, 1648
Barre d'outils:du Diagnostic GSC, 86
Bibliothèque des icônes
    Accès avec VBA, 1640
Bibliothèque des modules, 1640
    Accès avec VBA, 1640
    Copier un objet avec VBA, 1643
    Créer un dossier avec VBA, 1643
    éditer avec VBA, 1643
    Insérer un objet dans une vue avec VBA, 1645
    Supprimer un dossier avec VBA, 1643
Bibliothèque des outils, 1640
    Copier un objet avec VBA, 1643
    Créer un dossier avec VBA, 1643
    éditer avec VBA, 1643
    Insérer un objet dans une vue avec VBA, 1645
    Supprimer un dossier avec VBA, 1643
Breakpoint, 103, 105
Breakpoint:Définir dans le débogueur, 103
Breakpoint:Supprimer, 105

bsearch, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

C

c_bib, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
CalculateStatistic, 686
calloc, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Caractéristique, 870
ceil, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
char_io, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Classe, 2479
    HMIGO, 2479
Classe HMIGO, 2479
    Traitement des erreurs, 2479
    Utilisation, 2479
clearerr, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
clock, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Code, 45, 109
Code couleurs, 844
code de fonction, 861
    écrire, 861
Code VBA, 1611
    exporter, 1611
    global, 1607
    importer, 1611
    Ordre d'exécution, 1607
    Organisation dans le projet WinCC, 1607
    Protection par mot de passe, 1607
    protéger, 1607
    Références, 1607
    spécifique à une vue, 1607
    spécifique au projet, 1607
Code:d'une procédure, 45
Code:Imprimer, 109

Codes d'erreur, 432
Collections, 1724
Liste récapitulative, 1724
Comportement du système, 870
Configuration, 1617, 1694, 1697
Action en C avec VBA, 1697
Action VB avec VBA, 1699
Déclencheurs avec VBA, 1701
Liaison directe, 1694
pour plusieurs langues avec VBA, 1617
Configuration d'actions, 1693
sous VBA., 1693
Contenu, 11
Control Online Trend, 288
Contrôles
WinCC Alarm Control, 282
WinCC MediaControl, 260
Contrôles : WinCC FunctionTrendControl, 254
Contrôles : WinCC OnlineTableControl, 260, 264
Contrôles : WinCC RulerControl, 272
Contrôles : WinCC UserArchiveControl, 279
Controls:HMI Symbol Library, 246
Controls:WinCC Digital Analog Clock, 251
Controls:WinCC Function Trend Control, 284
Controls:WinCC Online Table Control, 286
Controls:WinCC Online Trend Control, 288
Controls:WinCC Slider Control,WinCC:WinCC Slider Control, 275
CopyRows, 686
cos, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
cosh, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
CreateTagSet, 687
Créer, 57, 62
Action, 57
Procédures, 40
Créer:Action, 62
Créer:Procédure, 44
CrossReference, 71
ctime, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
ctype, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
CutRows, 688

D

Debugger, 91, 93, 97, 99, 101, 102, 103, 105, 106, 107
activer dans WinCC, 91
Démarrage pour Global Script, 91
démarrage pour Graphics Runtime, 91
ouverture après message d'erreur, 91
ouverture automatique, 91
Debugger,Microsoft Script Debugger:Composants, 95
Debugger:Activer des signets, 105
Debugger:Call Stack, 95
Debugger:Changement de vue lors du débogage, 93
Debugger:Command Window, 95
Debugger:Composants, 95
Debugger:Définir des points d'arrêt, 103
Debugger:Déterminer les valeurs des propriétés, 106
Debugger:Déterminer les valeurs des variables, 106
Debugger:Exécuter les instructions Script, 107
Debugger:Modifier les valeurs des propriétés, 106
Debugger:Modifier les valeurs des variables, 106
Debugger:Noms des actions dans le fichier Script, 99
Debugger:Principes de base, 93, 102

Debugger:Running Documents, 95
Debugger:Sauter sur des signets, 105
Debugger:Sélectionner le script en cours, 101
Debugger:Structure des fichiers Script, 97
Debugger:Supprimer des points d'arrêt, 105
Debugger:Supprimer des signets, 105
Debugger:Traiter les Scripts étape par étape, 102
Déclenché par événement, 71
Déclenchement, 76, 77, 1701
    configurer sous VBA., 1701
    Cycle fenêtre, 1702
    Cycle standard, 1702
    Cycle vue, 1702
    les effets sur une action, 879
    modifier, 884
    supprimer, 884
    Variable, 1702
Déclenchement d'animation, 75
Déclenchement:Ajouter, de type "Variable", 77
Déclenchement:Ajouter, du type Temporisateur, 76
Déclenchement:Configurer un déclenchement pour une action, 76
Défaut:syntaxique, 69
DeleteRows, 689
DeleteTextLanguage, 2525
Démarrer, 1612
    Editeur VBA, 1607
    Macros VBA, 1612
DetachDB, 689
Diagnostic, 84, 85, 87, 88, 89, 90
Diagnostic GSC, 86
    Barre d'outils,
    placer dans une vue de process, 897
Diagnostic GSC:Attributs, 85
Diagnostic GSC:Barre d'outils, 86
Diagnostic GSC:Insérer dans une vue, 88
Diagnostic:Attributs de GSC Runtime, 89
Diagnostic:Attributs:Diagnostic GSC, 85
Diagnostic:Debugger, 90
Diagnostic:Diagnostic GSC, 84
Diagnostic:GSC Runtime, 87
Diagnostic:Insérer dans une vue GSC Runtime, 88
Diagnostic:Insérer dans une vue une fenêtre de Diagnostic GSC, 85
Diagnostic:VBS:Diagnostic, 83
Dialogue Dynamique, 1685
    Dynamiser une propriété avec VBA, 1685
difftime, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Digital Analog Clock, 251
Digital Clock, 251
Directio, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
div, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Documentation du projet
    Impression, 856
Domaine de validité, 838
Dynamisation, 1682
    de propriétés avec VBA, 1682
    Dialogue dynamique avec VBA, 1682
    Liaison de variable avec VBA, 1682
    scripts avec VBA, 1682
dynamiser
    Dynamisation d'une propriété par un script VB avec VBA, 1691
Dynamiser, 1684, 1688
    Dynamisation d'une propriété par un script C avec VBA, 1688
    Propriété avec le Dialogue dynamique avec VBA, 1685
    Propriété avec liaison de variable, 1684

E

Edit, 690
Editer, 63
Editer:Action, 63
Editeur, 28, 29, 35
Editeur VBA, 1607
    Démarrer, 1607
Editeur:Barres d'outils, 35
Editeur:Editeur VBS dans Graphics Designer, 28
Editeur:Global Script, 28, 29
Edition, 57, 1650
    Action, 57
    Edition d'une copie de vue avec VBA, 1651
    Niveaux avec VBA, 1650
    Procédures, 40
Edition d'objets avec VBA, 1653
Editor Global Script, 842
    Structure, 842
Enregistrer, 69
    Module, 53
    Procédure, 53
Enregistrer sous, 886
Enregistrer sous...
    utiliser, 850
Enregistrer:Action, 69
Entête, 852
    générer, 852
Erreurs logiques, 93
Error, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Evénement, 1704
    spécifique à l'application, 1704
    spécifique à une vue, 1704
    Transmission, 1704
Evénements, 1724
    Activated, 1730
    BeforeClose, 1731
    BeforeDocumentClose, 1732
    BeforeDocumentSave, 1733
    BeforeHMIObjectDelete, 1734
    BeforeLibraryFolderDelete, 1735
    BeforeLibraryObjectDelete, 1737
    BeforeQuit, 1738
    BeforeSave, 1739
    BeforeVisibleFalse, 1739
    ConnectionEvent, 1740
    DataLanguageChanged, 1741
    DesktopLanguageChanged, 1742
    DocumentActivated, 1743
    DocumentCreated, 1744
    DocumentOpened, 1745
    DocumentPropertyChanged, 1747
    DocumentSaved, 1746
    Gestion des événements VBA, 1724
    HMIObjectAdded, 1748
    HMIObjectMoved, 1749
    HMIObjectPropertyChanged, 1749
    HMIObjectResized, 1750
    LibraryFolderRenamed, 1751
    LibraryObjectAdded, 1754
    LibraryObjectRenamed, 1752
    MenuItemClicked, 1755
    NewLibraryFolder, 1756
    NewLibraryObject, 1757
    Opened, 1758
    Saved, 1759
    SelectionChanged, 1759
    Started, 1760
    ToolbarItemClicked, 1761
    ViewCreated, 1763
    voir référence : Evénements , 1724
    WindowStateChange, 1764
Evénements VBA
    voir VBAEvents : Gestion des événements VBA , 1724
Exécution, 1612
    Macros VBA, 1612
Exemple d'ordre général:Appeler les interfaces MS Automation, 822
Exemple pour VBS, 792
Exemple sous WinCC:Lire les valeurs variable, 799
Exemples, 792
Exemples dans WinCC, 793, 794, 795, 796, 797, 799, 802, 805
    Écrire les propriétés de l'objet, 802
    VBS sous WinCC, 792
Exemples d'ordre général:démarrer une application tierce, 824
Exemples d'ordre général:Intégration de la base de données avec VBS, 820
Exemples sous WinCC:Accéder à des objets dans Graphics Designer, 793
Exemples sous WinCC:Configurer l'affichage d'un diagnostic via Trace, 796

Exemples sous WinCC:Configurer un changement de langue, 794
Exemples sous WinCC:Configurer un changement de vue global, 795
Exemples sous WinCC:Configurer un changement via Property, 796
Exemples sous WinCC:Démarrer une action sur le serveur (objet Logging), 805
Exemples sous WinCC:Désactiver Runtime, 795
Exemples sous WinCC:Déterminer la couleur des objets, 794
Exemples, généralités, 819, 820, 822, 824
exit, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
ExitWinCC, 1507
exp, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Export, 690

É

Écrire les valeurs variable, 797

F

fabs, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fclose, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Fenêtre, 84, 87
Fenêtre de navigation
    Identification des actions dans la fenêtre de navigation, 879
Fenêtre de sortie, 842
Fenêtre d'édition, 32
    travailler dans la fenêtre d'édition, 844
Fenêtre d'édition:Codification des couleurs, 32
Fenêtre d'édition:Global Script, 32
feof, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
ferror, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fflush, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

fgetc, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fgetpos, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fgets, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Fichier, 854
    Recherche, 854
Fichier Script, 97
Fichier Script:Activer des signets, 105
Fichier Script:Longueur admise pour le nom de l'action, 99
Fichier Script:Nom des actions, 99
Fichier Script:Ouvrir dans Debugger, 101
Fichiers Script des VBScripts, 97
Fichiers VBScript, 97
Fichiers VBScript:Structure, 97
File, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
File_pos, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
FillDiagnoseInTags, 1510
floor, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fmod, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Fonction
    Différences entre actions et fonctions, 870
Fonction
    Classement, 828
    compiler, 853, 857, 865
    créer, 828, 857
    créer une nouvelle fonction, 860
    Distinction Fonction - Action, 828
    Editer, 857
    enregistrer, 865
    Impression, 855
    Localisation, 857
    protéger contre les modifications et l'affichage, 864
    renommer, 867
    Utilisation, 857
    Utilisation de DLL, 840
    Utiliser des fonctions d'autres sources, 868
    utiliser une fonction interne, 862
Fonction de projet, 831
    Caractéristiques, 831

supprimer, 851
Utilisation, 831
utiliser, 863
Fonction interne, 834
Caractéristiques, 834
Utilisation, 834
Fonction standard
Caractéristiques, 832
supprimer, 851
Utilisation, 832
utiliser, 863
Fonctions, 2489
CloseSingleAlarm (VBA), 2536
CloseTag (VBA), 2484
CloseTlgArchive (VBA), 2497
CloseTlgTag (VBA), 2498
CommitSingleAlarm (VBA), 2537
CommitTag (VBA), 2485
CommitTlgArchive (VBA), 2499
CommitTlgTag (VBA), 2501
CreateSingleAlarm (VBA), 2538
CreateTag (VBA), 2486
CreateText (VBA), 2522
CreateTextLanguage (VBA), 2520
CreateTlgArchive (VBA), 2503
CreateTlgTag (VBA), 2505
DeleteSingleAlarm (VBA), 2542
DeleteTag (VBA), 2488
DeleteText (VBA), 2523
DeleteTlgArchive (VBA), 2510
DeleteTlgTag (VBA), 2511
GetSingleAlarm (VBA), 2543
GetTag (VBA), 2489
GetText (VBA), 2526
GetTextID (VBA), 2527
GetTlgArchive (VBA), 2513
GetTlgTag (VBA), 2514
ListSingleAlarm (VBA), 2544
ListTag (VBA), 2490
ListText (VBA), 2529
ListTlgArchive (VBA), 2516
ListTlgTag (VBA), 2518
ModifyText (VBA), 2531
fopen, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fputc, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fputs, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fread, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
free, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

freopen, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
frexp, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fseek, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
fsetpos, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
ftell, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Function Trend Control, 284
FunctionTrendControl, 254
fwrite, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

G

Gauge Control, 258
GCreateMyOperationMsg, 902
GCS Runtime, 87
GCS Runtime:Attributs, 89
Gestion des événements, 1612, 1704, 1724
    déconnecter, 1704
    Evénement Activated, 1730
    interdire, 1704
    Liste récapitulative, 1724
    Transmission des événements, 1607
Gestion des événements VBAEvents , 1724
Get_Focus, 1098
GetActualPointLeft, 1107
GetActualPointTop, 1107
GetAdaptBorder, 1163
GetAdaptPicture, 1164
GetAdaptSize, 1165
GetAlarmHigh, 1131
GetAlarmLow, 1132
GetAlignment, 1064
GetAlignmentLeft, 1099
GetAlignmentTop, 1100
GetAssignments, 1121
GetAssumeOnExit, 1122
GetAssumeOnFull, 1122
GetAverage, 1165
GetAxisSection, 1065
GetBackBorderWidth, 1215
GetBackColor, 1072
GetBackColor2, 1073
GetBackColor3, 1074
GetBackColorBottom, 1074
GetBackColorTop, 1075
GetBackFlashColorOff, 1090
GetBackFlashColorOn, 1090
GetBasePicReferenced, 1208
GetBasePicTransColor, 1209
GetBasePicture, 1209
GetBasePicUseTransColor, 1210

GetBitNumber, 1123
GetBorderBackColor, 1076
GetBorderColor, 1076
GetBorderColorBottom, 1077
GetBorderColorTop, 1078
GetBorderEndStyle, 1216
GetBorderFlashColorOff, 1091
GetBorderFlashColorOn, 1092
GetBorderStyle, 1217
GetBorderWidth, 1217
GetBoxAlignment, 1218
GetBoxCount, 1108
GetBoxType, 1166
GetButtonColor, 1078
GetCaption, 1167
GetCheckAlarmHigh, 1133
GetCheckAlarmLow, 1133
GetCheckLimitHigh4, 1134
GetCheckLimitHigh5, 1135
GetCheckLimitLow4, 1136
GetCheckLimitLow5, 1136
GetCheckToleranceHigh, 1137
GetCheckToleranceLow, 1138
GetCheckWarningHigh, 1139
GetCheckWarningLow, 1139
GetClearOnError, 1124
GetClearOnNew, 1124
GetCloseButton, 1168
GetColorAlarmHigh, 1140
GetColorAlarmLow, 1141
GetColorBottom, 1079
GetColorChangeType, 1168
GetColorLimitHigh4, 1142
GetColorLimitHigh5, 1142
GetColorLimitLow4, 1143
GetColorLimitLow5, 1144
GetColorToleranceHigh, 1144
GetColorToleranceLow, 1145
GetColorTop, 1080
GetColorWarningHigh, 1146
GetColorWarningLow, 1147
GetColumn, 690, 701, 709, 710, 767, 768, 782, 783
GetColumnCollection, 691
GetCursorControl, 1169
GetCursorMode, 1170
GetDataFormat, 1125
GetDirection, 1108
GetEditAtOnce, 1170
GetEndAngle, 1109
getenv, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
GetExponent, 1065
GetExtendedOperation, 1171
GetFillColor, 1080
GetFilling, 1088
GetFillingIndex, 1089
GetFillStyle, 1219
GetFillStyle2, 1219
GetFlashBackColor, 1092
GetFlashBorderColor, 1093
GetFlashFlashPicture, 1211
GetFlashForeColor, 1094
GetFlashPicReferenced, 1211
GetFlashPicTransColor, 1212
GetFlashPicture, 1213
GetFlashPicUseTransColor, 1213
GetFlashRateBackColor, 1095
GetFlashRateBorderColor, 1095
GetFlashRateFlashPic, 1214
GetFlashRateForeColor, 1096
GetFontBold, 1100
GetFontItalic, 1101
GetFontName, 1102
GetFontSize, 1103
GetFontUnderline, 1103
GetForeColor, 1081
GetForeFlashColorOff, 1097
GetForeFlashColorOn, 1097
GetGrid, 1110
GetGridColor, 1082
GetGridHeight, 1110
GetGridWidth, 1111
GetHeight, 1111
GetHiddenInput, 1126
GetHitlistColumn , 693
GetHitlisteColumnCollection, 694
GetHotkey, 1172
GetHWDiag, 971
GetHWDiagLevel, 972
GetHysteresis, 1172
GetHysteresisRange, 1173
GetIndex, 1215
GetInputValueChar, 1126

GetInputValueDouble, 1127
GetItemBorderBackColor, 1082
GetItemBorderColor, 1083
GetItemBorderStyle, 1220
GetItemBorderWidth, 1221
GetKopFupAwl, 973
GetKopFupAwlLevel, 974
GetLanguage, 1507
GetLanguageSwitch, 1174
GetLastChange, 1174
GetLayer, 1106
GetLeft, 1112
GetLeftComma, 1066
GetLimitHigh4, 1147
GetLimitHigh5, 1148
GetLimitLow4, 1148
GetLimitLow5, 1149
GetLimitMax, 1150
GetLimitMin, 1150
GetLink, 1162
GetLinkedVariable, 905
GetListType, 1127
GetLocalPicture, 906
GetLongStrokesBold, 1067
GetLongStrokesOnly, 1067
GetLongStrokesSize, 1068
GetLongStrokesTextEach, 1069
GetMarker, 1151
GetMax, 1175
GetMaximizeButton, 1176
GetMessageBlock , 695
GetMessageBlockCollection , 696
GetMessageColumn , 697
GetMessageColumnCollection , 698, 702
GetMin, 1176
GetMoveable, 1177
GetNumberLines, 1128
GetOffsetLeft, 1178
GetOffsetTop, 1178
GetOnTop, 1179
GetOperation, 1179
GetOperationMessage, 1180
GetOperationReport, 1181
GetOperatorMessage , 699
GetOperatorMessageCollection , 700
GetOrientation, 1104
GetOutputFormat, 1129
GetOutputValueChar, 1130
GetOutputValueDouble, 1131
GetParentPicture, 907
GetParentPictureWindow, 908
GetPasswordLevel, 1182
GetPicDeactReferenced, 1194
GetPicDeactTransparent, 1195
GetPicDeactUseTransColor, 1195
GetPicDownReferenced, 1196
GetPicDownTransparent, 1197
GetPicDownUseTransColor, 1197
GetPicReferenced, 1198
GetPicTransColor, 1199
GetPictureDeactivated, 1199
GetPictureDown, 1200
GetPictureName, 1182
GetPictureUp, 1201
GetPicUpReferenced, 1202
GetPicUpTransparent, 1203
GetPicUpUseTransColor, 1203
GetPicUseTransColor, 1204
GetPointCount, 1113
GetPosition, 1192
GetPressed, 1221
GetProcess, 1183
GetPropBOOL, 1205
GetPropChar, 1206
GetPropDouble, 1207
GetPropWord, 1207
GetRadius, 1113
GetRadiusHeight, 1114
GetRadiusWidth, 1114
GetRangeMax, 1193
GetRangeMin, 1193
GetReferenceRotationLeft, 1115
GetReferenceRotationTop, 1116
GetRightComma, 1069
GetRotationAngle, 1116
GetRoundCornerHeight, 1117
GetRoundCornerWidth, 1118
GetRulerBlock , 703
GetRulerBlockCollection , 704
GetRulerColumn , 706
GetRulerColumnCollection , 707
GetRulerData, 708
GetScaleColor, 1084
GetScaleTicks, 1070
GetScaling, 1070
GetScalingType, 1071
GetScrollBars, 1184
GetSelBGColor, 1084
GetSelTextColor, 1085
GetServerName, 1185
GetServerTagPrefix, 1511
GetSizeable, 1185
GetSmallChange, 1186
GetStartAngle, 1118

GetStatisticAreaColumn, 711
GetStatisticAreaColumnCollection, 712
GetStatisticResultColumn, 713
GetStatisticResultColumnCollection, 714
GetStatusbarElement, 715
GetStatusbarElementCollection, 716
GetTagBit, 1452
GetTagBitState, 1408
GetTagBitStateQC, 1430
GetTagBitStateQCWait, 1417
GetTagBitStateWait, 1398
GetTagBitWait, 1442
GetTagByte, 1452
GetTagByteState, 1409
GetTagByteStateQC, 1431
GetTagByteStateQCWait, 1418
GetTagByteStateWait, 1399
GetTagByteWait, 1443
GetTagChar, 1453
GetTagCharState, 1410
GetTagCharStateQC, 1432
GetTagCharStateQCWait, 1419
GetTagCharStateWait, 1400
GetTagCharWait, 1443
GetTagDouble, 1454
GetTagDoubleState, 1411
GetTagDoubleStateQC, 1433
GetTagDoubleStateQCWait, 1420
GetTagDoubleStateWait, 1401
GetTagDoubleWait, 1444
GetTagDWord, 1454
GetTagDWordState, 1411
GetTagDWordStateQC, 1434
GetTagDWordStateQCWait, 1421
GetTagDWordStateWait, 1401
GetTagDWordWait, 1445
GetTagFloat, 1455
GetTagFloatState, 1412
GetTagFloatStateQC, 1435
GetTagFloatStateQCWait, 1422
GetTagFloatStateWait, 1402
GetTagFloatWait, 1445
GetTagMultiStateQCWait, 1423
GetTagMultiStateWait, 1403
GetTagMultiWait, 1446
GetTagPrefix, 1187
GetTagRaw, 1455
GetTagRawState, 1413
GetTagRawStateQC, 1436
GetTagRawStateQCWait, 1424
GetTagRawStateWait, 1404
GetTagRawWait, 1447
GetTagSByte, 1456
GetTagSByteState, 1414
GetTagSByteStateQC, 1437
GetTagSByteStateQCWait, 1425
GetTagSByteStateWait, 1405
GetTagSByteWait, 1448
GetTagSDWord, 1457
GetTagSDWordState, 1415
GetTagSDWordStateQC, 1438
GetTagSDWordStateQCWait, 1426
GetTagSDWordStateWait, 1406
GetTagSDWordWait, 1449
GetTagSWord, 1457
GetTagSWordState, 1415
GetTagSWordStateQC, 1439
GetTagSWordStateQCWait, 1427
GetTagSWordStateWait, 1407
GetTagSWordWait, 1449
GetTagValue, 1458
GetTagValueStateQC, 1440
GetTagValueStateQCWait, 1428
GetTagValueWait, 1450
GetTagWord, 1459
GetTagWordState, 1416
GetTagWordStateQC, 1441
GetTagWordStateQCWait, 1429
GetTagWordStateWait, 1407
GetTagWordWait, 1451
GetText, 1105
GetTimeAxis, 717
GetTimeAxisCollection, 719
GetTimeColumn, 720
GetTimeColumnCollection, 721
GetToggle, 1222
GetToleranceHigh, 1152
GetToleranceLow, 1152
GetToolbarButton, 722
GetToolbarButtonCollection, 723
GetTop, 1119
GetTrend, 724, 1187
GetTrendCollection, 725
GetTrendColor, 1086
GetTrendWindow, 727
GetTrendWindowCollection, 728
GetTypeAlarmHigh, 1153
GetTypeAlarmLow, 1154
GetTypeLimitHigh4, 1154
GetTypeLimitHigh5, 1155
GetTypeLimitLow4, 1156
GetTypeLimitLow5, 1157
GetTypeToleranceHigh, 1157
GetTypeToleranceLow, 1158

GetTypeWarningHigh, 1159
GetTypeWarningLow, 1160
GetUnselBGColor, 1086
GetUnselTextColor, 1087
GetUpdateCycle, 1188
GetValueAxis, 729
GetValueAxisCollection, 730
GetValueColumn, 731
GetValueColumnCollection, 732
GetVisible, 1189
GetWarningHigh, 1160
GetWarningLow, 1161
GetWidth, 1119
GetWindowBorder, 1190
GetWindowsStyle, 1223
GetXAxis, 734
GetXAxisCollection, 735
GetYAxis, 736
GetYAxisCollection, 737
GetZeroPoint, 1120
GetZeroPointValue, 1190
GetZoom, 1191
Global Script, 842
Accès aux fonctions, 857
ajouter des informations supplémentaires à la fonction, 864
Ajouter des informations supplémentaires à l'action, 876
ajouter un nouveau déclenchement du type déclenchement:nouveauDéclenchementTypeGlobal Script:AjouterDéclenchement, 882
attribuer autorisation, 885
autorisations, 885
Caractéristique, 870
Caractéristiques d'actions globales, 836
Caractéristiques d'actions locales, 835
Caractéristiques de fonctions de projet, 831
Caractéristiques de fonctions internes, 834
Caractéristiques de fonctions standard, 832
Cause de la non exécution d'une action, 870
Ce qui distingue les actions, 828
Ce qui distingue les fonctions, 828
Classement des actions, 828
Classement des fonctions, 828
Code couleurs, 844
compilation de l'action, 870, 877
Compiler une fonction, 853, 857, 865
Comportement des actions au Runtime, 890
Comportement du système, 870
Conception de l'éditeur Global Script, 842
Configurer l'impression, 855
Création des actions, 828, 870
Création des fonctions, 828
créer une fonction interne, 860
créer une nouvelle action, 874
créer une nouvelle fonction, 860
Définition de variables globales C, 838
Différences entre actions et fonctions, 870
Domaine de validité, 838
Ecrire code de fonction, 861
Editer une action, 870, 875, 895
Editer une fonction, 857
Enregistrer l'action, 877
Enregistrer sous, 886
enregistrer une fonction, 865
Exécuter les fonctions d'édition avec la souris, 846
Exécuter les fonctions d'édition avec le clavier, 846
Exporter action, 886
Générer l'en-tête, 852
GSC-Runtime, 891
Identification des actions dans la fenêtre de navigation, 879
Importer action, 887
Imprimer la documentation de projet, 856
Imprimer une action, 855
Imprimer une fonction, 855
Intégrer Global Script Runtime dans la liste de démarrage du projet, 837
La barre d'outils de Diagnostic GSC,
Les attributs de GSC Runtime, 895
Modifier un déclenchement, 884
nouveau déclenchement du type déclenchement:nouveauDéclenchementTypeVariableGlobalScript:AjouterTrigger, 883
Outil de développement, 828
Ouvrir, 887
Ouvrir aperçu avant impression, 856
paramétrer différentes vues, 850
Paramétrer style de police, 850
Placer Diagnostic GSC dans une vue de process, 897
Placer GSC-Runtime dans une vue de process, 894
Propriétés, 864, 876, 884
Propriétés Ordinateur, 837
protéger une action contre les modifications et l'affichage, 876
Protéger une fonction contre les modifications et l'affichage, 864
Quels sont les effets du déclenchement sur une action ?, 879
Recherche de fichier, 854

Recherche des actions, 870
Règle d'écriture du code de WinCC, 873
Renommer fonction, 867
Renommer l'action, 870, 887
Saisie du mot de passe, 864, 876
Supprimer des fonctions de projet, 851
Supprimer des fonctions standard, 851
Supprimer un Trigger, 884
Supprimer une action, 851
Symbole des actions, 870
travailler avec les barres d'outils, 847
travailler dans la fenêtre d'édition, 844
Type de déclenchement, 828
Utilisation d'actions locales, 835
Utilisation de DLL dans les actions, 840
Utilisation de DLL dans les fonctions, 840
Utilisation de fonctions, 857
Utilisation de variables C globales, 838
Utilisation des actions globales, 836
Utilisation des fonctions de projet, 831
Utilisation des fonctions internes, 834
Utilisation des fonctions standard, 832
Utiliser des actions étrangères au projet, 888
Utiliser des fonctions d'autres sources, 868
utiliser des fonctions internes, 862
Utiliser Enregistrer sous..., 850
Utiliser les fonctions de projet, 863
Utiliser les fonctions standard, 863
Global Script Runtime, 837
ajouter dans la liste de démarrage du projet, 837
Global Script:Barres d'outils, 35
Global Script:Structure, 29
Global Script:supprimer des actions et des procédures, 38
Global Script:Travailler dans la fenêtre d'édition, 32
GMsgFunction, 903
gmtime, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Graphics Designer, 1616
@GLOBAL.PDT, 1607
@PROJECT.PDT, 1607
Accès à la bibliothèque des modules avec VBA, 1640
Accès à la bibliothèque des outils avec VBA, 1640
Accès aux objets, 1653
Adaptation avec VBA, 1614, 1616
Affecter un macro VBA à un menu personnalisé, 1636
Affecter un macro VBA à une barre d'outils personnalisée, 1636
Ajouter un icône d'outil dans une barre d'outils personnalisée, 1630
Ajouter une option au menu personnalisé, 1623
Barre d'outils personnalisée, 1620
Barre d'outils spécifique à une application, 1620
Barre d'outils spécifique à une vue, 1620
Configurer un menu personnalisé, 1619
Configurer une barre d'outils personnalisée, 1619
Création d'un menu multilingue, 1626
Créer un menu spécifique à une application, 1622
Créer une barre d'outils, 1629
Edition d'objets avec VBA, 1653
Exécution de macros VBA, 1612
Exportation d'un code VBA, 1611
Fichier de modèles, 1607
Insérer avec VBA dans une vue un objet provenant de la bibliothèque des modules, 1645
Insérer avec VBA dans une vue un objet provenant de la bibliothèque des outils, 1645
Menu personnalisé, 1620
Menu spécifique à une application, 1620
Menu spécifique à une vue, 1620
Modèle d'objet, 1724
Organisation du code VBA, 1607
GSC Runtime:Insérer dans une vue, 88
GSC-Runtime
Attributs, 895
placer dans une vue de process, 894

H

HideAlarm, 738
HitlistRelTimeFactorType, 422
HMIObjects, 1724
voir objetsHMI , 1724

I

Imprimer:VBScript, 109
Information
Ajouter des informations supplémentaires à l'action, 876

Informations, 49, 66
Ajouter à la procédure, 49
Ajouter à l'Action, 66
Ajouter au module, 49
Insérer dans une vue, 85
Insérer:Diagnostic GSC, 85
Insérer:GSC Runtime, 88
Insertion, 85, 88
Intellisense, 45, 63
isahum, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
isalpha, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
isdigit, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
isgraph, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
islower, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
isprint, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
ispunct, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
isspace, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
isupper, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

isxdigit, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

L

labs, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Language, 431
Langue de configuration, 1617
    Accès avec VBA, 1617
Langue d'interface, 1617
    Accès avec VBA, 1617
ldexp, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
ldiv, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Les attributs de Diagnostic GSC, 85
Liaison directe, 1694
    configurer sous VBA., 1694
Listes
    Actions, 1872
    AnalogResultInfo, 1875
    Application, 1876
    DataLanguages, 1903
    Documents, 1911
    Events, 1924
    FolderItems, 1929
    GroupedObjects, 1938
    HMIDefaultObjects, 1939
    HMIObjects, 1945
    LanguageFonts, 1951
    LanguageTexts, 1954
    Layers, 1956
    Liste récapitulative, 1724
    MenuItems, 1970
    Menus, 1965
    Properties, 1992
    QualityCodeStateValues, 1997
    Selection, 2010
    SymbolLibraries, 2024
    ToolbarItems, 2034
    Toolbars, 2029
    VariableStateValues, 2046
    VariableTriggers, 2049
    Views, 2052
Listes dans VBS
    Objet d'alarmes (liste), 118
    Objet Dataset (liste), 123
    Objet Layers (liste), 128
    Objet ProcessValues (listes), 131
    Objet ScreenItems (liste), 136
    Objet Screens (listes), 141
    Objet Tags (Liste), 147
    Objet TagSet (Liste), 149
localtime, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
LockAlarm, 741

log, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
log10, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
LoopInAlarm, 741
lpszPictureName, 899

M

Macros VBA, 1612
    Exécution, 1612
    Particularités lors de l'exécution, 1607
malloc, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
math, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
MediaControl, 260
memchr, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
memcmp, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
memcpy, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
memmove, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
memory, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

memset, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Menu, 1620, 1647
  Affecter un macro VBA, 1636
  Affecter un texte d'aide, 1633
  Affecter un texte d'état, 1633
  Ajouter une option de menu, 1623
  configurer, 1619
  créer, 1622
  créer des menus multilingues, 1626
  personnalisé, 1620
  Placement, 1620
  Propriétés, 1620
  spécifique à une application, 1620
  spécifique à une vue, 1620, 1647
Menu personnalisé, 1612, 1620
  Affecter un macro VBA, 1636
  Affecter un texte d'aide, 1633
  Affecter un texte d'état, 1633
  Ajouter une option de menu, 1623
  configurer, 1619
  créer, 1622
  créer des menus multilingues, 1626
  Placement, 1620
  Propriétés, 1620
Menu spécifique à une application, 1620
  Affecter un macro VBA, 1636
  Affecter un texte d'aide, 1633
  Affecter un texte d'état, 1633
  Ajouter une option de menu, 1623
  configurer, 1619
  créer, 1622
  créer des menus multilingues, 1626
Menu spécifique à une vue, 1648
  Affecter un macro VBA, 1636
  Affecter un texte d'aide, 1633
  Affecter un texte d'état, 1633
  configurer, 1619
  créer, 1648
  créer des menus multilingues, 1626
Menus et barres d'outils
  VBScript, 12
Messages d'erreur, 386, 790
Méthodes, 686, 687, 688, 689, 690, 691, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 719, 720, 721, 722, 723, 724, 725, 727, 728, 729, 730, 731, 732, 734, 735, 736, 737, 738, 741, 742, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 757, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 786, 787, 788, 789, 1724
  Activate, 1765
  Add, 1765
  Add (GroupedObjects), 1770
  Add (listage AnalogResultInfos), 1766
  Add (listage CustomToolbars), 1768
  Add (listage Documents), 1769
  Add (listage VariableTriggers), 1772
  Add (listage Views), 1773
  AddAction, 1773
  AddActiveXControl, 1774
  AddFolder, 1778
  AddFromClipboard, 1779
  AddHMIObject, 1780
  AddItem, 1782
  AddOLEObject, 1777, 1783, 1785
  AlignBottom, 1786
  AlignLeft, 1787
  AlignRight, 1788
  AlignTop, 1789
  ArrangeMinimizedWindows, 1790
  BackwardOneLevel, 1791
  BringToFront, 1792
  CascadeWindows, 1793
  CenterHorizontally, 1794
  CenterVertically, 1795
  CheckSyntax, 1796
  Close, 1797
  CloseAll, 1798
  ConvertToScript, 1799
  CopySelection, 1800
  CopyToClipboard, 1801
  CreateCustomizedObject, 1802
  CreateDynamic, 1803
  CreateGroup, 1805
  Delete, 1807
  DeleteAll, 1808
  DeleteDynamic, 1809
  Destroy, 1811
  DuplicateSelection, 1812
  EvenlySpaceHorizontally, 1813
  EvenlySpaceVertically, 1814
  Exportation, 1815
  Find, 1816
  FindByDisplayName, 1818

FlipHorizontally, 1819
FlipVertically, 1820
ForwardOneLevel, 1821
GetDeclutterObjectSize, 1806
GetItemByPath, 1822
InsertFromMenuItem, 1823
InsertMenu, 1825
InsertMenuItem, 1826
InsertSeparator, 1828
InsertSubmenu, 1829
InsertToolbarItem, 1831
IsCSLayerVisible, 1832
IsRTLayerVisible, 1833
Item, 740, 1834
ItemByLcid, 1836
Liste récapitulative, 1724
LoadDefaultConfig, 1838
MoveOneLayerDown, 1839
MoveOneLayerUp, 1840
MoveSelection, 1841
Open, 1842
PasteClipboard, 1843
PrintDocument, 1844
PrintProjectDocumentation, 1845
Remove, 1846
Rotate, 1848
SameHeight, 1849
SameWidth, 1850
SameWidthAndHeight, 1852
Save, 1853
SaveAll, 1854
SaveAs, 1854
SaveDefaultConfig, 1855
SelectAll, 1856
SendToBack, 1857
SetCSLayerVisible, 1859
SetDeclutterObjectSize, 1860
SetOpenContext, 1859
SetRTLayerVisible, 1861
ShowPropertiesDialog, 1862
ShowSymbolLibraryDialog, 1863
ShowTagDialog, 1864
TileWindowsHorizontally, 1864
TileWindowsVertically, 1865
Ungroup, 1866
Méthodes dans VBS
ActivateDynamic , 683
Create, 687
DeactivateDynamic, 688
Méthodes:Activate, 682
Méthodes:Add, 684
Méthodes:Read, 753
Méthodes:Refresh, 758
Méthodes:Remove, 758
Méthodes:RemoveAll, 762
Méthodes:Restore, 764
Méthodes:Stop, 780
Méthodes:Trace, 781
Méthodes:Write, 783
Microsoft Script Debugger, 90, 95
Mise en valeur de la syntaxe, 45, 63
mktime, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Modèle d'objet, 1724
Listes, 114
Méthodes, 680
Objets, 114
Modèle d'objet VBA, 1724
Modèle d'objet:Propriétés, 292
Modèles de code, 63
Modèles de code pour VBS, 45
modf, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Modifier, 78
Modifier un déclenchement, 78
Modifier une procédure, 45
Modifier:Déclenchement, 78
Module, 40, 44
Ajouter des informations, 49
Dialogue "Info", 49, 51
Editer, 40
enregistrer, 53
Nom, 40
Protéger par un mot de passe, 51
Module:Créer, 44
Module:Renommer, 55
Mot de passe, 51, 68
attribuer à l'action, 68

pour un module, 51
pour une procédure, 51
MoveAxis, 742
MoveRuler, 742
MoveToFirst, 744
MoveToFirstLine, 745
MoveToFirstPage, 745
MoveToLast, 746
MoveToLastLine, 746
MoveToLastPage, 746
MoveToNext, 747
MoveToNextLine, 747
MoveToNextPage, 748
MoveToPrevious, 748
MoveToPreviousLine, 748
MoveToPreviousPage, 749

N

NextColumn, 749
NextTrend, 750
Noms, 99
Noms d'objet, 99
Noms d'objet:Déterminer dans Runtime, 99
Noms:des actions dans les fichiers VBScript, 99
Nouveau, 44, 62
Nouvelle:Action, 62
Nouvelle:Procédure, 44

O

Objet AlarmLogs, 119
Objet Column, 228
Objet ComboBox, 1895
Objet complexe, 1655
éditer avec VBA, 1657
Objet d'alarme, 117
Objet d'alarmes (liste), 118
Objet DataItem, 120
Objet DataLogs, 122
Objet DataSet (Liste), 123
Objet FaceplateObject, 1926
Objet graphique, 150
Objet graphique:Propriétés, 150
Objet graphique:Types dans VBS, 150
Objet graphique:WinCC Gauge Control, 258
Objet groupe, 1668
Accès aux propriétés de l'objet, 1668
Ajouter un objet avec VBA, 1670
créer avec VBA, 1670
dissocier avec VBA, 1670
Editer avec VBA, 1668
éditer avec VBA, 1670
éditer les objets contenus, 1673
Principes de base, 1668
supprimer avec VBA, 1670
Supprimer un objet avec VBA, 1670
Objet HitlistColumn, 229
Objet HMIRuntime, 125
Objet Item, 126
Objet Layer, 127
Objet Layers (Liste), 128
Objet ListBox, 1961
Objet Logging, 129
Objet MessageBlock, 230
Objet MessageColumn, 231
Objet MultiLineEdit, 1971
Objet OperatorMessage, 232
Objet ProcessValue, 130
Objet ProcessValues (liste), 131
Objet Project, 131
Objet RulerBlock, 234
Objet RulerColumn, 234
Objet Screen, 138
Objet ScreenItem, 132
Objet ScreenItems (liste), 136
Objet Screens (liste), 141
Objet simple, 1655
éditer avec VBA, 1657
Objet SmartTags, 143
Objet StatisticAreaColumn, 235
Objet StatisticResultColumn, 236
Objet StatusbarElement, 237
Objet Tag, 144
Objet Tags (Liste), 147
Objet TagSet (Liste), 149
Objet TimeAxis, 237
Objet TimeColumn, 238
Objet ToolbarButton, 239
Objet Trend, 240
Objet TrendWindow, 242
Objet TubeArcObject, 2037
Objet TubeDoubleTeeObject, 2039
Objet TubePolyline, 2041
Objet TubeTeeObject, 2043
Objet utilisateur, 1676
Accès avec VBA, 1676
dissocier, 1677
éditer avec VBA, 1677
Propriétés, 1676
supprimer, 1677
Objet ValueAxis, 243
Objet ValueColumn, 244

Objet Windows, 1655
éditer avec VBA, 1657
Objet XAxis, 245
Objet YAxis, 245
Objets, 1724
3DBarGraph, 1867
ActiveXControl, 1873
AnalogResultInfo, 1874
ApplicationWindow, 1879
BarGraph, 1881
BinaryResultInfo, 1884
BitResultInfo, 1885
Bouton, 1886
CheckBox, 1889
Circle, 1890
CircularArc, 1893
CustomizedObject, 1900
DataLanguage, 1902
Déclenchement, 2035
DestLink, 1904
DirectConnection, 1906
Document, 1908
DynamicDialog, 1912
Ellipse, 1914
EllipseArc, 1917
EllipseSegment, 1920
Event, 1923
FolderItem , 1927
GraphicObject, 1931
Group, 1934
GroupDisplay, 1935
HMIObject, 1943
IOField, 1947
LanguageFont, 1950
LanguageText , 1953
Layer, 1955
Line, 1958
Liste récapitulative, 1724
Menu , 1964
MenuItem , 1967
objConnection, 1973
Objet ConnectorPoints (liste), 1898
OLEObject, 1975
OptionGroup, 1977
PictureWindow, 1980
PieSegment, 1983
Polygone, 1986
PolyLine, 1989
Property, 1993
QualityCodeStateValue, 1995
Rectangle, 2000
RoundButton, 2003
RoundRectangle, 2006
ScriptInfo, 2009
Slider, 2013
SourceLink, 2016
StaticText, 2017
StatusDisplay, 2020
SymbolLibrary, 2023
TextList, 2025
Toolbar , 2028
ToolbarItem , 2031
VariableStateValue, 2045
VariableTrigger , 2048
View , 2050
Objets complexes
Champ de combinaison, 199
Contrôle, 190
Instance de bloc d'affichage, 196
Objets complexes : Champ de liste, 199
Objets complexes : Texte multiligne, 200
Objets complexes:Bargraphe, 184
Objets complexes:Bargraphes tridimensionnels, 179
Objets complexes:Champ d'E/S, 193
Objets complexes:Fenêtre d'application, 183
Objets complexes:Fenêtre de vue, 188
Objets complexes:Indicateur d'état, 208
Objets complexes:Liste de textes, 206
Objets complexes:Objet graphique, 196
Objets complexes:Objet OLE, 201
Objets complexes:Signalisation groupée, 203
Objets dans VBA
ComboBox , 1895
FaceplateObject, 1926
ListBox, 1961
MultiLineEdit, 1971
TubeArcObject, 2037
TubeDoubleTeeObject, 2039
TubePolyline, 2041
TubeTeeObject, 2043
Objets dans VBS
Objet AlarmLogs, 119
Objet d'alarme, 117
Objet DataItem, 120
Objet DataLogs, 122
Objet HMIRuntime, 125
Objet Item, 126
Objet Layer, 127
Objet Logging, 129
Objet ProcessValue, 130
Objet Project, 131
Objet Screen, 138
Objet ScreenItem, 132
Objet Tag, 144

SmartTags, 143
Objets standard:Arc de cercle, 160
Objets standard:Arc d'ellipse, 154
Objets standard:Cercle, 158
Objets standard:Connecteur, 178
Objets standard:Ellipse, 151
Objets standard:Polygone, 166
Objets standard:Rectangle, 170
Objets standard:Rectangle arrondi, 173
Objets standard:Segment de cercle, 162
Objets standard:Segment d'ellipse, 156
Objets standard:Texte statique, 176
Objets standard:Trait, 164
Objets standard:Trait polygonal, 168
Objets tuyaux : Coude de tuyau, 224
Objets tuyaux : Double raccord en T, 223
Objets tuyaux : Raccord en T, 222
Objets tuyaux : Tuyau polygonal, 222
Objets Windows:Bouton rond, 217
Objets Windows:Button, 210
Objets Windows:Case à cocher, 213
Objets Windows:Case d'option, 215
Objets Windows:Slider, 219
ObjetsHMI
voir listes , 1724
ObjetsRéférence VBA
Objetsetlistes , 1724
OnDeactivateExecute, 975
OnErrorExecute, 976
OneToOneView, 750
OnlineTableControl, 260
Exemple de script en C, 1580
Exemple VBS, 816
OnlineTrendControl, 264
Exemple de script en C, 1581
Exemple VBS, 812
Exemple VBS pour la courbe de consigne, 814
Objet TimeAxis, 237
Objet ValueAxis, 243
OnTime, 977
OpenHomePicture, 1390
OpenNextPicture, 1391
OpenPicture, 909
OpenPrevPicture, 1391
OpenStoredPicture, 1392
Outil de développement, 828
Output, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Ouvrir, 887

P

Page de codes, 2387
Palettes d'icônes
travailler avec les barres d'outils, 847
Paramètres d'impression, 855
paramétrer, 855
PasteRows , 750
Plan, 1650
Commander la visibilité avec VBA, 1650
CS, 1650
éditer avec VBA, 1650
RT, 1650
Point d'arrêt, 103, 105
Points d'arrêt:Définir dans le débogueur, 103
Points d'arrêt:Supprimer, 105
Police, 2387
pow, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
PreviousColumn, 751
PreviousTrend, 751
Principes de base, 825
de VBS, 825
Print, 752
Procédure, 1612
Ajouter des informations, 49
créer, 40
Dialogue "Info", 49, 51
Editer, 40
enregistrer, 53
Exécution, 1612
Nom, 40

Protéger par un mot de passe, 51
Procédure de projet:Utiliser, 48
Procédure standard, 48
Procédure standard:Utiliser, 48
Procédure:Créer nouvelle, 44
Procédure:Ecrire un code, 45
Procédure:Lieu d'archivage dans WinCC, 22
Procédure:Procédure de projet, 48
Procédure:Procédure standard, 48
Procédure:Renommer, 55
Procédure:Supprimer, 38
Procédure:utiliser plusieurs fois, 22
ProgramExecute, 977
Properties
Liste récapitulative, 1724
voir proprietés , 1724
Propriété, 2387
Propriété CommonVBSPropertyArea, 2139
Propriété dans VBS:UserValue1, 649
Propriété dans VBS:UserValue2, 649
Propriété dans VBS:UserValue3, 649
Propriété dans VBS:UserValue4, 650
Propriété FaceplateType, 2164
Propriété SmartTag, 553
Propriétés, 864, 876, 884
Actions, 2054
ActiveDocument, 2055
ActiveLayer, 2056
ActualPointLeft, 2056
ActualPointTop, 2057
AdaptBorder, 2058
AdaptPicture, 2059
AdaptSize, 2060
AlarmHigh, 2061
AlarmLow, 2062
Alignment, 2063
AlignmentLeft, 2063
AlignmentTop, 2064
AnalogResultInfos, 2065
AngleAlpha, 2066
AngleBeta, 2067
Application, 2067
ApplicationDataPath, 2068
Assignments, 2069
AssumeOnExit, 2069
AssumeOnFull, 2070
Attribut BackColor, 2076
AutomationName, 2070
AvailableDataLanguages, 2072
Average, 2073
Axe, 2074
AxisSection, 2075
BackBorderWidth, 2076
BackColor2, 2078
BackColor3, 2079
BackColorBottom, 2080
BackColorTop, 2080
BackFlashColorOff, 2081
BackFlashColorOn, 2082
Background, 2083
BackPictureAlignment, 2084
BackPictureName, 2084
BarDepth, 2085
BarHeight, 2085
BarWidth, 2086
BasePicReferenced, 2087
BasePicTransColor, 2087
BasePicture, 2088
BasePicUseTransColor, 2089
BaseX, 2089
BaseY, 2090
BinaryResultInfo, 2091
BitNotSetValue, 2091
BitNumber, 2092
BitResultInfo, 2092
BitSetValue, 2093
Bold, 2094
BorderBackColor, 2095
BorderColor, 2096
BorderColorBottom, 2098
BorderColorTop, 2098
BorderEndStyle, 2099
BorderFlashColorOff, 2100
BorderFlashColorOn, 2102
BorderStyle, 2103
BorderWidth, 2105
BottomConnectedConnectionPointIndex, 2107
BottomConnectedObjectName, 2106
BoxAlignment, 2107
BoxCount, 2108
BoxType, 2109
Button1Width, 2109
Button2Width, 2110
Button3Width, 2111
Button4Width, 2111
ButtonColor, 2112
Caption, 2113
CaptionText, 2114
CheckAlarmHigh, 2115
CheckAlarmLow, 2115
Checked, 2116
CheckLimitHigh4, 2117
CheckLimitHigh5, 2118
CheckLimitLow4, 2119

CheckLimitLow5, 2119
CheckToleranceHigh, 2120
CheckToleranceLow, 2121
CheckWarningHigh, 2122
CheckWarningLow, 2123
ClearOnError, 2124
ClearOnNew, 2124
CloseButton, 2125
CollectValue, 2125
ColorAlarmHigh, 2126
ColorAlarmLow, 2127
ColorBottom, 2128
ColorChangeType, 2128
ColorLimitHigh4, 2129
ColorLimitHigh5, 2130
ColorLimitLow4, 2131
ColorLimitLow5, 2132
ColorToleranceHigh, 2133
ColorToleranceLow, 2134
ColorTop, 2135
ColorWarningHigh, 2135
ColorWarningLow, 2136
CommonVBSCode, 2137
Compiled, 2140
ConfigurationFileName, 2141
Count, 2141
CurrentDataLanguage, 2143
CurrentDesktopLanguage, 2144
CursorControl, 2145
CursorMode, 2145
CustomMenus, 2146
CustomToolbars, 2147
CycleName, 2147
CycleTime, 2148
CycleType, 2148
DataFormat, 2149
Déclenchement, 2377
DefaultHMIObjects, 2149
DestinationLink, 2150
Direction, 2151
DisableVBAEvents, 2152
DisplayName, 2152
DisplayOptions, 2153
DisplayText, 2154
Documents, 2155
Dynamic, 2155
EditAtOnce, 2156
ElseCase, 2157
Enabled, 2158
EndAngle, 2158
Events, 2159
EventType, 2161
Exponent, 2162
ExtendedOperation, 2163
ExtendedZoomingEnable, 2163
Family, 2189
FillColor, 2164
Filling, 2166
FillingIndex, 2167
FillStyle, 2169
FillStyle2, 2171
FillStyleAlignment, 2173
FlashBackColor, 2173
FlashBorderColor, 2174
FlashFlashPicture, 2175
FlashForeColor, 2176
FlashPicReferenced, 2177
FlashPicTransColor, 2177
FlashPicture, 2178
FlashPicUseTransColor, 2179
FlashRate, 2180
FlashRateBackColor, 2181
FlashRateBorderColor, 2182
FlashRateFlashPic, 2184
FlashRateForeColor, 2185
FolderItems, 2187
FontBold, 2188
FontColor, 2194
FontFlashColorOff, 2194
FontFlashColorOn, 2195
FontItalic, 2191
FontName, 2191
FontSize, 2192
FontUnderline, 2193
ForeColor, 2194
ForeFlashColorOff, 2194
ForeFlashColorOn, 2195
GlobalColorScheme, 2196
GlobalShadow, 2197
Grid, 2197
GridColor, 2198
GridHeight, 2198
GridWidth, 2199
GroupedHMIObjects, 2200
GroupParent, 2200
Height, 2201
HiddenInput, 2203
Hide, 2202
Hysteresis, 2205
HysteresisRange, 2205
Icon, 2206, 2207, 2209
Indice, 2207
InputValue, 2210
IsActive, 2210

IsConnectedToProject, 2211
IsDynamicable, 2211
Italic, 2212
Item, 2213
ItemBorderBackColor, 2214
ItemBorderColor, 2215
ItemBorderStyle, 2216
ItemBorderWidth, 2216
Jour, 2366
Key, 2217
label, 2218
LanguageID, 2219
LanguageName, 2220
LanguageSwitch, 2221
Layer, 2222
Layer00Checked, 2223
Layer00Color, 2224
Layer00Value, 2225
Layer01Checked, 2225
Layer01Color, 2226
Layer01Value, 2227
Layer02Checked, 2227
Layer02Color, 2228
Layer02Value, 2229
Layer03Checked, 2229
Layer03Color, 2230
Layer03Value, 2231
Layer04Checked, 2231
Layer04Color, 2232
Layer04Value, 2233
Layer05Checked, 2233
Layer05Color, 2234
Layer05Value, 2235
Layer06Checked, 2235
Layer06Color, 2236
Layer06Value, 2237
Layer07Checked, 2237
Layer07Color, 2238
Layer07Value, 2239
Layer08Checked, 2239
Layer08Color, 2240
Layer08Value, 2241
Layer09Checked, 2241
Layer09Color, 2242
Layer09Value, 2243
Layer10Checked, 2243
Layer10Color, 2244
Layer10Value, 2245
LayerDecluttering, 2245
Layers, 2246
LDFonts, 2247
LDLabelTexts, 2248
LDNames, 2249
LDStatusTexts, 2250
LDTexts, 2251
LDTooltipTexts, 2252
Left, 2253
LeftComma, 2254
LightEffect, 2254
LimitHigh4, 2255
LimitHigh5, 2256
LimitLow4, 2257
LimitLow5, 2257
LimitMax, 2258
LimitMin, 2259
LineJoinStyle, 2260
Liste récapitulative, 1724
ListType, 2260
LockBackColor, 2261
LockedByCreatorID, 2261
LockStatus, 2263
LockText, 2264
LockTextColor, 2264
LongStrokesBold, 2265
LongStrokesOnly, 2266
LongStrokesSize, 2266
LongStrokesTextEach, 2267
Macro, 2267
Max, 2269
MaximizeButton, 2270
MaxZoom, 2270
MCGUBackColorOff, 2271
MCGUBackColorOn, 2272
MCGUBackFlash, 2272
MCGUTextColorOff, 2273
MCGUTextColorOn, 2274
MCGUTextFlash, 2274
MCKOBackColorOff, 2275
MCKOBackColorOn, 2276
MCKOBackFlash, 2276
MCKOTextColorOff, 2277
MCKOTextColorOn, 2278
MCKOTextFlash, 2278
MCKQBackColorOff, 2279
MCKQBackColorOn, 2280
MCKQBackFlash, 2280
MCKQTextColorOff, 2281
MCKQTextColorOn, 2282
MCKQTextFlash, 2282
MCText, 2283
MenuItems, 2284
MenuItemType, 2285
MessageClass, 2286
Méthode DeselectAll, 1810

Min, 2286
MinZoom, 2287
Modified, 2288
Moveable, 2289
NegativeValue, 2291
Nom, 2289
Number, 2291
NumberLines, 2292
ObjectName, 2293
ObjectSizeDecluttering, 2295
OffsetLeft, 2296
OffsetTop, 2296
OnTop, 2297
Opération, 2298
OperationMessage, 2299
OperationReport, 2300
Orientation, 2301
OutputFormat, 2301
OutputValue, 2302
Parent, 2303
PasswordLevel, 2305
Path, 2306
PdlProtection, 2307
PicDeactReferenced, 2307
PicDeactTransparent, 2308
PicDeactUseTransColor, 2309
PicDownReferenced, 2309
PicDownTransparent, 2310
PicDownUseTransColor, 2311
PicReferenced, 2311
PicTransColor, 2312
PictureDeactivated, 2313
PictureDown, 2314
PictureName, 2315
PictureUp, 2315
PicUpReferenced, 2316
PicUpTransparent, 2317
PicUpUseTransColor, 2317
PicUseTransColor, 2318
PointCount, 2319
Position, 2320
PositiveValue, 2321
PredefinedAngels, 2321
Pressed, 2322
Process, 2323
ProfileName, 2323
ProgID, 2324
ProjectName, 2325
ProjectType, 2325
Properties, 2326
Propriété CommandLine, 2139
Prototype, 2327
QualityCodeStateChecked, 2328
QualityCodeStateValues, 2329
Raccourci clavier, 2204
RadiusHeight, 2331
RadiusWidth, 2331
RangeTo, 2332
Rayon, 2330
ReferenceRotationLeft, 2332
ReferenceRotationTop, 2333
Relevant, 2334
Repère, 2268
ResultType, 2335
RightComma, 2335
RotationAngle, 2336
RoundCornerHeight, 2337
RoundCornerWidth, 2338
SameSize, 2338
ScaleColor, 2339
ScaleTicks, 2340
Scaling, 2340
ScalingMode, 2341
ScalingType, 2341
ScriptType, 2342
ScrollBars, 2343
ScrollPositionX, 2344
ScrollPositionY, 2345
ScrollPosX, 2346
ScrollPosY, 2346
SelBGColor, 2347
Selected, 2347
SelIndex, 2349
SelText, 2349
SelTextColor, 2349
ServerName, 2350
ServerPrefix, 2350
ShortCut, 2351
SignificantMask, 2353
Size, 2353
Sizeable, 2354
SmallChange, 2355
SnapToGrid, 2355
SourceCode, 2357
SourceLink, 2356
StartAngle, 2359
StatusText, 2359
SubMenu, 2360
SymbolLibraries, 2361
TabOrderAllHMIObjects, 2363
TabOrderAlpha, 2363
TabOrderKeyboard, 2364
TabOrderMouse, 2365
TabOrderOtherAction, 2365

TabOrderSwitch, 2362
TagPrefix, 2367
Tendance, 2376
Texte, 2369
Toggle, 2370
ToleranceHigh, 2370
ToleranceLow, 2371
ToolbarItems, 2372
TooltipText, 2373
Top, 2374
TopConnectedObjectName, 2374
TopConnectedPointIndex, 2375
Transparency, 2375
TrendColor, 2376
Type, 2378
TypeAlarmHigh, 2379
TypeAlarmLow, 2379
TypeLimitHigh4, 2380
TypeLimitHigh5, 2381
TypeLimitLow4, 2382
TypeLimitLow5, 2383
TypeToleranceHigh, 2383
TypeToleranceLow, 2384
TypeWarningHigh, 2385
TypeWarningLow, 2386
Underlined, 2388
UnselBGColor, 2389
UnselTextColor, 2390
UpdateCycle, 2390
UserValue1, 2391
UserValue2, 2392
UserValue3, 2393
UserValue4, 2394
value, 2395
VALUE_ACCESS_FAULT, 2395
VALUE_ADDRESS_ERROR, 2397
VALUE_BAD_COMMLUV, 2399
VALUE_BAD_COMMNUV, 2400
VALUE_BAD_CONFERROR, 2402
VALUE_BAD_DEVICE, 2404
VALUE_BAD_MISCSTATES, 2406
VALUE_BAD_NONSPECIFIC, 2408
VALUE_BAD_NOTCONNECTED, 2410
VALUE_BAD_OUTOFSERV, 2412
VALUE_BAD_PROCRELNOM, 2414
VALUE_BAD_PROCRELSUB, 2416
VALUE_CONVERSION_ERROR, 2418
VALUE_HANDSHAKE_ERROR, 2420
VALUE_HARDWARE_ERROR, 2422
VALUE_HIGHLIMITED, 2423
VALUE_INVALID_KEY, 2425
VALUE_LOWLIMITED, 2427
VALUE_MAX_LIMIT, 2429
VALUE_MAX_RANGE, 2431
VALUE_MIN_LIMIT, 2433
VALUE_MIN_RANGE, 2434
VALUE_NOT_ESTABLISHED, 2436
VALUE_SERVERDOWN, 2437
VALUE_STARTUP_VALUE, 2439
VALUE_TIMEOUT, 2440
VALUE_UNCERT_ENGVHIGHLIM, 2442
VALUE_UNCERT_ENGVLOWLIM, 2443
VALUE_UNCERT_ENGVONLIM, 2445
VALUE_UNCERT_INITVAL, 2447
VALUE_UNCERT_LUV, 2449
VALUE_UNCERT_MAINTDEM, 2451
VALUE_UNCERT_MISCSTATES, 2453
VALUE_UNCERT_NONSPECIFIC, 2455
VALUE_UNCERT_PROCRELNOM, 2457
VALUE_UNCERT_SIMVAL, 2459
VALUE_UNCERT_SUBSTSET, 2461
VariablesExist, 2463
VariableStateChecked, 2464
VariableStateType, 2465
VariableStateValues, 2465
VariableTriggers, 2466
VarName, 2467
VBAVersion, 2468
VBE, 2468
Version, 2468
Views, 2469
Visible, 2469
WarningHigh, 2470
WarningLow, 2471
Width, 2472, 2473
WindowBorder, 2473
WindowMonitorNumber, 2474
WindowPositionMode, 2475
WindowsStyle, 2475
WindowState, 2476
ZeroPoint, 2476
ZeroPointValue, 2477
Zoom, 2478
Propriétés - Ordinateur, 91
Onglet Runtime, 91
Propriétés CommonVBSEventArea, 2138
Propriétés dans VBA
CommonVBSEventArea, 2138
CommonVBSPropertyArea, 2139
FaceplateType, 2164
TagScaleParam1, 2368
TagScaleParam2, 2368
TagScaleParam3, 2369
TagScaleParam4, 2369

Propriétés dans VBS, 292
AlarmID, 299
BackStyle, 317
Comment, 366
ComputerName, 367
Context, 367
FillStyle, 394
FillStyleAlignment, 396
GlobalColorScheme, 412
HiddenInput, 302, 303, 418, 456, 488, 516, 549, 648, 672
IndependentWindow, 424
Index, 425
InputValue, 426
Instance, 427
ItemProviderClsid, 429
LineJoinStyle, 453
NumberLines, 483
ProcessValue, 517
RotationAngle, 524
SavedTrend, 530
Selected Trend, 538
SelIndex, 541
SelText, 541
SmartTag, 553
SortOrder, 553
State, 560
TableFocusOnButtonCommand, 567
Transparency, 616
UserName, 648
WinCCStyle, 670
WindowPositionMode, 670
WindowsStyle, 671
Propriétés dans VBS : BackPictureAlignment, 316, 508
Propriétés dans VBS : BackPictureName, 317
Propriétés dans VBS : DisplayOptions, 381
Propriétés dans VBS/ScrollPositionX, 534
Propriétés dans VBS/ScrollPositionY, 535
Propriétés dans VBS:AccessPath, 293
Propriétés dans VBS:Activate, 294
Propriétés dans VBS:ActiveProject, 294
Propriétés dans VBS:ActiveScreen, 295
Propriétés dans VBS:ActiveScreenItem, 295
Propriétés dans VBS:Actualize, 296
Propriétés dans VBS:ActualPointLeft, 296
Propriétés dans VBS:ActualPointTop, 297
Propriétés dans VBS:AdaptBorder, 297
Propriétés dans VBS:AdaptPicture, 298
Propriétés dans VBS:AdaptSize, 298
Propriétés dans VBS:AdjustRuler, 298
Propriétés dans VBS:AlarmHigh, 299
Propriétés dans VBS:AlarmLogs, 299
Propriétés dans VBS:AlarmLow, 300
Propriétés dans VBS:Alignment, 300
Propriétés dans VBS:AlignmentLeft, 300
Propriétés dans VBS:AlignmentTop, 301
Propriétés dans VBS:AllowPersistence, 301
Propriétés dans VBS:AllServer, 302
Propriétés dans VBS:Analog, 303
Propriétés dans VBS:AngleAlpha, 303
Propriétés dans VBS:AngleBeta, 303
Propriétés dans VBS:AngleMax, 304
Propriétés dans VBS:AngleMin, 304
Propriétés dans VBS:Application, 304, 319
Propriétés dans VBS:Archive, 305
Propriétés dans VBS:Assignments, 306
Propriétés dans VBS:AssumeOnExit, 306
Propriétés dans VBS:AssumeOnFull, 307
Propriétés dans VBS:AutoRange, 308
Propriétés dans VBS:AutorangeX, 308
Propriétés dans VBS:AutorangeY, 309
Propriétés dans VBS:AutoScroll, 309
Propriétés dans VBS:AutoSize, 311
Propriétés dans VBS:Average, 311
Propriétés dans VBS:Axe, 312
Propriétés dans VBS:AxisSection, 312
Propriétés dans VBS:BackBorderWidth, 312
Propriétés dans VBS:BackColor, 313
Propriétés dans VBS:BackColor2, 314
Propriétés dans VBS:BackColor3, 314
Propriétés dans VBS:BackColorBottom, 315
Propriétés dans VBS:BackColorTop, 315
Propriétés dans VBS:BackFlashColorOff, 315
Propriétés dans VBS:BackFlashColorOn, 316
Propriétés dans VBS:Background, 316
Propriétés dans VBS:BackgroundPicture, 316
Propriétés dans VBS:BarBackColor, 318
Propriétés dans VBS:BarDepth, 318
Propriétés dans VBS:BarFillColor, 318
Propriétés dans VBS:BarHeight, 319
Propriétés dans VBS:BasePicReferenced, 319
Propriétés dans VBS:BasePicTransColor, 319
Propriétés dans VBS:BasePicture, 320
Propriétés dans VBS:BasePicUseTransColor, 320
Propriétés dans VBS:BaseScreenName, 320
Propriétés dans VBS:BaseX, 321
Propriétés dans VBS:BaseY, 321
Propriétés dans VBS:BeginTime, 322
Propriétés dans VBS:BeginValue, 322
Propriétés dans VBS:BeginX, 323
Propriétés dans VBS:BeginY, 323
Propriétés dans VBS:BevelColorDown, 323
Propriétés dans VBS:BevelColorUp, 324

Propriétés dans VBS:BevelInner, 324
Propriétés dans VBS:BevelOuter, 324
Propriétés dans VBS:BevelWidth, 325
Propriétés dans VBS:BitNumber, 325
Propriétés dans VBS:BlinkColor, 325
Propriétés dans VBS:BorderBackColor, 332
Propriétés dans VBS:BorderColor, 332
Propriétés dans VBS:BorderColorBottom, 332
Propriétés dans VBS:BorderColorTop, 333
Propriétés dans VBS:BorderEndStyle, 333
Propriétés dans VBS:BorderFlashColorOff, 333
Propriétés dans VBS:BorderFlashColorOn, 334
Propriétés dans VBS:BorderStyle, 334
Propriétés dans VBS:BorderWidth, 334
Propriétés dans VBS:BottomConnectedConnectionPointIndex, 335
Propriétés dans VBS:BottomConnectedObjectName, 335
Propriétés dans VBS:BoxAlignment, 335
Propriétés dans VBS:BoxCount, 336
Propriétés dans VBS:BoxType, 336
Propriétés dans VBS:Button1Width, 338
Propriétés dans VBS:Button2Width, 338
Propriétés dans VBS:Button3Width, 338
Propriétés dans VBS:Button4Width, 339
Propriétés dans VBS:ButtonColor, 336
Propriétés dans VBS:ButtonCommand, 337
Propriétés dans VBS:Caption, 339
Propriétés dans VBS:CaptionColor, 340
Propriétés dans VBS:CaptionFont, 340
Propriétés dans VBS:CaptionOffset, 340
Propriétés dans VBS:CaptionText, 341
Propriétés dans VBS:CellCut, 341
Propriétés dans VBS:CenterColor, 342
Propriétés dans VBS:CenterScale, 342
Propriétés dans VBS:CheckAlarmHigh, 343
Propriétés dans VBS:CheckAlarmLow, 343
Propriétés dans VBS:CheckLimitHigh4, 343, 344
Propriétés dans VBS:CheckLimitHigh5, 344
Propriétés dans VBS:CheckLimitLow5, 344
Propriétés dans VBS:CheckToleranceHigh, 345
Propriétés dans VBS:CheckToleranceLow, 345
Propriétés dans VBS:CheckWarningHigh, 345
Propriétés dans VBS:CheckWarningLow, 346
Propriétés dans VBS:ClearOnError, 346
Propriétés dans VBS:ClearOnNew, 346
Propriétés dans VBS:Closeable, 346
Propriétés dans VBS:CloseButton, 347
Propriétés dans VBS:CoarseGrid, 347
Propriétés dans VBS:CoarseGridValue, 348
Propriétés dans VBS:CoarseGridValueX, 349
Propriétés dans VBS:CoarseGridValueY, 349
Propriétés dans VBS:CoarseGridX, 348
Propriétés dans VBS:CoarseGridY, 348
Propriétés dans VBS:CollectValue, 349
Propriétés dans VBS:ColMove, 349
Propriétés dans VBS:Color, 350
Propriétés dans VBS:ColorAlarmHigh, 350
Propriétés dans VBS:ColorAlarmLow, 350
Propriétés dans VBS:ColorBottom, 351
Propriétés dans VBS:ColorChangeType, 351
Propriétés dans VBS:ColorLimitHigh4, 351
Propriétés dans VBS:ColorLimitHigh5, 352
Propriétés dans VBS:ColorLimitLow4, 352
Propriétés dans VBS:ColorLimitLow5, 352
Propriétés dans VBS:ColorToleranceHigh, 353
Propriétés dans VBS:ColorToleranceLow, 353
Propriétés dans VBS:ColorTop, 353
Propriétés dans VBS:ColorWarningHigh, 354
Propriétés dans VBS:ColorWarningLow, 354
Propriétés dans VBS:ColTitle, 354
Propriétés dans VBS:ColWidth, 365
Propriétés dans VBS:Command, 365
Propriétés dans VBS:CommonTime, 366
Propriétés dans VBS:CommonX, 366
Propriétés dans VBS:CommonY, 367
Propriétés dans VBS:ContinousChange, 368
Propriétés dans VBS:Count, 368
Propriétés dans VBS:CurrentContext, 369
Propriétés dans VBS:Cursor, 370
Propriétés dans VBS:CursorControl, 370
Propriétés dans VBS:CurveForm, 371
Propriétés dans VBS:Danger, 372
Propriétés dans VBS:DangerColor, 372, 375
Propriétés dans VBS:DataFormat, 372
Propriétés dans VBS:DataIndex, 373
Propriétés dans VBS:DataLogs, 373
Propriétés dans VBS:DataSet, 373
Propriétés dans VBS:DataX, 374
Propriétés dans VBS:DataXY, 374
Propriétés dans VBS:DataY, 375
Propriétés dans VBS:DeleteData, 377
Propriétés dans VBS:Delta, 377
Propriétés dans VBS:DesiredCurveColor, 377
Propriétés dans VBS:DesiredCurveCurveForm, 378
Propriétés dans VBS:DesiredCurveNumberOfUAValues, 378
Propriétés dans VBS:DesiredCurveSourceUAArchive, 378
Propriétés dans VBS:DesiredCurveSourceUAArchiveStartID, 379
Propriétés dans VBS:DesiredCurveSourceUAColumnX, 379

Propriétés dans VBS:DesiredCurveSourceUAColumnY, 379
Propriétés dans VBS:DesiredCurveVisible, 380
Propriétés dans VBS:Direction, 380
Propriétés dans VBS:Edit, 381
Propriétés dans VBS:Editable, 382
Propriétés dans VBS:EditAtOnce, 382
Propriétés dans VBS:Enabled, 382
Propriétés dans VBS:EndAngle, 384
Propriétés dans VBS:EndTime, 384
Propriétés dans VBS:EndValue, 385
Propriétés dans VBS:EndX, 385
Propriétés dans VBS:EndY, 385
Propriétés dans VBS:ErrorDescription, 386
Propriétés dans VBS:Exponent, 387
Propriétés dans VBS:ExtendedOperation, 390
Propriétés dans VBS:ExtendedZoomingEnable, 390
Propriétés dans VBS:FillColor, 392
Propriétés dans VBS:Filling, 393
Propriétés dans VBS:FillingIndex, 393
Propriétés dans VBS:FillStyle2, 395
Propriétés dans VBS:FineGrid, 396
Propriétés dans VBS:FineGridValue, 396
Propriétés dans VBS:FineGridValueX, 397
Propriétés dans VBS:FineGridValueY, 397
Propriétés dans VBS:FineGridX, 397
Propriétés dans VBS:FineGridY, 397
Propriétés dans VBS:FlashBackColor, 398
Propriétés dans VBS:FlashBorderColor, 398
Propriétés dans VBS:FlashFlashPicture, 398
Propriétés dans VBS:FlashForeColor, 398
Propriétés dans VBS:FlashPicReferenced, 399
Propriétés dans VBS:FlashPicTransColor, 399
Propriétés dans VBS:FlashPicture, 399
Propriétés dans VBS:FlashPicUseTransColor, 400
Propriétés dans VBS:FlashRate, 400
Propriétés dans VBS:FlashRateBackColor, 400
Propriétés dans VBS:FlashRateBorderColor, 401
Propriétés dans VBS:FlashRateFlashPic, 401
Propriétés dans VBS:FlashRateForeColor, 401
Propriétés dans VBS:Flip, 402
Propriétés dans VBS:FocusColor, 403
Propriétés dans VBS:FocusRect, 403
Propriétés dans VBS:FocusWidth, 403
Propriétés dans VBS:Font, 404
Propriétés dans VBS:FontBold, 405
Propriétés dans VBS:FontItalic, 405
Propriétés dans VBS:FontName, 406
Propriétés dans VBS:FontPosition, 406
Propriétés dans VBS:FontSize, 407
Propriétés dans VBS:FontStrikeThru, 407
Propriétés dans VBS:FontUnderline, 407
Propriétés dans VBS:ForeColor, 408
Propriétés dans VBS:ForeFlashColorOff, 409
Propriétés dans VBS:ForeFlashColorOn, 409
Propriétés dans VBS:FrameColor, 409
Propriétés dans VBS:FrameColorDown, 410
Propriétés dans VBS:FrameColorUp, 410
Propriétés dans VBS:FramePicture, 410
Propriétés dans VBS:FrameScale, 411
Propriétés dans VBS:FrameWidth, 411
Propriétés dans VBS:FreezeProviderConnections, 411
Propriétés dans VBS:GraphDirection, 412
Propriétés dans VBS:GridLineHorz, 413
Propriétés dans VBS:GridLines, 412, 414
Propriétés dans VBS:GridlinesValueX, 414
Propriétés dans VBS:GridlinesValueY, 414
Propriétés dans VBS:GridlinesX, 414
Propriétés dans VBS:GridlinesY, 415
Propriétés dans VBS:GridLineValue, 415
Propriétés dans VBS:GridLineVert, 415
Propriétés dans VBS:HandFillColor, 416
Propriétés dans VBS:Handtype, 416
Propriétés dans VBS:HeaderSort, 417
Propriétés dans VBS:Height, 417
Propriétés dans VBS:HiddenInput, 418
Propriétés dans VBS:Hotkey, 422
Propriétés dans VBS:HourNeedleHeight, 423
Propriétés dans VBS:HourNeedleWidth, 423
Propriétés dans VBS:Hysteresis, 423
Propriétés dans VBS:HysteresisRange, 424
Propriétés dans VBS:InnerBevelOffset, 425
Propriétés dans VBS:InnerBevelStyle, 426
Propriétés dans VBS:InnerBevelWidth, 426
Propriétés dans VBS:InsertData, 427
Propriétés dans VBS:ItemBorderBackColor, 427
Propriétés dans VBS:ItemBorderColor, 428
Propriétés dans VBS:ItemBorderStyle, 428
Propriétés dans VBS:ItemBorderWidth, 428
Propriétés dans VBS:ItemVisible, 429
Propriétés dans VBS:Label, 430
Propriétés dans VBS:LabelColor, 430
Propriétés dans VBS:LabelX, 430
Propriétés dans VBS:LabelY, 431
Propriétés dans VBS:Language ID, 431
Propriétés dans VBS:LanguageSwitch, 431
Propriétés dans VBS:LastError, 432
Propriétés dans VBS:Layer, 433
Propriétés dans VBS:Layer00Checked, 434
Propriétés dans VBS:Layer00Color, 437
Propriétés dans VBS:Layer00Value, 445
Propriétés dans VBS:Layer01Checked, 434
Propriétés dans VBS:Layer01Color, 437

Propriétés dans VBS:Layer01Value, 446
Propriétés dans VBS:Layer02Checked, 435
Propriétés dans VBS:Layer02Color, 438
Propriétés dans VBS:Layer02Value, 446
Propriétés dans VBS:Layer03Checked, 435
Propriétés dans VBS:Layer03Color, 438
Propriétés dans VBS:Layer03Value, 446
Propriétés dans VBS:Layer04Checked, 435
Propriétés dans VBS:Layer04Color, 438
Propriétés dans VBS:Layer04Value, 447
Propriétés dans VBS:Layer05Checked, 435
Propriétés dans VBS:Layer05Color, 439
Propriétés dans VBS:Layer05Value, 447
Propriétés dans VBS:Layer06Checked, 436
Propriétés dans VBS:Layer06Color, 439
Propriétés dans VBS:Layer06Value, 447
Propriétés dans VBS:Layer07Checked, 436
Propriétés dans VBS:Layer07Color, 439
Propriétés dans VBS:Layer07Value, 447
Propriétés dans VBS:Layer08Checked, 436
Propriétés dans VBS:Layer08Color, 440
Propriétés dans VBS:Layer08Value, 448
Propriétés dans VBS:Layer09Checked, 437
Propriétés dans VBS:Layer09Color, 440
Propriétés dans VBS:Layer09Value, 448
Propriétés dans VBS:Layer10Checked, 437
Propriétés dans VBS:Layer10Color, 440
Propriétés dans VBS:Layer10Value, 448
Propriétés dans VBS:LayerDeclutteringEnable, 449
Propriétés dans VBS:Layers, 449
Propriétés dans VBS:Left, 449
Propriétés dans VBS:LeftComma, 450
Propriétés dans VBS:LightEffect, 450
Propriétés dans VBS:LimitHigh4, 451
Propriétés dans VBS:LimitHigh5, 451
Propriétés dans VBS:LimitLow4, 451
Propriétés dans VBS:LimitLow5, 452
Propriétés dans VBS:LimitMax, 452
Propriétés dans VBS:LimitMin, 452
Propriétés dans VBS:LineFont, 453
Propriétés dans VBS:LineHeight, 453
Propriétés dans VBS:LineTitle, 454
Propriétés dans VBS:LineWidth, 454
Propriétés dans VBS:ListType, 454
Propriétés dans VBS:LoadDataImmediately, 455
Propriétés dans VBS:LocaleID, 455
Propriétés dans VBS:LockBackColor, 456
Propriétés dans VBS:LockStatus, 456
Propriétés dans VBS:LockText, 457
Propriétés dans VBS:LockTextColor, 457
Propriétés dans VBS:Logging, 457
Propriétés dans VBS:LongStrokesBold, 457
Propriétés dans VBS:LongStrokesOnly, 458
Propriétés dans VBS:LongStrokesSize, 458
Propriétés dans VBS:LongStrokesTextEach, 458
Propriétés dans VBS:LowerLimit, 459
Propriétés dans VBS:LowerLimitColor, 460
Propriétés dans VBS:LowerLimitValue, 460
Propriétés dans VBS:Marker, 461
Propriétés dans VBS:Max, 461
Propriétés dans VBS:MaximizeButton, 461
Propriétés dans VBS:MCGUBackColorOff, 462
Propriétés dans VBS:MCGUBackColorOn, 462
Propriétés dans VBS:MCGUBackFlash, 462
Propriétés dans VBS:MCGUTextColorOff, 463
Propriétés dans VBS:MCGUTextColorOn, 463
Propriétés dans VBS:MCGUTextFlash, 463
Propriétés dans VBS:MCKOBackColorOff, 463
Propriétés dans VBS:MCKOBackColorOn, 464
Propriétés dans VBS:MCKOBackFlash, 464
Propriétés dans VBS:MCKOTextColorOff, 464
Propriétés dans VBS:MCKOTextColorOn, 465
Propriétés dans VBS:MCKOTextFlash, 465
Propriétés dans VBS:MCKQBackColorOff, 465
Propriétés dans VBS:MCKQBackColorOn, 465
Propriétés dans VBS:MCKQBackFlash, 466
Propriétés dans VBS:MCKQTextColorOff, 466
Propriétés dans VBS:MCKQTextColorOn, 466
Propriétés dans VBS:MCKQTextFlash, 467
Propriétés dans VBS:MCText, 467
Propriétés dans VBS:MeasurePoints, 467
Propriétés dans VBS:MessageClass, 476
Propriétés dans VBS:Min, 479
Propriétés dans VBS:MinuteNeedleHeight, 479
Propriétés dans VBS:MinuteNeedleWidth, 480
Propriétés dans VBS:Moveable, 480
Propriétés dans VBS:MsgCtrlFlags, 481
Propriétés dans VBS:MsgFilterSQL, 481
Propriétés dans VBS:Name, 482
Propriétés dans VBS:NeedleColor, 483
Propriétés dans VBS:NormalColor, 483
Propriétés dans VBS:NumItems, 484
Propriétés dans VBS:Object, 484
Propriétés dans VBS:ObjectName, 485
Propriétés dans VBS:ObjectSizeDeclutteringEnable, 486
Propriétés dans VBS:ObjectSizeDeclutteringMax, 486
Propriétés dans VBS:ObjectSizeDeclutteringMin, 487
Propriétés dans VBS:OffsetLeft, 487
Propriétés dans VBS:OffsetTop, 488
Propriétés dans VBS:Online, 489
Propriétés dans VBS:OnTop, 489

Propriétés dans VBS:OperationMessage, 490
Propriétés dans VBS:OperationReport, 498
Propriétés dans VBS:Orientation, 499
Propriétés dans VBS:OuterBevelStyle, 499
Propriétés dans VBS:OuterBevelWidth, 500
Propriétés dans VBS:Outline, 500
Propriétés dans VBS:OutputFormat, 500
Propriétés dans VBS:OutputValue, 500
Propriétés dans VBS:Parent, 501
Propriétés dans VBS:PasswordLevel, 503
Propriétés dans VBS:Path, 504
Propriétés dans VBS:PersistentRT, 505
Propriétés dans VBS:PersistentRTCS, 505
Propriétés dans VBS:PersistentRTCSPermission, 506
Propriétés dans VBS:PersistentRTPermission, 506
Propriétés dans VBS:PicDeactReferenced, 506
Propriétés dans VBS:PicDeactTransparent, 507
Propriétés dans VBS:PicDeactUseTransColor, 507
Propriétés dans VBS:PicDownReferenced, 507
Propriétés dans VBS:PicDownTransparent, 508
Propriétés dans VBS:PicDownUseTransColor, 508
Propriétés dans VBS:PicReferenced, 508
Propriétés dans VBS:PicTransColor, 509
Propriétés dans VBS:Picture, 509
Propriétés dans VBS:PictureBack, 509
Propriétés dans VBS:PictureDeactivated, 509
Propriétés dans VBS:PictureDown, 510
Propriétés dans VBS:PictureName, 510
Propriétés dans VBS:PictureSelected, 510
Propriétés dans VBS:PictureThumb, 511
Propriétés dans VBS:PictureUnselected, 511
Propriétés dans VBS:PictureUp, 511
Propriétés dans VBS:PicUpReferenced, 512
Propriétés dans VBS:PicUpTransparent, 512
Propriétés dans VBS:PicUpUseTransColor, 512
Propriétés dans VBS:PicUseTransColor, 513
Propriétés dans VBS:PointCount, 513
Propriétés dans VBS:Position, 513
Propriétés dans VBS:Precisions, 514
Propriétés dans VBS:PrecisionX, 514
Propriétés dans VBS:PrecisionY, 515
Propriétés dans VBS:PredefinedAngles, 515
Propriétés dans VBS:Pressed, 515
Propriétés dans VBS:PrintJob, 516
Propriétés dans VBS:Process, 516
Propriétés dans VBS:ProjectPath, 517
Propriétés dans VBS:ProviderClsid, 517
Propriétés dans VBS:ProviderType, 518
Propriétés dans VBS:QualityCode, 518
Propriétés dans VBS:Radius, 519
Propriétés dans VBS:RadiusHeigth, 520
Propriétés dans VBS:RadiusWidth, 520
Propriétés dans VBS:RangeMax, 520
Propriétés dans VBS:RangeMin, 521
Propriétés dans VBS:Rectangular, 521
Propriétés dans VBS:ReferenceRotationLeft, 521
Propriétés dans VBS:ReferenceRotationTop, 522
Propriétés dans VBS:RelayCurves, 522
Propriétés dans VBS:Relevant, 522
Propriétés dans VBS:Replacement, 523
Propriétés dans VBS:ReplacementColor, 523
Propriétés dans VBS:RightComma, 523
Propriétés dans VBS:RoundCornerHeight, 525
Propriétés dans VBS:RoundCornerWidth, 525
Propriétés dans VBS:RulerPrecisions, 528
Propriétés dans VBS:RulerPrecisionX, 529
Propriétés dans VBS:RulerPrecisionY, 529
Propriétés dans VBS:SameSize, 530
Propriétés dans VBS:ScaleColor, 530
Propriétés dans VBS:ScaleTicks, 530
Propriétés dans VBS:Scaling, 531
Propriétés dans VBS:ScalingType, 531
Propriétés dans VBS:ScalingTypeX, 532
Propriétés dans VBS:ScalingTypeY, 532
Propriétés dans VBS:Screen, 533
Propriétés dans VBS:ScreenItems, 533
Propriétés dans VBS:ScreenName, 533
Propriétés dans VBS:Screens, 300
Propriétés dans VBS:ScrollBars, 534
Propriétés dans VBS:SecondNeedleHeight, 535
Propriétés dans VBS:SecondNeedleWidth, 536
Propriétés dans VBS:SelBGColor, 536
Propriétés dans VBS:SelectionMode, 538
Propriétés dans VBS:SelectionRectColor, 539
Propriétés dans VBS:SelectionRectWidth, 540
Propriétés dans VBS:SelectionType, 540
Propriétés dans VBS:SelTextColor, 541
Propriétés dans VBS:ServerData, 542
Propriétés dans VBS:ServerNames, 544
Propriétés dans VBS:ServerPrefix, 544
Propriétés dans VBS:ShowBar, 545
Propriétés dans VBS:ShowDanger, 545
Propriétés dans VBS:ShowDecimalPoint, 546
Propriétés dans VBS:ShowNormal, 546
Propriétés dans VBS:ShowPeak, 546
Propriétés dans VBS:ShowPosition, 547
Propriétés dans VBS:ShowRulerImmediately, 547
Propriétés dans VBS:ShowThumb, 550
Propriétés dans VBS:ShowValuesExponentialX, 551
Propriétés dans VBS:ShowValuesExponentialY, 551
Propriétés dans VBS:ShowWarning, 551
Propriétés dans VBS:SignificantMask, 552
Propriétés dans VBS:SmallChange, 553

Propriétés dans VBS:SourceBeginTime, 554
Propriétés dans VBS:SourceEndTime, 555
Propriétés dans VBS:SourceNumberOfUAValues, 555
Propriétés dans VBS:SourceNumberOfValues, 555
Propriétés dans VBS:SourceTagNameX, 556
Propriétés dans VBS:SourceTagNameY, 556
Propriétés dans VBS:SourceTagProviderDataX, 556
Propriétés dans VBS:SourceTagProviderDataY, 557
Propriétés dans VBS:SourceTimeRange, 557
Propriétés dans VBS:SourceUAArchive, 557
Propriétés dans VBS:SourceUAArchiveStartID, 558
Propriétés dans VBS:SourceUAColumnX, 558
Propriétés dans VBS:SourceUAColumnY, 559
Propriétés dans VBS:SquareExtent, 559
Propriétés dans VBS:StartAngle, 559
Propriétés dans VBS:Statusbar, 560
Propriétés dans VBS:StatusbarPanes, 565
Propriétés dans VBS:StatusbarStretch, 566
Propriétés dans VBS:TagName, 568
Propriétés dans VBS:TagPrefix, 568
Propriétés dans VBS:TagProviderClsid, 569
Propriétés dans VBS:Tags, 569
Propriétés dans VBS:Template, 570
Propriétés dans VBS:Text, 571
Propriétés dans VBS:ThumbBackColor, 571
Propriétés dans VBS:TicColor, 571
Propriétés dans VBS:TicFont, 571
Propriétés dans VBS:Ticks, 573
Propriétés dans VBS:TicksColor, 574
Propriétés dans VBS:TickStyle, 574
Propriétés dans VBS:TicOffset, 572
Propriétés dans VBS:TicTextColor, 572
Propriétés dans VBS:TicTextOffset, 573
Propriétés dans VBS:TicWidth, 573
Propriétés dans VBS:TimeAxis, 574
Propriétés dans VBS:TimeAxisFormat, 577
Propriétés dans VBS:TimeAxisX, 581
Propriétés dans VBS:TimeColumnAlignment, 583
Propriétés dans VBS:TimeFormat, 591
Propriétés dans VBS:TimeJump, 592
Propriétés dans VBS:TimeJumpColor, 592
Propriétés dans VBS:TimeOverlap, 593
Propriétés dans VBS:TimeOverlapColor, 593
Propriétés dans VBS:TimeRange, 594
Propriétés dans VBS:TimeRangeBase, 594
Propriétés dans VBS:TimeRangeFactor, 594
Propriétés dans VBS:TimeStamp, 595
Propriétés dans VBS:TimeZone, 597
Propriétés dans VBS:TitleCut, 598
Propriétés dans VBS:Titleline, 599
Propriétés dans VBS:Toggle, 600
Propriétés dans VBS:ToleranceHigh, 600
Propriétés dans VBS:ToleranceLow, 601
Propriétés dans VBS:Toolbar, 601
Propriétés dans VBS:ToolbarAlignment, 602
Propriétés dans VBS:ToolbarButtons, 610
Propriétés dans VBS:ToolbarHotKeys, 611
Propriétés dans VBS:ToolTipText, 613
Propriétés dans VBS:Top, 614
Propriétés dans VBS:TopConnectedConnectionPointIndex, 615
Propriétés dans VBS:TopConnectedObjectName, 615
Propriétés dans VBS:Transparent, 616
Propriétés dans VBS:Trend, 616
Propriétés dans VBS:TrendColor, 619
Propriétés dans VBS:Type, 638
Propriétés dans VBS:TypeAlarmHigh, 640
Propriétés dans VBS:TypeAlarmLow, 640
Propriétés dans VBS:TypeLimitHigh4, 641
Propriétés dans VBS:TypeLimitHigh5, 641
Propriétés dans VBS:TypeLimitLow4, 641
Propriétés dans VBS:TypeLimitLow5, 642
Propriétés dans VBS:TypeToleranceHigh, 642
Propriétés dans VBS:TypeToleranceLow, 642
Propriétés dans VBS:TypeWarningHigh, 642
Propriétés dans VBS:TypeWarningLow, 643
Propriétés dans VBS:UnitColor, 643
Propriétés dans VBS:UnitFont, 643
Propriétés dans VBS:UnitOffset, 644
Propriétés dans VBS:UnitText, 644
Propriétés dans VBS:UnselBGColor, 644
Propriétés dans VBS:UnselTextColor, 645
Propriétés dans VBS:UpdateCycle, 645
Propriétés dans VBS:UpperLimit, 645
Propriétés dans VBS:UpperLimitColor, 646
Propriétés dans VBS:UpperLimitValue, 646
Propriétés dans VBS:Value, 652
Propriétés dans VBS:ValueColumnAlignment, 659
Propriétés dans VBS:ValueMax, 666
Propriétés dans VBS:ValueMin, 666
Propriétés dans VBS:Variable, 666
Propriétés dans VBS:Visible, 667
Propriétés dans VBS:Warning, 668
Propriétés dans VBS:WarningColor, 668
Propriétés dans VBS:WarningHigh, 668
Propriétés dans VBS:WarningLow, 669
Propriétés dans VBS:Width, 669
Propriétés dans VBS:WindowBorder, 670
Propriétés dans VBS:WindowsStyle, 671
Propriétés dans VBS:WindowType, 671
Propriétés dans VBS:WithAxes, 672
Propriétés dans VBS:WithLabels, 672

Propriétés dans VBS:ZeroPoint, 679
Propriétés dans VBS:ZeroPointValue, 679
Propriétés dans VBS:Zoom, 680
Propriétés de l'ordinateur, 837
Propriétés TagScaleParam1, 2368
Propriétés TagScaleParam2, 2368
Propriétés TagScaleParam3, 2369
Propriétés TagScaleParam4, 2369
Propriétés : FillingDirection, 2168
PropriétésProperties , 1724
Protéger le code VBA, 1607
Push Button Control, 268
putc, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

Q

qsort, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
QuitHorn, 752
QuitSelected, 752
QuitVisible, 753

R

rand, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
ReadTags, 757
realloc, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Référence, 1724
    Gestion des événements, 1724
    Méthodes, 1724
    Modèle d'objet, 1724
    Modèle d'objet VBA, 1724
    Objets et listes, 1724
    Propriétés, 1724
Référence VBA, 1724
    Evénements, 1724
    Gestion des événements, 1724
    Méthodes, 1724
    Modèle d'objet, 1724
    Modèle d'objet VBA, 1724
    Propriétés, 1724
Références, 1607
Registry2, 909
Règle d'écriture du code de WinCC, 873
Regroupement, 1668
    Accès aux propriétés de l'objet, 1668
    Ajouter un objet avec VBA, 1670
    créer avec VBA, 1670
    dissocier avec VBA, 1670
    éditer avec VBA, 1668, 1670
    éditer les objets contenus, 1673
    Principes de base, 1668
    supprimer avec VBA, 1670
    Supprimer un objet avec VBA, 1670
remove, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

rename, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Renommer, 55, 80
Renommer un module, 55
Renommer:Action, 80
Renommer:Procédure, 55
ReportJob, 933
rewind, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
RPTJobPreview, 968
RPTJobPrint, 969
RptShowError, 970
RulerControl, 272
Runtime, 82
Runtime:Activer des actions globales, 82

S

Saisie du mot de passe, 864, 876
ScreenItem, 260, 264, 272, 279, 288
    Contrôle, 190
    Contrôles, 224
    Instance de bloc d'affichage, 196
    Objet utilisateur, 290
    WinCC Alarm Control, 282
    WinCC AlarmControl, 248
ScreenItem : Champ de combinaison, 199
ScreenItem : Champ de liste, 199
ScreenItem : Coude de tuyau, 224
ScreenItem : Double raccord en T, 223
ScreenItem : Raccord en T, 222
ScreenItem : Texte multiligne, 200
ScreenItem : Tuyau polygonal, 222
ScreenItem : WinCC OnlineTableControl, 260
ScreenItem:Arc de cercle, 160
ScreenItem:Arc d'ellipse, 154
ScreenItem:Bargraphe, 184
ScreenItem:Bouton rond, 217
ScreenItem:Button, 210
ScreenItem:Case à cocher, 213
ScreenItem:Case d'option, 215
ScreenItem:Cercle, 158
ScreenItem:Champ d'E/S, 193
ScreenItem:Connecteur, 178
ScreenItem:Ellipse, 151
ScreenItem:Fenêtre d'application, 183
ScreenItem:Fenêtre de vue, 188
ScreenItem:Groupe, 291
ScreenItem:HMI Symbol Library, 246
ScreenItem:Indicateur d'état, 208
ScreenItem:Liste de textes, 206
ScreenItem:Objet graphique, 196
ScreenItem:Objet OLE, 201
ScreenItem:Polygone, 166
ScreenItem:Rectangle, 170
ScreenItem:Rectangle arrondi, 173
ScreenItem:Segment de cercle, 162
ScreenItem:Segment d'ellipse, 156
ScreenItem:Signalisation groupée, 203
ScreenItem:Slider, 219
ScreenItem:Texte statique, 176
ScreenItem:Trait, 164
ScreenItem:Trait polygonal, 168
ScreenItem:Types d'objet, 150
ScreenItem:WinCC Digital Analog Clock, 251
ScreenItem:WinCC Function Trend Control, 284
ScreenItem:WinCC Gauge Control, 258
ScreenItem:WinCC Online Table Control, 286
ScreenItem:WinCC Push Button Control, 268
ScreenItem:WinCC Slider Control, 275
ScreenItems : WinCC FunctionTrendControl, 254
Script en C, 1688
    Dynamiser une propriété avec VBA, 1688
Script VB, 1691
    Dynamisation d'une propriété par VBA, 1691
SelectedStatisticArea, 769
ServerExport, 769
ServerImport, 769
Set_Focus, 1265
SetActualPointLeft, 1274
SetActualPointTop, 1275
SetAlarmHigh, 1299
SetAlarmLow, 1299
SetAlignment, 1224
SetAlignmentLeft, 1266
SetAlignmentTop, 1267
SetAssumeOnExit, 1291
SetAssumeOnFull, 1292

SetAverage, 1337
SetAxisSection, 1225
SetBackBorderWidth, 1380
SetBackColor, 1233
SetBackColor2, 1234
SetBackColor3, 1235
SetBackColorBottom, 1236
SetBackColorTop, 1237
SetBackFlashColorOff, 1255
SetBackFlashColorOn, 1255
SetBasePicTransColor, 1374
SetBasePicUseTransColor, 1375
SetBitNumber, 1293
SetBorderBackColor, 1238
SetBorderColor, 1239
SetBorderColorBottom, 1239
SetBorderColorTop, 1240
SetBorderEndStyle, 1381
SetBorderFlashColorOff, 1256
SetBorderFlashColorOn, 1257
SetBorderStyle, 1382
SetBorderWidth, 1383
SetBoxAlignment, 1384
SetBoxCount, 1276
SetBoxType, 1338
setbuf, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
SetButtonColor, 1241
SetCheckAlarmHigh, 1300
SetCheckAlarmLow, 1301
SetCheckLimitHigh4, 1302
SetCheckLimitHigh5, 1303
SetCheckLimitLow4, 1303
SetCheckLimitLow5, 1304
SetCheckToleranceHigh, 1305
SetCheckToleranceLow, 1306
SetCheckWarningHigh, 1307
SetCheckWarningLow, 1307
SetClearOnError, 1293
SetClearOnNew, 1294
SetColorAlarmHigh, 1308
SetColorAlarmLow, 1309
SetColorBottom, 1242
SetColorChangeType, 1338
SetColorLimitHigh4, 1310
SetColorLimitHigh5, 1311
SetColorLimitLow4, 1312
SetColorLimitLow5, 1313
SetColorToleranceHigh, 1314
SetColorToleranceLow, 1314
SetColorTop, 1243
SetColorWarningHigh, 1315
SetColorWarningLow, 1316
SetCursorControl, 1339
SetCursorMode, 1340
SetDirection, 1277
SetEditAtOnce, 1341
SetEndAngle, 1278
SetExponent, 1225
SetExtendedOperation, 1342
SetFillColor, 1244
SetFilling, 1253
SetFillingIndex, 1253
SetFillStyle, 1385
SetFillStyle2, 1386
SetFlashBackColor, 1258
SetFlashBorderColor, 1259
SetFlashFlashPicture, 1376
SetFlashForeColor, 1260
SetFlashPicTransColor, 1377
SetFlashPicUseTransColor, 1378
SetFlashRateBackColor, 1261
SetFlashRateBorderColor, 1261
SetFlashRateFlashPic, 1378
SetFlashRateForeColor, 1262
SetFontBold, 1268
SetFontItalic, 1269
SetFontName, 1269
SetFontSize, 1270
SetFontUnderline, 1271
SetForeColor, 1245
SetForeFlashColorOff, 1263
SetForeFlashColorOn, 1264
SetHeight, 1278
SetHiddenInput, 1295
SetHysteresis, 1343
SetHysteresisRange, 1343
SetIndex, 1379
SetItemBorderBackColor, 1246
SetItemBorderColor, 1246
SetItemBorderStyle, 1386
SetItemBorderWidth, 1387
SetLanguage, 1509
SetLeft, 1279
SetLeftComma, 1226
SetLimitHigh4, 1317

SetLimitHigh5, 1318
SetLimitLow4, 1319
SetLimitLow5, 1319
SetLimitMax, 1320
SetLimitMin, 1321
SetLink, 1335
SetLongStrokesBold, 1227
SetLongStrokesOnly, 1228
SetLongStrokesSize, 1229
SetMarker, 1322
SetMax, 1344
SetMin, 1345
SetNumberLines, 1296
SetOffsetLeft, 1345
SetOffsetTop, 1346
SetOperation, 1347
SetOperationMessage, 1348
SetOperationReport, 1349
SetOrientation, 1272
SetOutputValueChar, 1297
SetOutputValueDouble, 1297
SetPasswordLevel, 1350
SetPicDeactTransparent, 1360
SetPicDeactUseTransColor, 1361
SetPicDownTransparent, 1362
SetPicDownUseTransColor, 1363
SetPicTransColor, 1364
SetPictureDeactivated, 1365
SetPictureDown , 1365
SetPictureName, 1351
SetPictureUp , 1366
SetPicUpTransparent, 1367
SetPicUpUseTransColor, 1368
SetPicUseTransColor, 1369
SetPointCount, 1280
SetPosition, 1358
SetPressed, 1388
SetProcess, 1351
SetPropBOOL, 1370
SetPropChar, 1371
SetPropDouble
0, 1372
SetPropWord, 1373
SetRadius, 1281
SetRadiusHeight, 1282
SetRadiusWidth, 1282
SetRangeMax, 1358
SetRangeMin, 1359
SetReferenceRotationLeft, 1283
SetReferenceRotationTop, 1284
SetRightComma, 1230
SetRotationAngle, 1285
SetRoundCornerHeight, 1286
SetRoundCornerWidth, 1286
SetScaleColor, 1247
SetScaleTicks, 1230
SetScaling, 1231
SetScalingType, 1232
SetSelBGColor, 1248
SetSelTextColor, 1249
SetSmallChange, 1352
SetStartAngle, 1287
SetTagBit, 1495
SetTagBitState, 1473
SetTagBitStateWait, 1461
SetTagBitWait, 1484
SetTagByte, 1496
SetTagByteState, 1474
SetTagByteStateWait, 1462
SetTagByteWait, 1485
SetTagChar, 1497
SetTagCharState, 1475
SetTagCharStateWait, 1463
SetTagCharWait, 1486
SetTagDouble, 1498
SetTagDoubleState, 1476
SetTagDoubleStateWait, 1464
SetTagDoubleWait, 1487
SetTagDWord, 1498
SetTagDWordState, 1477
SetTagDWordStateWait, 1465
SetTagDWordWait, 1487
SetTagFloat, 1499
SetTagFloatState, 1478
SetTagFloatStateWait, 1466
SetTagFloatWait, 1488
SetTagMultiStateWait, 1467
SetTagMultiWait, 1489
SetTagPrefix, 1353
SetTagRaw, 1500
SetTagRawState, 1479
SetTagRawStateWait, 1468
SetTagRawWait, 1490
SetTagSByte, 1501
SetTagSByteState, 1480
SetTagSByteStateWait, 1469
SetTagSByteWait, 1491
SetTagSDWord, 1502
SetTagSDWordState, 1481
SetTagSDWordStateWait, 1470
SetTagSDWordWait, 1492
SetTagSWord, 1503
SetTagSWordState, 1482
SetTagSWordStateWait, 1471

SetTagSWordWait, 1493
SetTagValue, 1504
SetTagValueWait, 1493
SetTagWord, 1505
SetTagWordState, 1483
SetTagWordStateWait, 1472
SetTagWordWait, 1495
SetText, 1273
SetToggle, 1389
SetToleranceHigh, 1323
SetToleranceLow, 1323
SetTop, 1288
SetTrend, 1354
SetTrendColor, 1250
SetTypeAlarmHigh, 1324
SetTypeAlarmLow, 1325
SetTypeLimitHigh4, 1326
SetTypeLimitHigh5, 1327
SetTypeLimitLow4, 1328
SetTypeLimitLow5, 1329
SetTypeToleranceHigh, 1330
SetTypeToleranceLow, 1331
SetTypeWarningHigh, 1332
SetTypeWarningLow, 1333
SetUnselBGColor, 1251
SetUnselTextColor, 1252
setvbuf, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
SetVisible, 1355
SetWarningHigh, 1334
SetWarningLow, 1334
SetWidth, 1289
SetWindowsStyle, 1390
SetZeroPoint, 1290
SetZeroPointValue, 1356
SetZoom, 1357
ShowColumnSelection, 770
ShowComment, 770
ShowDisplayOptionsDialog, 771
ShowEmergencyQuitDialog, 771
ShowHelp, 771
ShowHideList, 772
ShowHitList, 772
ShowInfoText, 773
ShowLockDialog, 773
ShowLockList, 773
ShowLongTermArchiveList, 774
ShowMessageList, 774
ShowPercentageAxis, 775
ShowPropertyDialog, 775
ShowSelectArchive, 775
ShowSelection, 776
ShowSelectionDialog, 777
ShowSelectTimeBase, 776
ShowShortTermArchiveList, 777
ShowSort, 777
ShowSortDialog, 778
ShowTagSelection, 778
ShowTimebaseDialog, 779
ShowTimeSelection, 779
ShowTrendSelection, 779
Signet, 105
Signets:Activer dans le débogueur, 105
Signets:Sauter sur, 105
Signets:Supprimer dans le débogueur, 105
sin, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
sinh, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Slider Control, 275
sqrt, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

srand, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
StartStopUpdate, 780
stdio, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
stdlib, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
StorePicture, 1392
strcat, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strchr, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strcmp, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strcpy, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strcspn, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strerror, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strftime, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

string (chaîne), 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strlen, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strncat, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strncmp, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strncpy, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strpbrk, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strrchr, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strspn, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strstr, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strtod, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

strtok, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strtol, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
strtoul, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Style de police
    paramétrer, 850
Supprimer
    Trigger, 79
Supprimer:Actions et procédures, 38
Symbol Library, 246
Symbole des actions
    Caractéristique, 870
Syntaxe, 69
Syntaxe:Contrôle, 69
Syntaxe:Défaut, 69
SysFree, 979
SysMalloc, 979
system, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063

T

Table Control, 286
Tag Logging, 2492
    créer une archive de valeurs de process avec VBA, 2492
    créer une variable d'archive avec VBA, 2492
    éditer une archive de valeurs de process avec VBA, 2492
    éditer une variable d'archive avec VBA, 2492
    supprimer une archive de valeurs de process avec VBA, 2492
    supprimer une variable d'archive avec VBA, 2492
tan, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
tanh, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Temporisateur, 71, 76
Tester, 90
Tester:avec Debugger, 90
Text Library, 2520
    créer un texte avec VBA, 2520
    créer une langue avec VBA, 2520
    Lire un texte avec VBA, 2520
    Lire un TextID avec VBA, 2520

modifier un texte avec VBA, 2520
supprimer un texte avec VBA, 2520
time, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
TlgGetColumnPosition, 966
TlgGetNumberOfColumns, 960
TlgGetNumberOfRows, 960
TlgGetNumberOfTrends, 961
TlgGetRowPosition, 962
TlgGetRulerArchivNameTrend, 962
TlgGetRulerTimeTrend, 963
TlgGetRulerValueTrend, 964
TlgGetRulerVariableNameTrend, 965
TlgGetTextAtPos, 965
TlgTableWindowPressEditRecordButton, 934
TlgTableWindowPressFirstButton, 934
TlgTableWindowPressInsertRecordButton, 936
TlgTableWindowPressLastButton, 936
TlgTableWindowPressNextButton, 937
TlgTableWindowPressNextItemButton, 938
TlgTableWindowPressOpenArchiveVariableSelectionDlgButton, 939
TlgTableWindowPressOpenDlgButton, 939
TlgTableWindowPressOpenItemSelectDlgButton, 940
TlgTableWindowPressOpenTimeSelectDlgButton, 941
TlgTableWindowPressPrevButton, 942
TlgTableWindowPressPrevItemButton, 942
TlgTableWindowPressRemoveRecordButton, 943
TlgTableWindowPressStartStopButton, 943
TlgTrendWindowActivateCurve, 967
TlgTrendWindowPressFirstButton, 944
TlgTrendWindowPressHelpButton, 945
TlgTrendWindowPressLastButton, 946
TlgTrendWindowPressLinealButton, 946
TlgTrendWindowPressNextButton, 947
TlgTrendWindowPressNextItemButton, 948
TlgTrendWindowPressOneToOneButton, 949
TlgTrendWindowPressOpenArchiveVariableSelectionDlgButton, 950
TlgTrendWindowPressOpenDlgButton, 950
TlgTrendWindowPressOpenItemSelectDlgButton, 951
TlgTrendWindowPressOpenTimeSelectDlgButton, 952
TlgTrendWindowPressPrevButton, 953
TlgTrendWindowPressPrevItemButton, 953
TlgTrendWindowPressPrintButton, 954
TlgTrendWindowPressStartStopButton, 956
TlgTrendWindowPressStatsResultButton, 956
TlgTrendWindowPressStatsSelectRangeButton, 957
TlgTrendWindowPressZoomInButton, 958
TlgTrendWindowPressZoomOutButton, 959
tmpfile, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
tmpnam, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
tolower, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
toupper, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Trabsmission, 1704
des événements, 1704

TraceText, 1511
TraceTime, 1512
Trigger
supprimer, 79
Trigger:Animationstrigger, 75
Trigger:Crossreference, 71
Trigger:cyclique, 71
Trigger:Variable, 71
Type de déclenchement, 828, 879
Types d'erreur, 93
Types d'erreur:Erreur de syntaxe, 93
Types d'erreur:Erreur Runtime, 93
Types d'erreur:Erreurs d'exécution, 93
Types d'erreur:Erreurs logiques, 93
Types d'objet:Groupe, 291
Types d'objets
Contrôles, 224
Objet utilisateur, 290

U

ungetc, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
UnhideAlarm, 781
UnlockAlarm, 782
UsedLanguage, 2387
UserArchiveControl, 279

V

Variable, 26, 77, 2481
créer avec VBA, 2481
Déterminer un type avec VBA, 2481
Lire les propriétés avec VBA, 2481
Lire un seuil avec VBA, 2481
Modifier un seuil avec VBA, 2481
Modifier un type avec VBA, 2481
supprimer avec VBA, 2481
Variable globale:Utilisation dans VBS, 26
Variable:Configurer un déclenchement pour une action, 77
Variable:globale dans VBS, 26
Variables C globales, 838
Définition, 838
Utilisation, 838
Variables globales, 26
VBA, 1606
Accès à la bibliothèque des modules, 1640
Accès à la bibliothèque des outils, 1640
Accès à la copie d'une vue, 1647
Accès à l'objet Groupe, 1668
Accès à MS Excel avec VBA, 1708
Accès aux applications externes, 1707
Accès aux autres programmes, 1707
Accès aux niveaux, 1647
Accès aux objets dans le Graphics Designer, 1653
Affecter un macro VBA à un menu personnalisé, 1636
Affecter un macro VBA à une barre d'outils personnalisée, 1636
Affecter un texte d'aide à un menu personnalisé, 1633
Affecter un texte d'aide à une barre d'outils personnalisée, 1633
Ajouter un icône d'outil dans une barre d'outils personnalisée, 1630
Ajouter une option au menu personnalisé, 1623
Alarm Logging, 2532
Application, 1606
Barre d'outils personnalisée, 1620
Classe HMIGO, 2479
code VBA global, 1607
Code VBA spécifique à un projet, 1607
code VBA spécifique à une vue, 1607
Commander la visibilité des niveaux, 1650
Conception multilingue, 1617
Configuration avec VBA d'une action C à l'occurrence d'un événement, 1697
Configuration d'actions, 1693
Configuration d'actions commandées par événement, 1693
Configuration d'une action VB déclenchée par évènement, 1699
Configuration multilingue, 1617
Configurer les déclencheurs, 1701
Configurer un menu personnalisé, 1619
Configurer une barre d'outils personnalisée, 1619
Configurer une liaison directe, 1694
Création d'un objet Groupe, 1670
Créer un menu multilingue personnalisé, 1626
Créer un menu personnalisé, 1622
Créer un menu spécifique à une application, 1622
Créer une barre d'outils personnalisée, 1629

Créer une barre d'outils spécifique à une application, 1629
dans d'autres éditeurs WinCC, 2479
Dans le Graphics Designer, 1614
dissocier un objet groupe, 1670
Dynamic Wizards (différence), 1606
Dynamisation, 1680
Dynamisation de propriétés, 1682
Dynamisation d'une propriété par le dialogue Dynamique, 1685
Dynamisation d'une propriété par un script C, 1688
Dynamisation d'une propriété par un script VB, 1691
Editer des niveaux, 1650
Editer la bibliothèque des modules, 1643
Editer la bibliothèque des outils, 1643
Editer un objet complexe, 1657
Editer un objet simple, 1657
Editer un objet Windows, 1657
Editer une vue, 1647
Edition de la copie d'une vue :, 1651
Edition d'objets dans le Graphics Designer, 1653
Edition d'objets d'un objet-groupe, 1673
Edition d'un objet Groupe, 1670
Edition d'un objet utilisateur, 1677
Exécution de macros VBA, 1612
Exporter un code, 1611
Gestion des événements, 1704
Importer un code, 1611
Insérer dans une vue un objet provenant de la bibliothèque des modules, 1645
Insérer dans une vue un objet provenant de la bibliothèque des outils, 1645
Insertion d'objets dans une vue, 1655
Insertion d'un ActiveX Control dans une vue, 1663
Langue de configuration, 1617
Langue d'interface, 1617
Menu personnalisé, 1620
Objet groupe, 1668
Objet OLE, 1661
Objet utilisateur, 1676
ODK (différence), 1606
Scripts en C (différence), 1606
Scripts en VB (différence), 1606
Stock de variables, 2481
Supprimer un objet groupe, 1670
Tag Logging, 2492
Text Library, 2520
VBA dans le Graphics Designer, 1614
VBS, 26, 792
Exemples, 819
Exemples dans WinCC, 792
Groupe cible de la documentation, 12
Méthodes, 680
Modèle d'objet, 110
Objets, 114
Principes de base, 825
Référence, 110
Scénarios d'application, 12
VBS:Action, 19
VBS:CrossReference, 24
VBS:Editeurs, 28
VBS:Fonctions standard, 45
VBS:Global Script:Appel, 29
VBS:Module, 16
VBS:Procédure, 16
VBS:Propriétés, 292
VBS:Types d'objet, 150
VBS:Utilisation de variables globales, 26
VBScript:Activer dans Runtime, 82
VBScript:débogger, 90
VBScript:Imprimer, 109
VBScript:Ouvrir dans Debugger, 101
vfprintf, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Visual Basic Script, 12
Visual Basic Script sous WinCC, 12
vsprintf, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063
Vue, 1651
éditer une copie avec VBA, 1651

W

WinCC, 12, 150
Langues de scriptage, 12

Visual Basic Script, 12
WinCC Alarm Control, 282
WinCC AlarmControl, 248
WinCC MediaControl, 260
WinCC : WinCC FunctionTrendControl, 254
WinCC : WinCC OnlineTableControl, 260
WinCC : WinCC OnlineTrendControl, 264
WinCC : WinCC RulerControl, 272
WinCC AlarmControl
Exemple VBS, 818
Objet HitlistColumn, 229
Objet MessageBlock, 230
Objet MessageColumn, 231
Objet OperatorMessage, 232
Objet StatusbarElement, 237
Objet ToolbarButton, 239
WinCC FunctionTrendControl
Objet StatusbarElement, 237
Objet ToolbarButton, 239
Objet Trend, 240
Objet TrendWindow, 242
Objet XAxis, 245
Objet YAxis, 245
WinCC OnlineTableControl
Exemple de script en C, 1580
Exemple VBS, 816
Objet StatusbarElement, 237
Objet TimeColumn, 238
Objet ToolbarButton, 239
Objet ValueColumn, 244
WinCC OnlineTrendControl
Exemple de script en C, 1581
Exemple VBS, 812
Exemple VBS pour la courbe de consigne, 814
Objet StatusbarElement, 237
Objet ToolbarButton, 239
Objet Trend, 240
Objet TrendWindow, 242
WinCC RulerControl
Objet RulerBlock, 234
Objet RulerColumn, 234
Objet StatisticAreaColumn, 235
Objet StatisticResultColumn, 236
Objet StatusbarElement, 237
Objet ToolbarButton, 239
WinCC UserArchiveControl
Objet Column, 228
Objet StatusbarElement, 237
Objet ToolbarButton, 239
WinCC:Types d'objet graphique, 150
WinCC:WinCC Digital Analog Clock, 251
WinCC:WinCC Function Trend Control, 284
WinCC:WinCC Gauge Control, 258
WinCC:WinCC Online Table Control Online, 286
WinCC:WinCC Online Trend Control, 288
WinCC:WinCC Push Button Control Controls:WinCC Push Button Control, 268
WinCC:WinCC UserArchiveControl, 279
WriteTag, 786

Z

Zone de déclaration, 63
Zone de déclaration:d'une action, 63
ZoomArea, 787
ZoomInOut, 787
ZoomInOutTime, 788
ZoomInOutValues, 788
ZoomInOutX, 788
ZoomInOutY, 789
ZoomMove, 789