

SIEMENS

SIMATIC

WinCC WinCC Professional V13 SP2 - Programmierreferenz

Systemhandbuch

C-Scripting (RT Professional)

1

Runtime API (RT
Professional)

2

VB-Scripting (Panels,
Comfort Panels, RT
Advanced, RT Professional)

3

Ausdruck der Onlinehilfe

03/2017

Ausdruck der Onlinehilfe

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten wird , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten kann , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 VORSICHT
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Inhaltsverzeichnis

1	C-Scripting (RT Professional)	23
1.1	Systemfunktionen (RT Professional).....	23
1.1.1	ActivateNextScreen (RT Professional).....	23
1.1.2	ActivatePreviousScreen (RT Professional).....	24
1.1.3	ActivateScreen (RT Professional).....	25
1.1.4	ActivateScreenInScreenWindow (RT Professional).....	26
1.1.5	ActivateStartScreen (RT Professional).....	27
1.1.6	ActivateStoredScreen (RT Professional).....	28
1.1.7	DecreaseTag (RT Professional).....	29
1.1.8	GetLocalScreen (RT Professional).....	31
1.1.9	GetLinkedTag (RT Professional).....	33
1.1.10	GetLanguageByLocaleID (RT Professional).....	34
1.1.11	GetParentScreen (RT Professional).....	36
1.1.12	GetParentScreenWindow (RT Professional).....	37
1.1.13	GetProp (RT Professional).....	39
1.1.13.1	GetPropBOOL (RT Professional).....	39
1.1.13.2	GetPropChar (RT Professional).....	40
1.1.13.3	GetPropDouble (RT Professional).....	42
1.1.13.4	GetPropLong (RT Professional).....	43
1.1.14	GetServerTagPrefix (RT Professional).....	44
1.1.15	GetTag (RT Professional).....	47
1.1.15.1	GetTag-Funktionen (RT Professional).....	47
1.1.15.2	GetTagDateTime-Funktion (RT Professional).....	48
1.1.15.3	GetTagMultiStateQCWait-Funktionen (RT Professional).....	49
1.1.15.4	GetTagMultiStateWait-Funktionen (RT Professional).....	50
1.1.15.5	GetTagMultiWait-Funktionen (RT Professional).....	52
1.1.15.6	GetTagState-Funktionen (RT Professional).....	53
1.1.15.7	GetTagStateQC-Funktionen (RT Professional).....	56
1.1.15.8	GetTagStateQCWait-Funktionen (RT Professional).....	58
1.1.15.9	GetTagStateWait-Funktionen (RT Professional).....	61
1.1.15.10	GetTagValue-Funktionen (RT Professional).....	63
1.1.15.11	GetTagValueStateQC-Funktionen (RT Professional).....	65
1.1.15.12	GetTagValueStateQCWait-Funktionen (RT Professional).....	66
1.1.15.13	GetTagValueWait-Funktionen (RT Professional).....	68
1.1.15.14	GetTagWait-Funktionen (RT Professional).....	69
1.1.16	IncreaseTag (RT Professional).....	71
1.1.17	InquireLanguage (RT Professional).....	72
1.1.18	InverseLinearScaling (RT Professional).....	74
1.1.19	InvertBit (RT Professional).....	76
1.1.20	InvertBitInTag (RT Professional).....	77
1.1.21	IsUserAuthorized (RT Professional).....	78
1.1.22	LinearScaling (RT Professional).....	79
1.1.23	ReportJob (RT Professional).....	81
1.1.24	ResetBit (RT Professional).....	82
1.1.25	ResetBitInTag (RT Professional).....	83
1.1.26	SetBit (RT Professional).....	84

1.1.27	SetBitInTag (RT Professional).....	85
1.1.28	SetLanguageByLocaleID (RT Professional).....	87
1.1.29	SetProp (RT Professional).....	88
1.1.29.1	SetPropBOOL (RT Professional).....	88
1.1.29.2	SetPropChar (RT Professional).....	90
1.1.29.3	SetPropDouble (RT Professional).....	91
1.1.29.4	SetPropLong (RT Professional).....	93
1.1.30	SetPropertyByConstant (RT Professional).....	95
1.1.31	SetPropertyByProperty (RT Professional).....	96
1.1.32	SetPropertyByTag (RT Professional).....	98
1.1.33	SetPropertyByTagIndirect (RT Professional).....	100
1.1.34	SetTag (RT Professional).....	101
1.1.34.1	SetTag-Funktionen (RT Professional).....	101
1.1.34.2	SetTagDateTime (RT Professional).....	103
1.1.34.3	SetTagMultiStateWait-Funktionen (RT Professional).....	104
1.1.34.4	SetTagMultiWait-Funktionen (RT Professional).....	105
1.1.34.5	SetTagState-Funktionen (RT Professional).....	108
1.1.34.6	SetTagStateWait-Funktionen (RT Professional).....	110
1.1.34.7	SetTagValue-Funktionen (RT Professional).....	113
1.1.34.8	SetTagValueWait-Funktionen (RT Professional).....	115
1.1.34.9	SetTagWait-Funktionen (RT Professional).....	116
1.1.34.10	SetTag (RT Professional).....	118
1.1.34.11	SetTagByProperty (RT Professional).....	119
1.1.34.12	SetTagByTagIndirect (RT Professional).....	121
1.1.34.13	SetTagIndirect (RT Professional).....	122
1.1.34.14	SetTagIndirectByProperty (RT Professional).....	122
1.1.34.15	SetTagIndirectByTagIndirect (RT Professional).....	124
1.1.34.16	SetTagIndirectWithOperatorInputAlarm (RT Professional).....	125
1.1.34.17	SetTagWithOperatorEvent (RT Professional).....	125
1.1.35	StartProgram (RT Professional).....	126
1.1.36	StopRuntime (RT Professional).....	127
1.1.37	StoreScreen (RT Professional).....	128
1.1.38	TriggerOperatorEvent (RT Professional).....	129
1.1.39	UA (Recipe) (RT Professional).....	130
1.1.39.1	uaAddArchive (RT Professional).....	130
1.1.39.2	uaAddField (RT Professional).....	131
1.1.39.3	uaArchiveClose (RT Professional).....	132
1.1.39.4	uaArchiveDelete (RT Professional).....	132
1.1.39.5	uaArchiveExport (RT Professional).....	133
1.1.39.6	uaArchiveGetCount (RT Professional).....	134
1.1.39.7	uaArchiveGetFieldLength (RT Professional).....	135
1.1.39.8	uaArchiveGetFieldName (RT Professional).....	135
1.1.39.9	uaArchiveGetFields (RT Professional).....	136
1.1.39.10	uaArchiveGetFieldType (RT Professional).....	137
1.1.39.11	uaArchiveGetFieldValueDate (RT Professional).....	137
1.1.39.12	uaArchiveGetFieldValueDouble (RT Professional).....	138
1.1.39.13	uaArchiveGetFieldValueFloat (RT Professional).....	139
1.1.39.14	uaArchiveGetFieldValueLong (RT Professional).....	140
1.1.39.15	uaArchiveGetFieldValueString (RT Professional).....	141
1.1.39.16	uaArchiveGetFilter (RT Professional).....	141
1.1.39.17	uaArchiveGetID (RT Professional).....	142
1.1.39.18	uaArchiveGetName (RT Professional).....	143

1.1.39.19	uaArchiveGetSor (RT Professional).....	143
1.1.39.20	uaArchiveImport (RT Professional).....	144
1.1.39.21	uaArchiveInsert (RT Professional).....	145
1.1.39.22	uaArchiveMoveFirst (RT Professional).....	146
1.1.39.23	uaArchiveMoveLast (RT Professional).....	146
1.1.39.24	uaArchiveMoveNext (RT Professional).....	147
1.1.39.25	uaArchiveMovePrevious (RT Professional).....	147
1.1.39.26	uaArchiveOpen (RT Professional).....	148
1.1.39.27	uaArchiveReadTagValues (RT Professional).....	149
1.1.39.28	uaArchiveReadTagValuesByName (RT Professional).....	150
1.1.39.29	uaArchiveRequery (RT Professional).....	151
1.1.39.30	uaArchiveSetFieldValueDate (RT Professional).....	152
1.1.39.31	uaArchiveSetFieldValueDouble (RT Professional).....	152
1.1.39.32	uaArchiveSetFieldValueFloat (RT Professional).....	153
1.1.39.33	uaArchiveSetFieldValueLong (RT Professional).....	154
1.1.39.34	uaArchiveSetFieldValueString (RT Professional).....	155
1.1.39.35	uaArchiveSetFilter (RT Professional).....	156
1.1.39.36	uaArchiveSetSort (RT Professional).....	156
1.1.39.37	uaArchiveUpdate (RT Professional).....	157
1.1.39.38	uaArchiveWriteTagValues (RT Professional).....	158
1.1.39.39	uaArchiveWriteTagValuesByName (RT Professional).....	159
1.1.39.40	uaConnect (RT Professional).....	159
1.1.39.41	uaDisconnect (RT Professional).....	160
1.1.39.42	uaGetArchive (RT Professional).....	161
1.1.39.43	uaGetField (RT Professional).....	161
1.1.39.44	uaGetLastError (RT Professional).....	162
1.1.39.45	uaGetLastHResult (RT Professional).....	164
1.1.39.46	uaGetNumArchives (RT Professional).....	165
1.1.39.47	uaGetNumFields (RT Professional).....	165
1.1.39.48	uaQueryArchive (RT Professional).....	166
1.1.39.49	uaQueryArchiveByName (RT Professional).....	167
1.1.39.50	uaQueryConfiguration (RT Professional).....	168
1.1.39.51	uaReleaseArchive (RT Professional).....	169
1.1.39.52	uaReleaseConfiguration (RT Professional).....	169
1.1.39.53	uaRemoveAllArchives (RT Professional).....	170
1.1.39.54	uaRemoveAllFields (RT Professional).....	171
1.1.39.55	uaRemoveArchive (RT Professional).....	172
1.1.39.56	uaRemoveField (RT Professional).....	172
1.1.39.57	uaSetArchive (RT Professional).....	173
1.1.39.58	uaSetField (RT Professional).....	174
1.2	C-bib (RT Professional).....	175
1.2.1	ctype-Funktionen (RT Professional).....	175
1.2.2	Funktionsgruppe c_bib (RT Professional).....	176
1.2.3	math-Funktionen (RT Professional).....	176
1.2.4	memory-Funktionen (RT Professional).....	177
1.2.5	multibyte-Funktionen (RT Professional).....	178
1.2.6	stdio-Funktionen (RT Professional).....	179
1.2.7	stdlib-Funktionen (RT Professional).....	180
1.2.8	string-Funktionen (RT Professional).....	181
1.2.9	time-Funktionen (RT Professional).....	182
1.3	Strukturdefinition (RT Professional).....	182

1.3.1	Strukturdefinition CCAPErrorExecute (RT Professional).....	182
1.3.2	Strukturdefinition CCAPTime (RT Professional).....	184
1.3.3	Strukturdefinition CMN_ERROR (RT Professional).....	185
1.3.4	Strukturdefinition DM_TYPEREF (RT Professional).....	185
1.3.5	Strukturdefinition DM_VAR_UPDATE_STRUCT (RT Professional).....	186
1.3.6	Strukturdefinition DM_VAR_UPDATE_STRUCTEX (RT Professional).....	187
1.3.7	Strukturdefinition DM_VARKEY (RT Professional).....	188
1.3.8	Strukturdefinition LINKINFO (RT Professional).....	189
1.3.9	Strukturdefinition MSG_FILTER_STRUCT (RT Professional).....	190
1.3.10	Strukturdefinition MSG_RTDATA_STRUCT (RT Professional).....	193
2	Runtime API (RT Professional).....	195
2.1	Runtime API (RT Professional).....	195
2.2	Funktionen der Datenhaltung (RT Professional).....	196
2.2.1	Grundlagen (RT Professional).....	196
2.2.1.1	Übersicht über die Strukturen (RT Professional).....	196
2.2.1.2	Übersicht über die Funktionen (RT Professional).....	197
2.2.1.3	Quality Codes von HMI-Variablen (RT Professional).....	198
2.2.1.4	Konstanten (RT Professional).....	202
2.2.1.5	Fehlermeldungen (RT Professional).....	206
2.2.1.6	Konvertierungsroutinen (Control Center) (RT Professional).....	208
2.2.2	Strukturen (RT Professional).....	214
2.2.2.1	DM_CONNECTION_DATA (RT Professional).....	214
2.2.2.2	DM_CONNKEY (RT Professional).....	215
2.2.2.3	DM_CYCLE_INFO (RT Professional).....	216
2.2.2.4	DM_DATA_SERVICE (RT Professional).....	217
2.2.2.5	DM_DIRECTORY_INFO (RT Professional).....	218
2.2.2.6	DM_DLGOPTIONS (RT Professional).....	219
2.2.2.7	DM_FORMAT_INFO (RT Professional).....	220
2.2.2.8	DM_MACHINE_TABLE (RT Professional).....	221
2.2.2.9	DM_PROJECT_INFO (RT Professional).....	222
2.2.2.10	DM_SEND_DATA_STRUCT (RT Professional).....	223
2.2.2.11	DM_SD_TARGET_MACHINE (RT Professional).....	225
2.2.2.12	DM_SD_TARGET_APP (RT Professional).....	226
2.2.2.13	DM_TYPEREF (RT Professional).....	227
2.2.2.14	DM_VAR_UPDATE_STRUCT (RT Professional).....	228
2.2.2.15	DM_VAR_UPDATE_STRUCTEX (RT Professional).....	230
2.2.2.16	DM_VARFILTER (RT Professional).....	233
2.2.2.17	DM_VARGRP_DATA (RT Professional).....	235
2.2.2.18	DM_VARGRPKEY (RT Professional).....	236
2.2.2.19	DM_VARIABLE_DATA (RT Professional).....	237
2.2.2.20	DM_VARIABLE_DATA4 (RT Professional).....	239
2.2.2.21	DM_VARKEY (RT Professional).....	241
2.2.2.22	DM_VARLIMIT (RT Professional).....	244
2.2.2.23	MCP_NEWVARIABLE_DATA (RT Professional).....	245
2.2.2.24	MCP_NEWVARIABLE_DATA_4 (RT Professional).....	247
2.2.2.25	MCP_NEWVARIABLE_DATA_5 (RT Professional).....	249
2.2.2.26	MCP_NEWVARIABLE_DATA_EX (RT Professional).....	252
2.2.2.27	MCP_NEWVARIABLE_DATA_EX4 (RT Professional).....	254
2.2.2.28	MCP_VARIABLE_COMMON (RT Professional).....	256
2.2.2.29	MCP_VARIABLE_COMMON_EX (RT Professional).....	258
2.2.2.30	MCP_VARIABLE_LIMITS (RT Professional).....	260

2.2.2.31	MCP_VARIABLE_LIMITS5 (RT Professional).....	262
2.2.2.32	MCP_VARIABLE_LIMITS_EX (RT Professional).....	264
2.2.2.33	MCP_VARIABLE_PROTOCOL (RT Professional).....	266
2.2.2.34	MCP_VARIABLE_PROTOCOL_EX (RT Professional).....	267
2.2.2.35	MCP_VARIABLE_SCALES (RT Professional).....	268
2.2.3	Allgemeine Funktionen (RT Professional).....	269
2.2.3.1	DMActivateRTProject (RT Professional).....	269
2.2.3.2	DMAddNotify (RT Professional).....	270
2.2.3.3	DMChangeDataLocale (RT Professional).....	277
2.2.3.4	DMConnect (RT Professional).....	278
2.2.3.5	DM_NOTIFY_PROC (RT Professional).....	281
2.2.3.6	DMDeactivateRTProject (RT Professional).....	284
2.2.3.7	DMDisconnect (RT Professional).....	285
2.2.3.8	DMEnumNumberFormats (RT Professional).....	286
2.2.3.9	DM_ENUM_FORMATS_PROC (RT Professional).....	288
2.2.3.10	DMEnumUpdateCycles (RT Professional).....	289
2.2.3.11	DM_ENUM_CYCLES_PROC (RT Professional).....	291
2.2.3.12	DMExitWinCC (RT Professional).....	292
2.2.3.13	DMExitWinCCEx (RT Professional).....	293
2.2.3.14	DMFireNotifyData (RT Professional).....	294
2.2.3.15	DMGetConnectionState (RT Professional).....	297
2.2.3.16	DMGetDataLocale (RT Professional).....	298
2.2.3.17	DMGetHotkey (RT Professional).....	299
2.2.3.18	DMGetMachineInfo (RT Professional).....	300
2.2.3.19	DMGetMachineTable (RT Professional).....	301
2.2.3.20	DMRemoveNotify (RT Professional).....	302
2.2.4	Funktionen zur Projektverwaltung (RT Professional).....	305
2.2.4.1	DMEnumOpenedProjects (RT Professional).....	305
2.2.4.2	DM_ENUM_OPENED_PROJECTS_PROC (RT Professional).....	306
2.2.4.3	DMGetProjectDirectory (RT Professional).....	308
2.2.4.4	DMGetProjectInformation (RT Professional).....	309
2.2.4.5	DMGetRuntimeProject (RT Professional).....	311
2.2.4.6	DMOpenProjectDocPlus (RT Professional).....	312
2.2.4.7	DMOpenProjectPlus (RT Professional).....	313
2.2.5	Datentransportkanäle (RT Professional).....	315
2.2.5.1	DMClearBlockQueue (RT Professional).....	315
2.2.5.2	DMEnumDataServices (RT Professional).....	316
2.2.5.3	DM_ENUM_DATA_SERVICE_PROC (RT Professional).....	317
2.2.5.4	DM_DATA_SERVICE_PROC (RT Professional).....	319
2.2.5.5	DMGetNumPendingBlocks (RT Professional).....	320
2.2.5.6	DMInstallDataService (RT Professional).....	321
2.2.5.7	DMSendApplicationData (RT Professional).....	323
2.2.5.8	DMSetBlockQueueSize (RT Professional).....	324
2.2.6	Funktionen zur Bearbeitung von Variablen (RT Professional).....	325
2.2.6.1	DMEnumVarData (RT Professional).....	325
2.2.6.2	DM_ENUM_VARIABLE_PROC (RT Professional).....	327
2.2.6.3	DMEnumVarData4 (RT Professional).....	329
2.2.6.4	DM_ENUM_VARIABLE_PROC4 (RT Professional).....	331
2.2.6.5	DMEnumVarGrpData (RT Professional).....	332
2.2.6.6	DMEnumVarGrpDataExStr (RT Professional).....	334
2.2.6.7	DM_ENUM_VARGRP_PROC (RT Professional).....	336
2.2.6.8	DMEnumVariables (RT Professional).....	337

2.2.6.9	DM_ENUM_VAR_PROC (RT Professional).....	339
2.2.6.10	DMGetValue (RT Professional).....	341
2.2.6.11	DMGetValueEx (RT Professional).....	343
2.2.6.12	DMGetValueExStr (RT Professional).....	344
2.2.6.13	DMGetValueWait (RT Professional).....	353
2.2.6.14	DMGetValueWaitEx (RT Professional).....	356
2.2.6.15	DMGetValueWaitExStr (RT Professional).....	358
2.2.6.16	DMGetVarInfo (RT Professional).....	360
2.2.6.17	DMGetVarInfoExStr (RT Professional).....	362
2.2.6.18	DMGetVarLimits (RT Professional).....	372
2.2.6.19	DMGetVarLimitsExStr (RT Professional).....	373
2.2.6.20	DMGetVarType (RT Professional).....	378
2.2.6.21	DMGetVarTypeExStr (RT Professional).....	380
2.2.6.22	DMSetValue (RT Professional).....	385
2.2.6.23	DMSetValueExStr (RT Professional).....	388
2.2.6.24	DMSetValueMessage (RT Professional).....	391
2.2.6.25	DMSetValueMessageExStr (RT Professional).....	393
2.2.6.26	DMSetValueWait (RT Professional).....	394
2.2.6.27	DMSetValueWaitExStr (RT Professional).....	397
2.2.6.28	DMSetValueWaitMessage (RT Professional).....	400
2.2.6.29	DMSetValueWaitMessageExStr (RT Professional).....	403
2.2.6.30	DM_COMPLETITION_PROC (RT Professional).....	405
2.2.6.31	DMShowVarPropertiesExStr (RT Professional).....	407
2.2.6.32	DMShowVarDatabase (RT Professional).....	410
2.2.6.33	DMShowVarDatabaseExStr (RT Professional).....	412
2.2.6.34	DMShowVarDatabaseMulti (RT Professional).....	415
2.2.6.35	DMShowVarDatabaseMultiExStr (RT Professional).....	417
2.2.6.36	DM_NOTIFY_SELECT_VAR_PROC (RT Professional).....	421
2.2.6.37	GAPICreateNewVariable (RT Professional).....	422
2.2.6.38	GAPICreateNewVariable4 (RT Professional).....	424
2.2.6.39	GAPICreateNewVariable5 (RT Professional).....	425
2.2.6.40	GAPICreateNewVariableEx4 (RT Professional).....	427
2.2.7	Funktionen zur Bearbeitung von strukturierten Variablen (RT Professional).....	429
2.2.7.1	GAPIEnumTypeMembers (RT Professional).....	429
2.2.7.2	DM_ENUM_TPEMEMBERS_PROC (RT Professional).....	430
2.2.7.3	GAPIEnumTypeMembersEx (RT Professional).....	432
2.2.7.4	GAPIEnumTypeMembersExStr (RT Professional).....	433
2.2.7.5	DM_ENUM_TPEMEMBERS_PROC_EX (RT Professional).....	435
2.2.7.6	GAPIEnumTypeMembersEx4 (RT Professional).....	436
2.2.7.7	DM_ENUM_TPEMEMBERS_PROC_EX4 (RT Professional).....	438
2.2.7.8	GAPIEnumTypes (RT Professional).....	439
2.2.7.9	DM_ENUM_TYPES_PROC (RT Professional).....	441
2.2.8	Funktionen zur Bearbeitung von Verbindungen (RT Professional).....	443
2.2.8.1	DMEnumConnectionData (RT Professional).....	443
2.2.8.2	DMEnumConnectionDataExStr (RT Professional).....	444
2.2.8.3	DM_ENUM_CONNECTION_PROC (RT Professional).....	446
2.2.9	Funktionen zur Arbeitsplattform (RT Professional).....	447
2.2.9.1	DMGetOSVersion (RT Professional).....	447
2.2.9.2	DMGetSystemLocale (RT Professional).....	448
2.2.9.3	DMSetLanguage (RT Professional).....	449
2.2.9.4	DMShowLanguageDialog (RT Professional).....	450
2.2.10	Funktionen zur Aktualisierung von Variablen (RT Professional).....	452

2.2.10.1	DMBeginStartVarUpdate (RT Professional).....	452
2.2.10.2	DMEndStartVarUpdate (RT Professional).....	453
2.2.10.3	DMResumeVarUpdate (RT Professional).....	454
2.2.10.4	DMStartVarUpdate (RT Professional).....	456
2.2.10.5	DM_NOTIFY_VARIABLE_PROC (RT Professional).....	458
2.2.10.6	DMStartVarUpdateEx (RT Professional).....	460
2.2.10.7	DMStartVarUpdateExStr (RT Professional).....	462
2.2.10.8	DM_NOTIFY_VARIABLEEX_PROC (RT Professional).....	469
2.2.10.9	DMStopAllUpdates (RT Professional).....	471
2.2.10.10	DMStopVarUpdate (RT Professional).....	472
2.2.10.11	DMSuspendVarUpdate (RT Professional).....	473
2.2.11	Beispiele (RT Professional).....	476
2.2.11.1	Connection to DM (RT Professional).....	476
2.2.11.2	Enum Data of Tags (RT Professional).....	479
2.2.11.3	Enum open projects (RT Professional).....	481
2.2.11.4	Enumerate all structured types (RT Professional).....	483
2.2.11.5	Enumerate all connections (RT Professional).....	485
2.2.11.6	Inquire project informations (RT Professional).....	487
2.2.11.7	OnTestDeactivateRuntimeProject (RT Professional).....	488
2.2.11.8	OnTestEnumGroupsAll (RT Professional).....	488
2.2.11.9	OnTestEnumVariables (RT Professional).....	489
2.2.11.10	OnTestEnumConnectionDataAll (RT Professional).....	490
2.2.11.11	OnTestMachines (RT Professional).....	491
2.2.11.12	OnTestProjectInfo (RT Professional).....	491
2.2.11.13	OnTestProjectPaths (RT Professional).....	493
2.2.11.14	OnTestOpenProject (RT Professional).....	494
2.2.11.15	OnTestOpenProjects (RT Professional).....	495
2.2.11.16	OnTestRuntimeProject (RT Professional).....	495
2.2.11.17	OnTestSystemLocale (RT Professional).....	496
2.2.11.18	OnTestUpdateCycles (RT Professional).....	496
2.2.11.19	OnTestVariablenBeginstartvarupdate (RT Professional).....	497
2.2.11.20	OnTestVariablenEndstartvarupdate (RT Professional).....	498
2.2.11.21	OnTestVariablenGetvalue (RT Professional).....	499
2.2.11.22	OnTestVariablenGetvaluawait (RT Professional).....	502
2.2.11.23	OnTestVariablenGetVarInfo (RT Professional).....	506
2.2.11.24	OnTestVariablenGetvarlimits (RT Professional).....	507
2.2.11.25	OnTestVariablenGetvartype (RT Professional).....	508
2.2.11.26	OnTestVariablenResumevarupdate (RT Professional).....	509
2.2.11.27	OnTestVariablenSetvalue (RT Professional).....	510
2.2.11.28	OnTestVariablenSetvaluawait (RT Professional).....	511
2.2.11.29	OnTestVariablenStopallupdates (RT Professional).....	512
2.2.11.30	OnTestVariablenStopvarupdate (RT Professional).....	512
2.2.11.31	OnTestVariablenSuspendvarupdate (RT Professional).....	513
2.2.11.32	OnTestWinCCShutdown (RT Professional).....	513
2.2.11.33	Open project by means of dialog (RT Professional).....	514
2.2.11.34	Read tag (RT Professional).....	515
2.2.11.35	Write tag (RT Professional).....	520
2.3	Funktionen des Grafiksystems (RT Professional).....	522
2.3.1	Grundlagen (RT Professional).....	522
2.3.1.1	Übersicht über die Funktionen (RT Professional).....	522
2.3.1.2	Übersicht über die Strukturen (RT Professional).....	523
2.3.1.3	Fehlermeldungen (RT Professional).....	523

2.3.1.4	Konstanten (RT Professional).....	525
2.3.1.5	Liste der Objekteigenschaften (A-K) (Graphics Designer) (RT Professional).....	526
2.3.1.6	Liste der Objekteigenschaften (L-Z) (Graphics Designer) (RT Professional).....	538
2.3.1.7	API-Aufrufe in OCXen (RT Professional).....	550
2.3.2	Strukturen (RT Professional).....	551
2.3.2.1	LINKINFO (RT Professional).....	551
2.3.2.2	MULTILINK (RT Professional).....	552
2.3.2.3	MULTILINKINFO (RT Professional).....	553
2.3.2.4	FOCUSINFO (RT Professional).....	554
2.3.3	Allgemeine Funktionen (RT Professional).....	555
2.3.3.1	PDLRTClosePicture (RT Professional).....	555
2.3.3.2	PDLRTDisableClosePicture (RT Professional).....	557
2.3.3.3	PDLRTEnableClosePicture (RT Professional).....	558
2.3.3.4	PDLRTGotoPicture (RT Professional).....	560
2.3.3.5	PDLRTInquireFreeArea (RT Professional).....	561
2.3.3.6	PDLRTOpenPicture (RT Professional).....	563
2.3.3.7	PDLRTPictureNavigation (RT Professional).....	565
2.3.3.8	PDLRTShowApp (RT Professional).....	566
2.3.3.9	PDLRT_CALLBACK (RT Professional).....	568
2.3.4	Funktionen zur Beeinflussung des Runtime-Cursors (RT Professional).....	570
2.3.4.1	PDLRTGetCursorKeys (RT Professional).....	570
2.3.4.2	PDLRTGetFocus (RT Professional).....	572
2.3.4.3	PDLRTSetCursorKeys (RT Professional).....	574
2.3.4.4	PDLRTSetFocus (RT Professional).....	576
2.3.5	Funktionen zum Bearbeiten von Objekteigenschaften (RT Professional).....	578
2.3.5.1	PDLRTGetDefPropEx (RT Professional).....	578
2.3.5.2	PDLRTGetPropEx (RT Professional).....	581
2.3.5.3	PDLRTSetPropEx (RT Professional).....	584
2.3.6	Funktionen zum Bearbeiten von Dynamiken (RT Professional).....	588
2.3.6.1	PDLRTGetLink (RT Professional).....	588
2.3.6.2	PDLRTSetLink (RT Professional).....	590
2.3.6.3	PDLRTSetMultiLink (RT Professional).....	592
2.4	Funktionen des Scripting (RT Professional).....	594
2.4.1	Grundlagen (RT Professional).....	594
2.4.1.1	Übersicht über die Funktionen (RT Professional).....	594
2.4.1.2	Übersicht über die Strukturen (RT Professional).....	595
2.4.1.3	Fehlermeldungen (RT Professional).....	595
2.4.1.4	Konstanten (RT Professional).....	598
2.4.2	Strukturen (RT Professional).....	599
2.4.2.1	AP_ACT_KEY (RT Professional).....	599
2.4.2.2	AP_ACT_RESULT_STRUCT (RT Professional).....	602
2.4.2.3	CREATE_USER_HEADER_FILE (RT Professional).....	603
2.4.2.4	GENERATE_COMPILE (RT Professional).....	604
2.4.2.5	GET_ACTION_STREAM (RT Professional).....	604
2.4.3	Allgemeine Funktionen (RT Professional).....	605
2.4.3.1	APConnect (RT Professional).....	605
2.4.3.2	APDisconnect (RT Professional).....	607
2.4.3.3	APSetLanguage (RT Professional).....	610
2.4.3.4	AP_RT_PROC (RT Professional).....	611
2.4.4	Funktionen zum Bearbeiten von Source Code (RT Professional).....	613
2.4.4.1	APCompile (RT Professional).....	613
2.4.4.2	APCompileEx (RT Professional).....	616

2.4.4.3	GSCGenCompile (RT Professional).....	618
2.4.4.4	GSCGenCompileUserFunctions (RT Professional).....	620
2.4.5	Funktionen zum Bearbeiten von Aktionen (RT Professional).....	621
2.4.5.1	GSCGenGetActionStream (RT Professional).....	621
2.4.6	Funktionen der Aktionsprogrammierung (RT Professional).....	622
2.4.6.1	APActive (RT Professional).....	622
2.4.6.2	APEndAct (RT Professional).....	624
2.4.6.3	APFreeResultStruct (RT Professional).....	626
2.4.6.4	APInactive (RT Professional).....	627
2.4.6.5	APStart (RT Professional).....	629
2.4.6.6	APTransact (RT Professional).....	631
2.4.7	Beispiele (RT Professional).....	633
2.4.7.1	Establish connection to script programming (RT Professional).....	633
2.5	Funktionen der Benutzerverwaltung (RT Professional).....	635
2.5.1	Grundlagen (RT Professional).....	635
2.5.1.1	Übersicht über die Funktionen (RT Professional).....	635
2.5.1.2	Übersicht über die Strukturen (RT Professional).....	636
2.5.1.3	Fehlermeldungen (RT Professional).....	636
2.5.1.4	Konstanten (RT Professional).....	638
2.5.2	Strukturen (RT Professional).....	638
2.5.2.1	PWGEN_GROUPINFO (RT Professional).....	638
2.5.2.2	PWGEN_LEVELINFO (RT Professional).....	639
2.5.2.3	PWGEN_USERINFO (RT Professional).....	639
2.5.3	Allgemeine Funktionen (RT Professional).....	640
2.5.3.1	PWGENConnect (RT Professional).....	640
2.5.3.2	PWGENDisconnect (RT Professional).....	642
2.5.4	Funktionen zum Bearbeiten von Benutzern (RT Professional).....	643
2.5.4.1	PWGENAddUser (RT Professional).....	643
2.5.4.2	PWGENAddUserEx (RT Professional).....	645
2.5.4.3	PWGENChangePassword (RT Professional).....	647
2.5.4.4	PWGENCheckUser (RT Professional).....	648
2.5.4.5	PWGENDeleteUser (RT Professional).....	649
2.5.4.6	PWGENEnumUsers (RT Professional).....	651
2.5.4.7	PWGEN_ENUM_USERS_CALLBACK (RT Professional).....	652
2.5.5	Funktionen zum Bearbeiten von Benutzergruppen (RT Professional).....	654
2.5.5.1	PWGENAddGroup (RT Professional).....	654
2.5.5.2	PWGENEnumGroups (RT Professional).....	655
2.5.5.3	PWGEN_ENUM_GROUPS_CALLBACK (RT Professional).....	657
2.5.6	Funktionen zum Bearbeiten von Berechtigungen (RT Professional).....	658
2.5.6.1	PWGENAddPermLevel (RT Professional).....	658
2.5.6.2	PWGENCheckPermission (RT Professional).....	659
2.5.6.3	PWGENDeletePermLevel (RT Professional).....	661
2.5.6.4	PWGENEnumPermLevels (RT Professional).....	662
2.5.6.5	PWGEN_ENUM_LEVELS_CALLBACK (RT Professional).....	663
2.5.6.6	PWGENReadUserPerm (RT Professional).....	665
2.5.6.7	PWRTCheckPermission (RT Professional).....	666
2.5.6.8	PWRTCheckPermissionOnPicture (RT Professional).....	667
2.5.6.9	PWRTPermissionLevelDialog (RT Professional).....	668
2.5.6.10	PWRTPermissionLevelDialogEx (RT Professional).....	669
2.5.6.11	PWRTPermissionToString (RT Professional).....	670
2.5.7	Funktionen zum Anmelden, Abmelden (RT Professional).....	671
2.5.7.1	PWRTGetCurrentUser (RT Professional).....	671

2.5.7.2	PWRTGetLoginPriority (RT Professional).....	672
2.5.7.3	PWRTIsLoggedInByCard (RT Professional).....	673
2.5.7.4	PWRTLogin (RT Professional).....	674
2.5.7.5	PWRTLogout (RT Professional).....	676
2.5.7.6	PWRTLogoutEx (RT Professional).....	677
2.5.7.7	PWRTSilentLogin (RT Professional).....	678
2.5.7.8	PWRTSilentLoginEx (RT Professional).....	679
2.5.8	Beispiele (RT Professional).....	681
2.5.8.1	PWRT check permission (RT Professional).....	681
2.5.8.2	Checks admission of a certain level for a picture (RT Professional).....	682
2.5.8.3	Gets a string associated with the permission number (RT Professional).....	683
2.5.8.4	Permission level query through a dialog with specifying a possible error (RT Professional).....	684
2.5.8.5	Return the name of the current user (RT Professional).....	685
2.5.8.6	Queries the current login priority (RT Professional).....	686
2.5.8.7	Checks for the user has been logged on by card (RT Professional).....	686
2.5.8.8	PWRT login - dialog provided by WinCC itself (RT Professional).....	687
2.5.8.9	PWRT logout (RT Professional).....	688
2.5.8.10	Silent logout with priority level (RT Professional).....	689
2.5.8.11	Login without using a dialog (RT Professional).....	690
2.5.8.12	Silent login with priority level (RT Professional).....	691
2.6	Funktionen des Textsystems (RT Professional).....	692
2.6.1	Grundlagen (RT Professional).....	692
2.6.1.1	Übersicht über die Funktionen (RT Professional).....	692
2.6.1.2	Fehlermeldungen (RT Professional).....	692
2.6.1.3	Sprachcodes (RT Professional).....	694
2.6.2	Allgemeine Funktionen (RT Professional).....	695
2.6.2.1	TXTCloseProject (RT Professional).....	695
2.6.2.2	TXTGetMaxTextID (RT Professional).....	697
2.6.2.3	TXTOpenProject (RT Professional).....	698
2.6.2.4	TXTRTConnect (RT Professional).....	700
2.6.2.5	TXTRTDisconnect (RT Professional).....	701
2.6.3	Funktionen zur Bearbeitung von Hilfetexten (RT Professional).....	702
2.6.3.1	TXTEnumInfoText (RT Professional).....	702
2.6.3.2	TXT_ENUM_INFOTEXTS_PROC (RT Professional).....	704
2.6.3.3	TXTUpdateRuntime (RT Professional).....	705
2.6.3.4	TXTRTGetInfoText (RT Professional).....	707
2.6.3.5	TXTRTGetInfoTextMC (RT Professional).....	708
2.6.4	Funktionen zur Bearbeitung von Sprachen (RT Professional).....	710
2.6.4.1	TXTEnumLanguages (RT Professional).....	710
2.6.4.2	TXT_ENUM_LANGUAGES_PROC (RT Professional).....	712
2.6.4.3	TXTGetFont (RT Professional).....	713
2.6.4.4	TXTShowLanguagesDialog (RT Professional).....	715
2.6.4.5	TXTRTGetLanguageID (RT Professional).....	716
2.6.4.6	TXTRTSetLanguage (RT Professional).....	718
2.6.5	Beispiele (RT Professional).....	719
2.6.5.1	Get Helptext (RT Professional).....	719
2.6.5.2	Enumerate Helptexts (RT Professional).....	721
2.7	Funktionen des Protokollsystems (RT Professional).....	723
2.7.1	Grundlagen (RT Professional).....	723
2.7.1.1	Übersicht über die Funktionen (RT Professional).....	723

2.7.1.2	Konstanten (RT Professional).....	724
2.7.1.3	Fehlermeldungen (RT Professional).....	726
2.7.1.4	Liste der Objekteigenschaften (RT Professional).....	727
2.7.1.5	Allgemeine Vorgehensweise beim Bearbeiten von Druckauftragseigenschaften (Report Designer) (RT Professional).....	732
2.7.2	Funktionen zum Verbindungsaufbau (RT Professional).....	736
2.7.2.1	RPJAttach (RT Professional).....	736
2.7.2.2	RPJDetach (RT Professional).....	737
2.7.2.3	RPJMemFree (RT Professional).....	738
2.7.3	Funktionen zum Bearbeiten von Projekteigenschaften (RT Professional).....	739
2.7.3.1	RPJGetNumProjectProperties (RT Professional).....	739
2.7.3.2	RPJGetProjectPropertyAt (RT Professional).....	740
2.7.3.3	RPJGetProjectProperty (RT Professional).....	741
2.7.3.4	RPJProjectLock (RT Professional).....	742
2.7.3.5	RPJProjectUnlock (RT Professional).....	744
2.7.3.6	RPJProjectUnlockAll (RT Professional).....	745
2.7.4	Funktionen zum Bearbeiten von Druckaufträgen (RT Professional).....	746
2.7.4.1	RPJCreateJob (RT Professional).....	746
2.7.4.2	RPJDeleteJob (RT Professional).....	747
2.7.4.3	RPJCreatePropertyHandle (RT Professional).....	749
2.7.4.4	RPJDeletePropertyHandle (RT Professional).....	750
2.7.4.5	RPJGetJobNameAt (RT Professional).....	752
2.7.4.6	RPJGetNumJobs (RT Professional).....	753
2.7.4.7	RPJJobLock (RT Professional).....	754
2.7.4.8	RPJJobUnlock (RT Professional).....	756
2.7.4.9	RPJJobUnlockAll (RT Professional).....	757
2.7.5	Funktionen zum Bearbeiten von Druckauftragsmethoden (RT Professional).....	759
2.7.5.1	RPJCallJobMethod (RT Professional).....	759
2.7.5.2	RPJGetJobMethodAt (RT Professional).....	760
2.7.5.3	RPJGetNumJobMethods (RT Professional).....	761
2.7.6	Funktionen zum Bearbeiten von Druckauftragseigenschaften (RT Professional).....	762
2.7.6.1	RPJGetJobPropertyAt (RT Professional).....	762
2.7.6.2	RPJGetJobProps (RT Professional).....	764
2.7.6.3	RPJGetNumJobProperties (RT Professional).....	765
2.7.6.4	RPJGetProperty (RT Professional).....	766
2.7.6.5	RPJPropertyClear (RT Professional).....	769
2.7.6.6	RPJSetProperty (RT Professional).....	771
2.7.7	Beispiele (RT Professional).....	774
2.7.7.1	Get print job methodname (RT Professional).....	774
2.7.7.2	Get print job names (RT Professional).....	777
2.7.7.3	Get print job properties (RT Professional).....	780
2.7.7.4	Modify print job properties (RT Professional).....	783
2.7.7.5	Show print job preview (RT Professional).....	788
2.8	Funktionen für Variablen und Archive (RT Professional).....	791
2.8.1	Grundlagen (RT Professional).....	791
2.8.1.1	Übersicht über die Funktionen (RT Professional).....	791
2.8.1.2	Übersicht über die Strukturen (RT Professional).....	792
2.8.1.3	Fehlermeldungen (RT Professional).....	793
2.8.1.4	Konstanten (RT Professional).....	795
2.8.2	Strukturen (RT Professional).....	805
2.8.2.1	TLG_ARCHIV_STR (RT Professional).....	805
2.8.2.2	TLG_ARCHIVDATARAW (RT Professional).....	808

2.8.2.3	TLG_BACKUP_TABLE_INFO (RT Professional).....	811
2.8.2.4	TLG_CURVESCALEX (RT Professional).....	813
2.8.2.5	TLG_CURVESCALEY (RT Professional).....	818
2.8.2.6	TLG_GETARCHIVDATA (RT Professional).....	822
2.8.2.7	TLG_IO_BACKUP_SELECT (RT Professional).....	823
2.8.2.8	TLG_PROT_CURVE_INFOS (RT Professional).....	824
2.8.2.9	TLG_SCAL_STR (RT Professional).....	826
2.8.2.10	TLG_TABLE_INFO (RT Professional).....	827
2.8.2.11	TLG_TABLESCALE (RT Professional).....	828
2.8.2.12	TLG_TEMPLATEITEM_INFO (RT Professional).....	831
2.8.2.13	TLG_TIME_STR (RT Professional).....	833
2.8.2.14	TLG_TIMEDATA (RT Professional).....	834
2.8.2.15	TLG_TPLITEM_CURVE (RT Professional).....	835
2.8.2.16	TLG_TPLITEM_INFO (RT Professional).....	836
2.8.2.17	TLG_TPLITEM_TABLE (RT Professional).....	837
2.8.2.18	TLG_VAR_STR (RT Professional).....	838
2.8.2.19	TLG_VARIABLE_INFO (RT Professional).....	844
2.8.3	Allgemeine Funktionen (RT Professional).....	845
2.8.3.1	TLGCSCConnect (RT Professional).....	845
2.8.3.2	TLGCSCConnectEx (RT Professional).....	846
2.8.3.3	TLGCSDisConnect (RT Professional).....	847
2.8.3.4	TLGChangeLanguage (RT Professional).....	849
2.8.3.5	TLGConnect (RT Professional).....	850
2.8.3.6	TLGDisconnect (RT Professional).....	851
2.8.4	Projektverwaltungsfunktionen (RT Professional).....	853
2.8.4.1	TLGCloseProject (RT Professional).....	853
2.8.4.2	TLGOpenProject (RT Professional).....	854
2.8.4.3	TLGEnumProject (RT Professional).....	856
2.8.4.4	TLG_ENUM_PROJECT_NAME_CALLBACK (RT Professional).....	857
2.8.5	Funktionen zum Bearbeiten von Variablen (RT Professional).....	858
2.8.5.1	TLGEnumVariables (RT Professional).....	858
2.8.5.2	TLG_ENUM_VARIABLE_NAME_CALLBACK (RT Professional).....	860
2.8.5.3	TLGEnumVariablesEx (RT Professional).....	861
2.8.5.4	TLG_ENUMVARIABLES (RT Professional).....	863
2.8.5.5	TLGReadVariable (RT Professional).....	864
2.8.6	Funktionen zum Bearbeiten von Archiven (RT Professional).....	866
2.8.6.1	TLGEnumArchives (RT Professional).....	866
2.8.6.2	TLG_ENUM_ARCHIV_CALLBACK (RT Professional).....	867
2.8.6.3	TLGEnumArchiv (RT Professional).....	868
2.8.6.4	TLGEnumArchivEx (RT Professional).....	870
2.8.6.5	TLGEnumArchivSel (RT Professional).....	872
2.8.6.6	TLG_ENUMTABLES (RT Professional).....	874
2.8.6.7	TLGFreeMemory (RT Professional).....	876
2.8.6.8	TLGGetArchivData (RT Professional).....	877
2.8.6.9	TLG_GETARCHIVDATA_CALLBACK (RT Professional).....	879
2.8.6.10	TLGGetArchivDataEx (RT Professional).....	881
2.8.6.11	TLGGetClosestTime (RT Professional).....	883
2.8.6.12	TLGGetClosestTimeEx (RT Professional).....	885
2.8.6.13	TLGInsertArchivData (RT Professional).....	887
2.8.6.14	TLGLockArchiv (RT Professional).....	890
2.8.6.15	TLGLockVariable (RT Professional).....	891
2.8.6.16	TLGReadArchiv (RT Professional).....	893

2.8.7	Funktionen zur Kurven- und Tabellendarstellung (RT Professional).....	894
2.8.7.1	TLGCloseWindow (RT Professional).....	894
2.8.7.2	TLGDrawCurvesInDC (RT Professional).....	895
2.8.7.3	TLGInsertTemplatItem (RT Professional).....	897
2.8.7.4	TLGPressToolBarButton (RT Professional).....	898
2.8.7.5	TLGSetRulerWindowVisible (RT Professional).....	900
2.8.7.6	TLGShowWindow (RT Professional).....	901
2.8.8	Funktionen zum Bearbeiten von Zeitsystemen (RT Professional).....	902
2.8.8.1	TLGEnumTime (RT Professional).....	902
2.8.8.2	TLG_ENUM_TIME_NAME_CALLBACK (RT Professional).....	904
2.8.8.3	TLGEnumTimes (RT Professional).....	905
2.8.8.4	TLG_ENUMTIMES_CALLBACK (RT Professional).....	907
2.8.8.5	TLGReadTime (RT Professional).....	908
2.8.9	Funktionen zum Sichern und Wiederherstellen (RT Professional).....	909
2.8.9.1	TLGEnumBackupEntries (RT Professional).....	909
2.8.9.2	TLG_ENUMBACKUP_ENTRIES (RT Professional).....	911
2.8.9.3	TLGExport (RT Professional).....	912
2.8.9.4	TLGGetBackupSize (RT Professional).....	914
2.8.10	Beispiele (RT Professional).....	916
2.8.10.1	Edit curve template - Example 1 (RT Professional).....	916
2.8.10.2	Enum all acquisition and archiving times (RT Professional).....	919
2.8.10.3	Enum archives (RT Professional).....	922
2.8.10.4	Enum variables of an archive (RT Professional).....	925
2.8.10.5	Enumerate archives (RT Professional).....	927
2.8.10.6	Read archive (RT Professional).....	929
2.8.10.7	Read parameters of time object (RT Professional).....	933
2.9	Funktionen für Rezepturen (RT Professional).....	935
2.9.1	Grundlagen (RT Professional).....	935
2.9.1.1	Übersicht über die Funktionen (RT Professional).....	935
2.9.1.2	Aufbau von Rezepturen (RT Professional).....	937
2.9.1.3	Abhängigkeiten Aufrufreihenfolge API-Funktionen (RT Professional).....	939
2.9.1.4	Fehlermeldungen (RT Professional).....	940
2.9.1.5	Konstanten (RT Professional).....	941
2.9.2	Allgemeine Funktionen (RT Professional).....	943
2.9.2.1	uaGetLastError (RT Professional).....	943
2.9.2.2	uaIsActive (RT Professional).....	945
2.9.2.3	uaUsers (RT Professional).....	946
2.9.2.4	uaSetLocalEvents (RT Professional).....	947
2.9.3	Funktionen zum Verbindungsaufbau (RT Professional).....	948
2.9.3.1	uaConnect (RT Professional).....	948
2.9.3.2	uaDisconnect (RT Professional).....	949
2.9.3.3	uaQueryArchive (RT Professional).....	950
2.9.3.4	uaQueryArchiveByName (RT Professional).....	952
2.9.3.5	uaQueryConfiguration (RT Professional).....	954
2.9.3.6	uaReleaseArchive (RT Professional).....	955
2.9.3.7	uaReleaseConfiguration (RT Professional).....	957
2.9.4	Funktionen zum Bearbeiten von Rezepturen (RT Professional).....	958
2.9.4.1	uaArchiveClose (RT Professional).....	958
2.9.4.2	uaArchiveDelete (RT Professional).....	959
2.9.4.3	uaArchiveExport (RT Professional).....	960
2.9.4.4	uaArchiveGetID (RT Professional).....	962
2.9.4.5	uaArchiveGetName (RT Professional).....	963

2.9.4.6	uaArchiveGetSort (RT Professional).....	964
2.9.4.7	uaArchiveImport (RT Professional).....	965
2.9.4.8	uaArchiveOpen (RT Professional).....	967
2.9.4.9	uaArchiveUpdate (RT Professional).....	968
2.9.4.10	uaGetArchive (RT Professional).....	969
2.9.4.11	uaGetNumArchives (RT Professional).....	970
2.9.4.12	uaOpenArchives (RT Professional).....	971
2.9.5	Funktionen zum Bearbeiten von Rezepturelementen (RT Professional).....	972
2.9.5.1	uaArchiveGetCount (RT Professional).....	972
2.9.5.2	uaArchiveGetFilter (RT Professional).....	973
2.9.5.3	uaArchiveInsert (RT Professional).....	974
2.9.5.4	uaArchiveMoveFirst (RT Professional).....	975
2.9.5.5	uaArchiveMoveLast (RT Professional).....	976
2.9.5.6	uaArchiveMoveNext (RT Professional).....	977
2.9.5.7	uaArchiveMovePrevious (RT Professional).....	978
2.9.5.8	uaGetNumFields (RT Professional).....	979
2.9.6	Funktionen zum Bearbeiten von Feldern (RT Professional).....	980
2.9.6.1	uaArchiveGetFieldLength (RT Professional).....	980
2.9.6.2	uaArchiveGetFieldName (RT Professional).....	981
2.9.6.3	uaArchiveGetFields (RT Professional).....	983
2.9.6.4	uaArchiveGetFieldType (RT Professional).....	984
2.9.6.5	uaArchiveGetFieldValueDate (RT Professional).....	985
2.9.6.6	uaArchiveGetFieldValueDouble (RT Professional).....	986
2.9.6.7	uaArchiveGetFieldValueLong (RT Professional).....	987
2.9.6.8	uaArchiveGetFieldValueString (RT Professional).....	988
2.9.6.9	uaArchiveSetFieldValueDate (RT Professional).....	990
2.9.6.10	uaArchiveSetFieldValueDouble (RT Professional).....	991
2.9.6.11	uaArchiveSetFieldValueLong (RT Professional).....	992
2.9.6.12	uaArchiveSetFieldValueString (RT Professional).....	993
2.9.6.13	uaGetField (RT Professional).....	995
2.9.7	Funktionen zum Filtern und Sortieren (RT Professional).....	996
2.9.7.1	uaArchiveRequery (RT Professional).....	996
2.9.7.2	uaArchiveSetFilter (RT Professional).....	997
2.9.7.3	uaArchiveSetSort (RT Professional).....	999
2.9.8	Funktionen zum Bearbeiten von Rezeptursichten (RT Professional).....	1000
2.9.8.1	uaOpenViews (RT Professional).....	1000
2.9.9	Funktionen zum Bearbeiten von Variablen (RT Professional).....	1001
2.9.9.1	uaArchiveReadTagValues (RT Professional).....	1001
2.9.9.2	uaArchiveReadTagValuesByName (RT Professional).....	1002
2.9.9.3	uaArchiveWriteTagValues (RT Professional).....	1003
2.9.9.4	uaArchiveWriteTagValuesByName (RT Professional).....	1005
2.10	Funktionen für Meldungen (RT Professional).....	1006
2.10.1	Grundlagen (RT Professional).....	1006
2.10.1.1	Übersicht über die Funktionen (RT Professional).....	1006
2.10.1.2	Übersicht über die Strukturen (RT Professional).....	1007
2.10.1.3	Fehlermeldungen (RT Professional).....	1008
2.10.1.4	Konstanten (RT Professional).....	1010
2.10.2	Strukturen (RT Professional).....	1022
2.10.2.1	MSG_RTCREATE_STRUCT_PLUS (RT Professional).....	1022
2.10.2.2	MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (RT Professional).....	1024
2.10.2.3	MSG_FILTER_STRUCT_PLUS (RT Professional).....	1027
2.10.2.4	MSG_RTDATA_INSTANCE_STRUCT_PLUS (RT Professional).....	1031

2.10.2.5	MSG_CLASS_STRUCT_PLUS (RT Professional).....	1034
2.10.2.6	MSG_CSDATA_STRUCT_PLUS (RT Professional).....	1037
2.10.2.7	MSG_RTGROUPENUM_STRUCT_PLUS (RT Professional).....	1039
2.10.2.8	MSG_RTGROUPSET_STRUCT_PLUS (RT Professional).....	1040
2.10.2.9	MSG_COMMENT_INSTANCE_STRUCT_PLUS (RT Professional).....	1041
2.10.2.10	MSG_INFOTEXT_STRUCT_PLUS (RT Professional).....	1043
2.10.2.11	MSG_BACKUP_STRUCT_PLUS (RT Professional).....	1044
2.10.2.12	MSG_RTLOCK_STRUCT_PLUS (RT Professional).....	1045
2.10.2.13	MSG_TEXTVAL256_STRUCT_PLUS (RT Professional).....	1046
2.10.3	Allgemeine Funktionen (RT Professional).....	1047
2.10.3.1	MSRTEnumArchivDataPlus (RT Professional).....	1047
2.10.3.2	MSRTStartMsgServicePlus (RT Professional).....	1049
2.10.3.3	MSRTStopMsgServicePlus (RT Professional).....	1053
2.10.3.4	MSRTQuitHornPlus (RT Professional).....	1054
2.10.3.5	MSG_SERVICE_NOTIFY_PROCPlus (RT Professional).....	1055
2.10.3.6	MSRTWebClientPlus (RT Professional).....	1059
2.10.4	Funktionen zum Bearbeiten von Protokollen (RT Professional).....	1059
2.10.4.1	MSRTActivateMProtPlus (RT Professional).....	1059
2.10.4.2	MSRTEnumProtDataPlus (RT Professional).....	1060
2.10.4.3	MSRTPrintMProtPlus (RT Professional).....	1062
2.10.5	Funktionen zum Bearbeiten von Meldungen (RT Professional).....	1063
2.10.5.1	MSRTEnumArchivInstancePlus (RT Professional).....	1063
2.10.5.2	MSRTCreateMsgInstanceWithCommentPlus (RT Professional).....	1064
2.10.5.3	MSRTCreateMsgInstancePlus (RT Professional).....	1066
2.10.5.4	MSRTDialogMsgLockPlus (RT Professional).....	1068
2.10.5.5	MSRTCreateMsgPlus (RT Professional).....	1069
2.10.5.6	MSRTEnumLockedMsgPlus (RT Professional).....	1071
2.10.5.7	MSRTEnumMsgRTDataPlus (RT Professional).....	1072
2.10.5.8	MSRTGetClassInfoPlus (RT Professional).....	1073
2.10.5.9	MSRTGetLastMsgWithCommentPlus (RT Professional).....	1075
2.10.5.10	MSRTGetMsgActualPlus (RT Professional).....	1077
2.10.5.11	MSRTGetMsgCSDDataPlus (RT Professional).....	1078
2.10.5.12	MSRTGetMsgPriorityPlus (RT Professional).....	1080
2.10.5.13	MSRTGetMsgQuitPlus (RT Professional).....	1081
2.10.5.14	MSRTGetMsgTextPlus (RT Professional).....	1083
2.10.5.15	MSRTGetSelectedMsgPlus (RT Professional).....	1084
2.10.5.16	MSRTResetMsgPlus (RT Professional).....	1086
2.10.5.17	MSRTLoopInAlarmPlus (RT Professional).....	1088
2.10.6	Funktionen zum Bearbeiten von Meldegruppen (RT Professional).....	1089
2.10.6.1	MSRTEnumGroupMsgPlus (RT Professional).....	1089
2.10.6.2	MSRTLockGroupPlus (RT Professional).....	1091
2.10.6.3	MSRTQuitGroupPlus (RT Professional).....	1092
2.10.7	Funktionen zum Bearbeiten von Meldefiltern (RT Professional).....	1094
2.10.7.1	MSRTGetFilterDataPlus (RT Professional).....	1094
2.10.7.2	MSRTCheckWinFilterPlus (RT Professional).....	1095
2.10.7.3	MSRTSetMsgFilterPlus (RT Professional).....	1097
2.10.7.4	MSRTSetMsgWinFilterPlus (RT Professional).....	1098
2.10.8	Funktionen zum Bearbeiten von Meldeanzeige (RT Professional).....	1100
2.10.8.1	MSRTMsgWinCommandPlus (RT Professional).....	1100
2.10.9	Funktionen zum Bearbeiten von Kommentaren (RT Professional).....	1102
2.10.9.1	MSRTGetCommentInstancePlus (RT Professional).....	1102
2.10.9.2	MSRTSetCommentInstancePlus (RT Professional).....	1103

2.10.10	Funktionen zum Bearbeiten von Infotexten (RT Professional).....	1105
2.10.10.1	MSRTGetInfotextPlus (RT Professional).....	1105
2.10.10.2	MSRTSetInfotextPlus (RT Professional).....	1106
2.10.11	Archivierungsfunktionen (RT Professional).....	1108
2.10.11.1	MSRTEnumBackupListPlus (RT Professional).....	1108
2.10.11.2	MSRTExportPlus (RT Professional).....	1109
2.11	Funktionen zur Anzeige von PLC-Code (RT Professional).....	1111
2.11.1	Anzeige in STEP 7 (RT Professional).....	1111
2.11.1.1	Grundlagen (RT Professional).....	1111
2.11.1.2	OpenTIAPortalProject (RT Professional).....	1112
2.11.1.3	OpenTIAPortalIECPLByCall (RT Professional).....	1114
2.11.1.4	OpenTIAPortalIECPLByAssignment (RT Professional).....	1116
2.11.1.5	OpenTIAPortalS7GraphByBlock (RT Professional).....	1118
2.11.1.6	Beispiel: Einbinden in eine WinCC Funktion (RT Professional).....	1121
2.11.2	Anzeige in der PLC-Code-Anzeige (RT Professional).....	1122
2.11.2.1	Grundlagen (RT Professional).....	1122
2.11.2.2	OpenViewerS7GraphByBlock (RT Professional).....	1123
2.11.2.3	OpenViewerIECPLByCall (RT Professional).....	1125
2.11.2.4	OpenViewerIECPLByFCCall (RT Professional).....	1129
2.11.2.5	OpenViewerIECPLByAssignment (RT Professional).....	1132
2.11.3	Fehlerbehandlung (RT Professional).....	1134
2.12	Fehlerbehandlung (RT Professional).....	1136
2.12.1	CMN_ERROR (RT Professional).....	1136
2.12.2	CCStorageError.h (RT Professional).....	1137
3	VB-Scripting (Panels, Comfort Panels, RT Advanced, RT Professional).....	1149
3.1	Systemfunktionen (Panels, Comfort Panels, RT Advanced).....	1149
3.1.1	AcknowledgeAlarm (Panels, Comfort Panels, RT Advanced).....	1149
3.1.2	ActivatePreviousScreen (Panels, Comfort Panels, RT Advanced).....	1149
3.1.3	ActivateScreen (Panels, Comfort Panels, RT Advanced).....	1150
3.1.4	ActivateScreenByNumber (Panels, Comfort Panels, RT Advanced).....	1151
3.1.5	ActivateSystemDiagnosticsView (Panels, Comfort Panels, RT Advanced).....	1152
3.1.6	ArchiveLogFile (Panels, Comfort Panels, RT Advanced).....	1153
3.1.7	BackupRAMFileSystem (Panels, Comfort Panels, RT Advanced).....	1155
3.1.8	CalibrateTouchScreen (Panels, Comfort Panels, RT Advanced).....	1155
3.1.9	ChangeConnection (Panels, Comfort Panels, RT Advanced).....	1156
3.1.10	ClearAlarmBuffer (Panels, Comfort Panels, RT Advanced).....	1158
3.1.11	ClearAlarmBufferProtool (Panels, Comfort Panels, RT Advanced).....	1159
3.1.12	ClearDataRecord (Panels, Comfort Panels, RT Advanced).....	1159
3.1.13	ClearDataRecordMemory (Panels, Comfort Panels, RT Advanced).....	1161
3.1.14	ClearLog (Panels, Comfort Panels, RT Advanced).....	1162
3.1.15	CloseAllLogs (Panels, Comfort Panels, RT Advanced).....	1162
3.1.16	ControlSmartServer (Panels, Comfort Panels, RT Advanced).....	1163
3.1.17	ControlWebServer (Panels, Comfort Panels, RT Advanced).....	1164
3.1.18	CopyLog (Panels, Comfort Panels, RT Advanced).....	1165
3.1.19	DecreaseTag (Panels, Comfort Panels, RT Advanced).....	1166
3.1.20	EditAlarm (Panels, Comfort Panels, RT Advanced).....	1167
3.1.21	Encode (Panels, Comfort Panels, RT Advanced).....	1167
3.1.22	EncodeEx (Panels, Comfort Panels, RT Advanced).....	1168
3.1.23	EstablishPROFIsafe (Panels, Comfort Panels, RT Advanced).....	1169
3.1.24	ExportDataRecords (Panels, Comfort Panels, RT Advanced).....	1170

3.1.25	ExportDataRecordsWithChecksum (Panels, Comfort Panels, RT Advanced).....	1172
3.1.26	ExportImportUserAdministration (Panels, Comfort Panels, RT Advanced).....	1175
3.1.27	GetBrightness (Panels, Comfort Panels, RT Advanced).....	1176
3.1.28	GetDataRecordFromPLC (Panels, Comfort Panels, RT Advanced).....	1176
3.1.29	GetDataRecordName (Panels, Comfort Panels, RT Advanced).....	1178
3.1.30	GetDataRecordTagsFromPLC (Panels, Comfort Panels, RT Advanced).....	1180
3.1.31	GetGroupNumber (Panels, Comfort Panels, RT Advanced).....	1180
3.1.32	GetPassword (Panels, Comfort Panels, RT Advanced).....	1181
3.1.33	GetUserName (Panels, Comfort Panels, RT Advanced).....	1182
3.1.34	GoToEnd (Panels, Comfort Panels, RT Advanced).....	1182
3.1.35	GoToHome (Panels, Comfort Panels, RT Advanced).....	1183
3.1.36	ImportDataRecords (Panels, Comfort Panels, RT Advanced).....	1183
3.1.37	ImportDataRecordsWithChecksum (Panels, Comfort Panels, RT Advanced).....	1185
3.1.38	IncreaseTag (Panels, Comfort Panels, RT Advanced).....	1187
3.1.39	InverseLinearScaling (Panels, Comfort Panels, RT Advanced).....	1187
3.1.40	InvertBit (Panels, Comfort Panels, RT Advanced).....	1189
3.1.41	InvertBitInTag (Panels, Comfort Panels, RT Advanced).....	1190
3.1.42	LinearScaling (Panels, Comfort Panels, RT Advanced).....	1191
3.1.43	LoadDataRecord (Panels, Comfort Panels, RT Advanced).....	1192
3.1.44	Logoff (Panels, Comfort Panels, RT Advanced).....	1193
3.1.45	Logon (Panels, Comfort Panels, RT Advanced).....	1194
3.1.46	LookupText (Panels, Comfort Panels, RT Advanced).....	1194
3.1.47	NotifyUserAction (Panels, Comfort Panels, RT Advanced).....	1195
3.1.48	OpenAllLogs (Panels, Comfort Panels, RT Advanced).....	1197
3.1.49	OpenCommandPrompt (Panels, Comfort Panels, RT Advanced).....	1198
3.1.50	OpenControlPanel (Panels, Comfort Panels, RT Advanced).....	1198
3.1.51	OpenInternetExplorer (Panels, Comfort Panels, RT Advanced).....	1199
3.1.52	OpenScreenKeyboard (Panels, Comfort Panels, RT Advanced).....	1200
3.1.53	OpenTaskManager (Panels, Comfort Panels, RT Advanced).....	1201
3.1.54	PageDown (Panels, Comfort Panels, RT Advanced).....	1201
3.1.55	PageUp (Panels, Comfort Panels, RT Advanced).....	1202
3.1.56	PrintReport (Panels, Comfort Panels, RT Advanced).....	1203
3.1.57	PrintScreen (Panels, Comfort Panels, RT Advanced).....	1203
3.1.58	ResetBit (Panels, Comfort Panels, RT Advanced).....	1204
3.1.59	ResetBitInTag (Panels, Comfort Panels, RT Advanced).....	1205
3.1.60	SafelyRemoveHardware (Panels, Comfort Panels).....	1206
3.1.61	SaveDataRecord (Panels, Comfort Panels, RT Advanced).....	1207
3.1.62	SendEMail (Panels, Comfort Panels, RT Advanced).....	1208
3.1.63	SetAcousticSignal (Panels, Comfort Panels, RT Advanced).....	1209
3.1.64	SetAlarmReportMode (Panels, Comfort Panels, RT Advanced).....	1210
3.1.65	SetAndGetBrightness (Panels, Comfort Panels, RT Advanced).....	1211
3.1.66	SetBit (Panels, Comfort Panels, RT Advanced).....	1211
3.1.67	SetBitInTag (Panels, Comfort Panels, RT Advanced).....	1212
3.1.68	SetBrightness (Panels, Comfort Panels, RT Advanced).....	1214
3.1.69	SetConnectionMode (Panels, Comfort Panels, RT Advanced).....	1215
3.1.70	SetDataRecordTagsToPLC (Panels, Comfort Panels, RT Advanced).....	1216
3.1.71	SetDataRecordToPLC (Panels, Comfort Panels, RT Advanced).....	1217
3.1.72	SetDaylightSavingTime (Panels, Comfort Panels, RT Advanced).....	1218
3.1.73	SetDeviceMode (Panels, Comfort Panels, RT Advanced).....	1219
3.1.74	SetDisplayMode (Panels, Comfort Panels, RT Advanced).....	1220
3.1.75	SetLanguage (Panels, Comfort Panels, RT Advanced).....	1220
3.1.76	SetPLCDateTime (Panels, Comfort Panels, RT Advanced).....	1221

3.1.77	SetRecipeTags (Panels, Comfort Panels, RT Advanced).....	1222
3.1.78	SetScreenKeyboardMode (Panels, Comfort Panels, RT Advanced).....	1223
3.1.79	SetTag (Panels, Comfort Panels, RT Advanced).....	1224
3.1.80	ShiftAndMask (Panels, Comfort Panels, RT Advanced).....	1225
3.1.81	ShowAlarmWindow (Panels, Comfort Panels, RT Advanced).....	1227
3.1.82	ShowOperatorNotes (Panels, Comfort Panels, RT Advanced).....	1228
3.1.83	ShowPopupScreen (Panels, Comfort Panels, RT Advanced).....	1229
3.1.84	ShowSlideInScreen (Panels, Comfort Panels, RT Advanced).....	1230
3.1.85	ShowSoftwareVersion (Panels, Comfort Panels, RT Advanced).....	1230
3.1.86	ShowSystemAlarm (Panels, Comfort Panels, RT Advanced).....	1231
3.1.87	ShowSystemDiagnosticsWindow (Panels, Comfort Panels).....	1232
3.1.88	StartLogging (Panels, Comfort Panels, RT Advanced).....	1232
3.1.89	StartNextLog (Panels, Comfort Panels, RT Advanced).....	1233
3.1.90	StartProgram (Panels, Comfort Panels, RT Advanced).....	1234
3.1.91	StopLogging (Panels, Comfort Panels, RT Advanced).....	1235
3.1.92	StopRuntime (Panels, Comfort Panels, RT Advanced).....	1236
3.1.93	TerminatePROFIsafe (Panels, Comfort Panels, RT Advanced).....	1237
3.1.94	WinACMPGetVersion (Panels, Comfort Panels).....	1238
3.1.95	WinACMPSetStartAtBoot (Panels, Comfort Panels).....	1238
3.1.96	WinACSetStartMode (Panels, Comfort Panels).....	1239
3.2	Systemfunktionen (RT Professional).....	1240
3.2.1	ActivateScreen (RT Professional).....	1240
3.2.2	ActivateScreenInScreenWindow (RT Professional).....	1241
3.2.3	DecreaseTag (RT Professional).....	1242
3.2.4	ExportImportUserAdministration (RT Professional).....	1243
3.2.5	IncreaseTag (RT Professional).....	1244
3.2.6	InverseLinearScaling (RT Professional).....	1245
3.2.7	InvertBit (RT Professional).....	1247
3.2.8	InvertBitInTag (RT Professional).....	1248
3.2.9	LinearScaling (RT Professional).....	1249
3.2.10	ResetBit (RT Professional).....	1250
3.2.11	ResetBitInTag (RT Professional).....	1251
3.2.12	SetBit (RT Professional).....	1252
3.2.13	SetBitInTag (RT Professional).....	1253
3.2.14	SetPropertyByConstant (RT Professional).....	1255
3.2.15	SetPropertyByProperty (RT Professional).....	1256
3.2.16	SetPropertyByTag (RT Professional).....	1258
3.2.17	SetPropertyByTagIndirect (RT Professional).....	1260
3.2.18	SetTag (RT Professional).....	1261
3.2.19	SetTagByProperty (RT Professional).....	1262
3.2.20	SetTagByTagIndirect (RT Professional).....	1263
3.2.21	SetTagIndirect (RT Professional).....	1264
3.2.22	SetTagIndirectByProperty (RT Professional).....	1265
3.2.23	SetTagIndirectByTagIndirect (RT Professional).....	1266
3.2.24	SetTagIndirectWithOperatorInputAlarm (RT Professional).....	1267
3.2.25	SetTagWithOperatorEvent (RT Professional).....	1268
3.2.26	ShowLogonDialog (RT Professional).....	1269
3.2.27	StopRuntime (RT Professional).....	1269
3.3	VBScript für Windows (Panels, Comfort Panels, RT Advanced, RT Professional).....	1270
3.3.1	VBScript für Windows (Panels, Comfort Panels, RT Advanced, RT Professional).....	1270
3.4	VBScript für Windows CE (Panels, Comfort Panels).....	1271

3.4.1	VBScript für Windows CE (Panels, Comfort Panels).....	1271
3.4.1.1	Attr (Panels, Comfort Panels).....	1271
3.4.1.2	Close (Panels, Comfort Panels).....	1272
3.4.1.3	CreateObject (Panels, Comfort Panels).....	1272
3.4.1.4	Dir (Panels, Comfort Panels).....	1273
3.4.1.5	EOF (Panels, Comfort Panels).....	1274
3.4.1.6	FileCopy (Panels, Comfort Panels).....	1275
3.4.1.7	FileLen (Panels, Comfort Panels).....	1275
3.4.1.8	FileDateTime (Panels, Comfort Panels).....	1276
3.4.1.9	Get (Panels, Comfort Panels).....	1277
3.4.1.10	GetAttr (Panels, Comfort Panels).....	1278
3.4.1.11	Input (Panels, Comfort Panels).....	1279
3.4.1.12	InputFields (Panels, Comfort Panels).....	1280
3.4.1.13	InputB (Panels, Comfort Panels).....	1281
3.4.1.14	Kill (Panels, Comfort Panels).....	1281
3.4.1.15	LineInputString (Panels, Comfort Panels).....	1282
3.4.1.16	LinePrint (Panels, Comfort Panels).....	1283
3.4.1.17	Loc (Panels, Comfort Panels).....	1283
3.4.1.18	LOF (Panels, Comfort Panels).....	1284
3.4.1.19	MkDir (Panels, Comfort Panels).....	1284
3.4.1.20	MoveFile (Panels, Comfort Panels).....	1285
3.4.1.21	Open (Panels, Comfort Panels).....	1286
3.4.1.22	Put (Panels, Comfort Panels).....	1287
3.4.1.23	RmDir (Panels, Comfort Panels).....	1288
3.4.1.24	Seek (Panels, Comfort Panels).....	1288
3.4.1.25	SetAttr (Panels, Comfort Panels).....	1289
3.4.1.26	WriteFields (Panels, Comfort Panels).....	1290
3.5	VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional).....	1292
3.5.1	VBS Objektmodell (RT Professional).....	1292
3.5.2	Objekte (Panels, Comfort Panels, RT Advanced).....	1294
3.5.2.1	HMIRuntime (Panels, Comfort Panels, RT Advanced).....	1294
3.5.2.2	Screens (Auflistung) (Panels, Comfort Panels, RT Advanced).....	1295
3.5.2.3	Screen (Panels, Comfort Panels, RT Advanced).....	1297
3.5.2.4	ScreenItem (Panels, Comfort Panels, RT Advanced).....	1298
3.5.2.5	ScreenItems (Panels, Comfort Panels, RT Advanced).....	1300
3.5.2.6	SmartTags (Panels, Comfort Panels, RT Advanced).....	1301
3.5.2.7	SmartTag (Panels, Comfort Panels, RT Advanced).....	1302
3.5.3	Objekte (RT Professional).....	1304
3.5.3.1	Alarm (RT Professional).....	1304
3.5.3.2	Alarms (Auflistung) (RT Professional).....	1305
3.5.3.3	AlarmLogs (Auflistung) (RT Professional).....	1306
3.5.3.4	DataItem (RT Professional).....	1308
3.5.3.5	DataLogs (Auflistung) (RT Professional).....	1309
3.5.3.6	DataSet (Auflistung) (RT Professional).....	1311
3.5.3.7	HMIRuntime (RT Professional).....	1313
3.5.3.8	Item (RT Professional).....	1314
3.5.3.9	Layer (RT Professional).....	1315
3.5.3.10	Layers (Auflistung) (RT Professional).....	1317
3.5.3.11	Logging (RT Professional).....	1318
3.5.3.12	Project (RT Professional).....	1319
3.5.3.13	Screen (RT Professional).....	1320
3.5.3.14	ScreenItem (RT Professional).....	1323

3.5.3.15	ScreenItems (Auflistung) (RT Professional).....	1325
3.5.3.16	Screens (Auflistung) (RT Professional).....	1327
3.5.3.17	SmartTag (RT Professional).....	1329
3.5.3.18	SmartTags (RT Professional).....	1331
3.5.3.19	Tag (RT Professional).....	1332
3.5.3.20	Tags (Auflistung) (RT Professional).....	1335
3.5.3.21	TagSet (Auflistung) (RT Professional).....	1336
3.5.4	Objekttypen (Panels, Comfort Panels, RT Advanced, RT Professional).....	1338
3.5.4.1	Objekte A-I (Panels, Comfort Panels, RT Advanced, RT Professional).....	1338
3.5.4.2	Objekte K-Z (Panels, Comfort Panels, RT Advanced, RT Professional).....	1422
3.5.5	Eigenschaften (Panels, Comfort Panels, RT Advanced, RT Professional).....	1545
3.5.5.1	Eigenschaften A (Panels, Comfort Panels, RT Advanced, RT Professional).....	1545
3.5.5.2	Eigenschaften B (Panels, Comfort Panels, RT Advanced, RT Professional).....	1573
3.5.5.3	Eigenschaften C (Panels, Comfort Panels, RT Advanced, RT Professional).....	1608
3.5.5.4	Eigenschaften D (Panels, Comfort Panels, RT Advanced, RT Professional).....	1632
3.5.5.5	Eigenschaften E-F (Panels, Comfort Panels, RT Advanced, RT Professional).....	1642
3.5.5.6	Eigenschaften G-H (Panels, Comfort Panels, RT Advanced, RT Professional).....	1677
3.5.5.7	Eigenschaften I-J (Panels, Comfort Panels, RT Advanced, RT Professional).....	1697
3.5.5.8	Eigenschaften K-L (Panels, Comfort Panels, RT Advanced, RT Professional).....	1702
3.5.5.9	Eigenschaften M-N (Panels, Comfort Panels, RT Advanced, RT Professional).....	1728
3.5.5.10	Eigenschaften O-P (Panels, Comfort Panels, RT Advanced, RT Professional).....	1754
3.5.5.11	Eigenschaften Q-R (Panels, Comfort Panels, RT Advanced, RT Professional).....	1780
3.5.5.12	Eigenschaften S (Panels, Comfort Panels, RT Advanced, RT Professional).....	1792
3.5.5.13	Eigenschaften T (Panels, Comfort Panels, RT Advanced, RT Professional).....	1850
3.5.5.14	Eigenschaften U-W (Panels, Comfort Panels, RT Advanced, RT Professional).....	1951
3.5.5.15	Eigenschaften X-Z (Panels, Comfort Panels, RT Advanced, RT Professional).....	1989
3.5.6	Methoden (Panels, Comfort Panels, RT Advanced, RT Professional).....	2157
3.5.6.1	Methoden A-G (Panels, Comfort Panels, RT Advanced, RT Professional).....	2157
3.5.6.2	Methoden H-R (Panels, Comfort Panels, RT Advanced, RT Professional).....	2223
3.5.6.3	Methoden S-Z (Panels, Comfort Panels, RT Advanced, RT Professional).....	2251
Index		2279

C-Scripting (RT Professional)

1.1 Systemfunktionen (RT Professional)

1.1.1 ActivateNextScreen (RT Professional)

Beschreibung

WinCC speichert die Namen der Bilder, die zur Laufzeit durch den Anwender geöffnet wurden, sowie die Reihenfolge, in der diese Bilder geöffnet wurden.

Nur in C-Scripting verwendbar.

Die Größe des Bildpuffers wird im Editor "Runtime-Einstellungen > Bilder > Bildpuffer" festgelegt.

Die Systemfunktion ActivateNextScreen öffnet das Bild, das vor dem letzten Aufruf von ActivatePreviousScreen geöffnet wurde.

Syntax

```
BOOL ActivateNextScreen();
```

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ruft mit der Funktion ActivateNextScreen das nächste Bild auf und speichert den Rückgabewert in der Variablen b_error.

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

```
{
BOOL b_error;

//Open next screen
b_error = ActivateNextScreen();

if(b_error)
{
    // User defined code if
    // function succeeds without error
    ...
}
else
{
    // User defined code in case of error
    ...
}
...
}
```

1.1.2 ActivatePreviousScreen (RT Professional)

Beschreibung

WinCC speichert die Namen der Bilder, die zur Laufzeit durch den Anwender geöffnet wurden, sowie die Reihenfolge, in der diese Bilder geöffnet wurden.

Die Systemfunktion kann nur in C-Scripting verwendet werden.

Die Größe des Bildpuffers wird im Editor "Runtime-Einstellungen > Bilder > Bildpuffer" festgelegt.

Die Systemfunktion ActivatePreviousScreen öffnet das Bild, das vor dem aktuell geöffneten Bild geöffnet war.

Syntax

```
BOOL ActivatePreviousScreen();
```

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ruft mit der Funktion `ActivatePreviousScreen` das vorherige Bild auf und speichert den Rückgabewert in der Variablen `b_error`.

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

```
{
BOOL b_error;

//Open previous screen
b_error = ActivatePreviousScreen();

if(b_error)
{
    // User defined code if
    // function succeeds without error
    ...
}
else
{
    // User defined code in case of error
    ...
}
...
}
```

1.1.3 ActivateScreen (RT Professional)

Beschreibung

Führt einen Bildwechsel zu dem angegebenen Bild aus.

Um vom Grundbild in das Permanentfenster oder umgekehrt zu wechseln, verwenden Sie die Systemfunktion "AktiviereBildMitNummer".

Verwendung in der Funktionsliste

AktiviereBild (Bildname, Objektnummer)

Verwendung in benutzerdefinierten Funktionen

ActivateScreen (Screen_name, Object_number)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bildname

Name des Bildes, zu dem gewechselt wird.

Objektnummer

Das Bedienelement, das nach dem Bildwechsel im angegebenen Bild den Fokus erhält. Die Nummer des Bedienelements legen Sie während der Projektierung durch die Tab-Reihenfolge fest.

Wenn Sie "0" angeben:

- Ist der Fokus beim Aufruf der Systemfunktion im Permanentfenster, behält das Permanentfenster den Fokus.
- Ist der Fokus beim Aufruf der Systemfunktion im Grundbild, erhält das erste Bedienelement im angegebenen Bild den Fokus.

Hinweis

Wenn die Systemfunktion "AktiviereBild" an das Ereignis "Rand erreicht" projiziert wird, so ist für den Parameter "Objektnummer" nur der Wert 0 zulässig. Das aktive Objekt wird nicht anhand der Objektnummer festgelegt, sondern anhand der X-Position des aktiven Objekts vor dem Bildwechsel.

Beispiel

Der folgende Programmcode aktiviert mit der Funktion ActivateScreen beim Klicken auf eine Taste das Bild "Screen_2".

```
{  
  
// User defined code  
// i.e. when pressing a button  
ActivateScreen ("Screen_2", 0);  
...  
}
```

1.1.4 ActivateScreenInScreenWindow (RT Professional)

Beschreibung

Führt im angegebenen Bildfenster einen Bildwechsel zu dem angegebenen Bild aus.

Verwendung in der Funktionsliste

AktiviereBildInBildfenster (Bildname, Bildfenster, Neuer Bildname)

Verwendung in benutzerdefinierten Funktionen

ActivateScreenInScreenWindow (Screen_name, Screen_window, New_screen_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Bildname

Name des Bildes, das im Bildfenster angezeigt wird.

Bildfenster

Name des Bildfensters, in dem das neue Bild angezeigt wird.

Neuer Bildname

Name des neuen Bildes, das im Bildfenster angezeigt wird.

Beispiel

Der folgende Programmcode aktiviert mit der Funktion ActivateScreenInScreenWindow beim Klicken auf eine Taste das Bild "Screen_2".

```
{  
// User defined code  
// i.e. when pressing a button  
ActivateScreenInScreenWindow (GetParentScreen(screenName),  
GetParentScreenWindow(screenName), "Screen_2");  
...  
}
```

1.1.5 ActivateStartScreen (RT Professional)

Beschreibung

Öffnet das projektierte Startbild.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL ActivateStartScreen();
```

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ruft mit der Funktion `ActivateStartScreen` das projektierte Startbild auf und speichert den Rückgabewert in der Variablen `b_error`.

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

```
{  
BOOL b_error;  
  
//Open start screen  
b_error = ActivateStartScreen();  
  
if(b_error)  
{  
    // User defined code if  
    // function succeeds without error  
    ...  
}  
else  
{  
    // User defined code in case of error  
    ...  
}  
...  
}
```

1.1.6 ActivateStoredScreen (RT Professional)

Beschreibung

Öffnet das Bild, das mit der Systemfunktion `StoreScreen` gespeichert worden ist.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL ActivateStoredScreen();
```

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ruft mit der Funktion `ActivateStoredScreen` das gespeicherte Bild auf und speichert den Rückgabewert in der Variablen `b_error`.

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

```
{
BOOL b_error;

//Open stored screen
b_error = ActivateStoredScreen();

if(b_error)
{
    // User defined code if
    // function succeeds without error
    ...
}
else
{
    // User defined code in case of error
    ...
}
...
}
```

Siehe auch

[StoreScreen](#) (Seite 128)

1.1.7 DecreaseTag (RT Professional)

Beschreibung

Subtrahiert den angegebenen Wert vom Wert der Variablen.

$X = X - a$

Hinweis

Die Systemfunktion verwendet als Eingangswert und Ausgangswert dieselbe Variable. Wenn Sie diese Systemfunktion zum Umrechnen eines Wertes verwenden, müssen Sie mit einer Hilfsvariablen arbeiten. Der Hilfsvariablen können Sie mit der Systemfunktion "SetzeVariable" den Variablenwert zuweisen.

Wenn Sie die Systemfunktion an Ereignisse einer Meldung projektieren und die Variable wird im aktuellen Bild nicht verwendet, ist nicht sichergestellt, dass der tatsächliche Variablenwert in der Steuerung verwendet wird. Dies kann durch Setzen der Erfassungsart "Zyklisch fortlaufend" verbessert werden.

Verwendung in der Funktionsliste

VerringereVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

DecreaseTag (Tag, Value)

Verwendbar, sofern das projizierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variable

Die Variable, von welcher der angegebene Wert subtrahiert wird.

Wert

Der Wert, der subtrahiert wird.

Beispiel

Der folgende Programmcode erniedrigt den Wert der Variablen varX um den Wert in der Variable value. Der eingegebene Wert wird in der Variable old_value gesichert und zusammen mit dem neuen Wert varX ausgegeben.

```
{
BYTE varX;
BYTE value;

//user input
...
BYTE old_value = varX;

//Decrease tag
DecreaseTag(varX, value);

//print original value and function result
printf ("User input: %i\r\n, Result of function DecreaseTag: %i\r\n", old_value, varX);
...
}
```

1.1.8 GetLocalScreen (RT Professional)

Beschreibung

Liefert einen Zeiger auf den Namen des Bildes.

Nur in C-Scripting verwendbar.

Syntax

```
char* GetLocalScreen(char* Screen name);
```

Parameter

Screen name

Zeiger auf den Namen des Bildes.

Rückgabewert

Zeiger auf den Namen des Bildes.

Hinweis

Der übergebene Aufrufparameter "Screen name" muss den Aufbau haben, wie ihn das Grafiksystem für Bildpfade bildet:

<Bildname>.<Bildfenstername>.<Bildname>.<Bildfenstername>.<Bildname>...

Prinzip

In einem Bild "Bild_A" liegt ein Bildfenster "Bildfenster_B". Im Bildfenster wird erneut ein Bild aufgerufen und als Abbild verkleinert dargestellt usw.

Aufruf der Bilder aus den Bildfenstern

1. In "Bildfenster_B" wird "Bild_C" abgebildet.
Im Bild "Bild_C" liegt ein Bildfenster "Bildfenster_D".
2. In "Bildfenster_D" wird "Bild_E" abgebildet.

Zeiger auf den Bildnamen

liefert im folgenden Schritt die Systemfunktion `GetLocalScreen(Screen name)` mit folgenden Werten:

- "Bild_E", wenn die Systemfunktion im Bild "Bild_E" aufgerufen wird
- "Bild_C", wenn die Systemfunktion im Bild "Bild_C" aufgerufen wird
- "Bild_A", wenn die Systemfunktion im Bild "Bild_A" aufgerufen wird

Beispiel

Der folgende Programmcode speichert den Rückgabewert der Funktion GetLocalScreen in der Variable pszScrName. Ist der Rückgabewert gültig (nicht NULL), wird er mit maximal _MAX_PATH Zeichen in der Variablen szScrName gespeichert.

```
{
char* pszScrName = NULL;
char szScrName[_MAX_PATH+1];

//Get the Local Screen
pszScrName = GetLocalScreen(lpszScreenName);

//Copy the string
if (pszScrName != NULL)
{
    strncpy(szScrName,pszScrName,_MAX_PATH);
// print local screen name
    printf ("Local screen name: %s\r\n", szScrName);
}
...
}
```

1.1.9 GetLinkedTag (RT Professional)

Beschreibung

Gibt den Namen der Variablen zurück, die mit der angegebenen Objekteigenschaft verknüpft ist.

Nur in C-Scripting verwendbar.

Syntax

```
char* GetLinkedTag(char* Screen Name, char* Object, char* Name of Property);
```

Parameter

Screen Name

Zeiger auf den Namen des Bildes.

Object

Zeiger auf den Namen des Objekts.

Name of Property

Zeiger auf den Namen der Objekteigenschaft.

Rückgabewert

Zeiger auf den Namen der Variablen, die mit der angegebenen Objekteigenschaft verknüpft ist.

Beispiel

Der folgende Programmcode liest eine Variablenverbindung aus und füllt die übergebene Struktur linkinfo mit den Informationen der Variablenverbindung.

```
{  
LINKINFO linkinfo;  
  
//Get the linked Tag  
GetLinkedTag(lpszScreenName, "Bar1", "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  
...  
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.10 GetLanguageByLocaleID (RT Professional)

Beschreibung

Ermittelt die aktuelle Runtime Sprache.
Nur in C-Scripting verwendbar.

Syntax

```
DWORD GetLanguageByLocaleID ();
```

Rückgabewert

Sprachkennung.

Es gelten folgende Zuordnungen (Sprachkennung hexadezimal):

symbolische Bezeichnung	Wert(hexadezimal)	Kürzel
LANG_ARABIC	0x0401	
LANG_AFRIKAANS	0x0436	

symbolische Bezeichnung	Wert(hexadezimal)	Kürzel
LANG_ALBANIAN	0x041C	
LANG_BASQUE	0x042D	
LANG_BULGARIAN	0x0402	
LANG_BYELORUSSIAN	0x0423	
LANG_CATALAN	0x0403	
LANG_CHINESE	0x0804	
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	SVE
LANG_THAI	0x041E	
LANG_TURKISH	0x041F	TRK
LANG_UKRAINIAN	0x0422	

Beispiel

Der folgende Programmcode liest die aktuelle Runtime-Sprache aus und speichert den Rückgabewert in die Variable `rt_language`.

Der gespeicherte Rückgabewert kann im nachfolgenden Code (hier Ausgabe mit `printf`) verarbeitet werden.

```
{  
DWORD rt_language;  
  
//Get the current language  
rt_language = GetLanguageByLocaleID ();  
  
//print language code  
printf ("Language code: %d\r\n", rt_language);  
...  
}
```

1.1.11 GetParentScreen (RT Professional)

Beschreibung

Liefert einen Zeiger auf den Namen des Bildes.

Nur in C-Scripting verwendbar.

Syntax

```
char* GetParentScreen(char* Screen name);
```

Parameter

Screen name

Zeiger auf den Namen des Bildes.

Rückgabewert

Namen des aktuellen Bildes, wenn die Systemfunktion in einem Bild aufgerufen wird.

Namenspfad des übergeordneten Bildes, wenn die Systemfunktion in einem Bildfenster aufgerufen wird.

Hinweis

Der übergebene Aufrufparameter Screen name muss den Aufbau haben, wie ihn das Grafiksystem für Bildpfade bildet:

<Bildname>.<Bildfenstername>:<Bildname>.<Bildfenstername>:<Bildname>...

Die Zeichen "." und ":" verwenden Sie ausschließlich für die Syntax. Verwenden Sie in Namensbezeichnungen daher ausschließlich die Trennzeichen "-" und "_".

Beispiel

Der folgende Programmcode speichert den Rückgabewert der Funktion GetParentScreen in der Variable pszScrName . Ist der Rückgabewert gültig (nicht NULL), wird er mit maximal _MAX_PATH Zeichen in der Variablen szScrName gespeichert.

```
{
char* pszScrName = NULL;
char szScrName[_MAX_PATH+1];

//Get the Parent Screen
pszScrName = GetParentScreen(lpszScreenName);

//Copy the string
if (pszScrName != NULL)
{
    strncpy(szScrName,pszScrName,_MAX_PATH);
// print Screen name
    printf ("Screen name: %s\r\n", szScrName);
}
...
}
```

1.1.12 GetParentScreenWindow (RT Professional)

Beschreibung

Liefert einen Zeiger auf den Namen des Bildfensters.

Nur in C-Scripting verwendbar.

Syntax

```
char* GetParentScreenWindow(char* Screen name);
```

Parameter

Screen name

Zeiger auf den Namen des Bildes.

Rückgabewert

Zeiger auf den Namen des Bildfensters, wenn die Systemfunktion in einem Bild aufgerufen wird, das in einem Bildfenster eines übergeordneten Bildes angezeigt wird.

Der unveränderte Aufrufparameter "Screen name", wenn die Systemfunktion in einem Bild aufgerufen wird

Hinweis

Der übergebene Aufrufparameter "Screen name" muss den Aufbau haben, wie ihn das Grafiksystem für Bildpfade bildet:

<Bildname>.<Bildfenstername>:<Bildname>.<Bildfenstername>:<Bildname>...

Die Zeichen ":" und "." verwenden Sie ausschließlich für die Syntax. Verwenden Sie in Namensbezeichnungen daher ausschließlich die Trennzeichen "-" und "_".

Prinzip

In einem Bild "Bild_1" liegt ein Bildfenster "Bildfenster_1". Im Bildfenster wird erneut ein Bild aufgerufen und als Abbild verkleinert dargestellt usw.

Aufruf der Bilder aus den Bildfenstern

1. In "Bildfenster_1" wird "Bild_2" abgebildet.
Im Bild "Bild_2" liegt ein Bildfenster "Bildfenster_2".
2. In "Bildfenster_2" wird "Bild_3" abgebildet.

Zeiger auf den Bildfensternamen

liefert im folgenden Schritt die Systemfunktion `GetParentScreenWindow(Screen name)` mit folgenden Werten:

- "Bild_2", wenn die Systemfunktion im Bild "Bild_3" aufgerufen wird
- "Bildfenster_1", wenn die Systemfunktion im Bild "Bild_2" aufgerufen wird
- "Bild_1", wenn die Systemfunktion im Bild "Bild_1" aufgerufen wird

Beispiel

Der folgende Programmcode speichert den Rückgabewert der Funktion GetParentScreenWindow in der Variablen pszScrName. Ist der Rückgabewert gültig (nicht NULL), wird er mit maximal _MAX_PATH Zeichen in der Variablen szScrName gespeichert.

```
{
char* pszScrName = NULL;
char szScrName[_MAX_PATH+1];

//Get the Parent Screen Window
pszScrName = GetParentScreenWindow(lpszScreenName);

//Copy the string
if (pszScrName != NULL)
{
    strncpy(szScrName,pszScrName,_MAX_PATH);
// print name of the parent screen window
    printf ("Parent screen window: %s\r\n", szScrName);
}
...
}
```

1.1.13 GetProp (RT Professional)

1.1.13.1 GetPropBOOL (RT Professional)

Beschreibung

Gibt den Wert einer Eigenschaft vom Datentyp "BOOL" zurück.

Nur in C-Scripting verwendbar.

Syntax

BOOL GetPropBOOL(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property)

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest aus, ob das Objekt sichtbar ist oder nicht. Der Wert wird in der Variable `b_error` gespeichert.

```
{
BOOL b_error;

//Get the property Visible
b_error = GetPropBOOL("gs_graph_iofield","IOField1","Visible");

if(b_error)
{
    // User defined code if the
    // object is visible
    ...
}
else
{
    // User defined code if the
    // object is not visible
    ...
}
}
```

Abhängig vom Rückgabewert wird ein spezifischer Code verarbeitet.

1.1.13.2 GetPropChar (RT Professional)

Beschreibung

Gibt den Wert einer Eigenschaft vom Datentyp "Char" zurück.

Nur in C-Scripting verwendbar.

Syntax

Char* GetPropChar(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property)

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Rückgabewert

Wert der Eigenschaft im Datentyp "Char".

Beispiel

Der folgende Programmcode liest den Toolliptext des Objekts mit der Funktion GetPropChar aus und verarbeitet ihn wie folgt:

1. Speichern des Rückgabewert in der Variable pszProp
2. Prüfen auf Gültigkeit des Rückgabewertes: Falls der Wert gültig (nicht NULL) ist, folgt Schritt 3.
3. Die ersten 13 Zeichen der Zeichenfolge werden in der Variable szProp gespeichert.

```
{
char* pszProp = NULL;
char szProp[14];

//Get the property Toolliptext
pszProp = GetPropChar("lpszScreenName", "IOField1", "Toolliptext");

if(pszProp != NULL)
{
//Copy the string and trim
strncpy(szProp,pszProp,13);
// print trimmed string
printf ("Short description of tooltip: %s\r\n", szProp);
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.13.3 GetPropDouble (RT Professional)

Beschreibung

Gibt den Wert einer Eigenschaft vom Datentyp "Double" zurück.

Nur in C-Scripting verwendbar.

Syntax

```
double GetPropDouble(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property)
```

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Rückgabewert

Wert der Eigenschaft im Datentyp "Double".

Beispiel

Der folgende Programmcode liest die Eigenschaft "BackColor" (Hintergrundfarbe des Objekts "Button_1") mit der Funktion GetPropDouble aus und verarbeitet den Rückgabewert wie folgt:

1. Speichern des Rückgabewert in der Variable szprop
2. Prüfen auf Gültigkeit des Rückgabewertes: Falls der Wert gültig (nicht NULL) ist, folgt Schritt 3.
3. Ausgabe der Hintergrundfarbe

```
{
double szProp = NULL;

//Get the property of the IO Field 1
szprop = GetPropDouble("Screen_1","Button_1","BackColor");

if(szProp != NULL)
{
// print output value
printf ("Background color: %s\r\n", szProp);}
}
...
}
```

1.1.13.4 GetPropLong (RT Professional)

Beschreibung

Gibt den Wert einer Eigenschaft vom Datentyp "long" zurück.

Nur in C-Scripting verwendbar.

Syntax

long GetPropLong(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property)

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Rückgabewert

Wert der Eigenschaft im Datentyp "long".

Beispiel

Der folgende Programmcode liest die Eigenschaft "CaptionBackColor" (Hintergrundfarbe der Überschrift) des Objekts "TemperatureField" mit der Funktion GetPropLong aus und verarbeitet den Wert wie folgt:

1. Speichern des Rückgabewert in der Variable szProp
2. Prüfen auf Gültigkeit des Rückgabewertes: Falls der Wert gültig (nicht NULL) ist, folgt Schritt 3.
3. Ausgabe

```
{
long szProp = NULL;

//Get the property Tooltiptext
szProp = GetPropLong("Screen_1", "TemperatureField", "CaptionBackColor");

if(szProp != NULL)
{
// print caption
printf ("Caption of window: %d\r\n", szProp);
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.14 GetServerTagPrefix (RT Professional)

Beschreibung

Um in einem verteilten System von einem WinCC-Client auf Variablen des zugehörigen Servers zugreifen zu können, sind die Variablennamen um das Serverpräfix zu ergänzen.

Hinweis

Diese Systemfunktion wird derzeit nicht unterstützt.

Es wird jeweils ein Zeiger vom Typ "char" auf Server-Präfix, Tag-Präfix und Window-Präfix zurückgeliefert.

Durch den Anwender darf der Speicher weder verändert (auch kein strcat) noch freigegeben werden.

Nur in C-Scripting verwendbar.

Syntax

```
void GetServerTagPrefix (char** ppszServerPrefix, char** ppszTagPrefix, char**  
ppszWindowPrefix);
```

Rückgabewert

ppszServerPrefix

Zeiger auf einen Zeiger, der auf das Server-Präfix verweist

ppszTagPrefix

Zeiger auf einen Zeiger, der auf das Tag-Präfix verweist

ppszWindowPrefix

Zeiger auf einen Zeiger, der auf das Window-Präfix verweist

Beispiel

Der folgende Programmcode liest **Server-Präfix**, **Tag-Präfix** und **Window-Präfix** aus und überprüft ihre Gültigkeit. Im Fehlerfall wird ein Text ausgegeben und die Funktion verlassen. Bei Erfolg wird ein Variablenname erstellt und zurückgegeben. Die Verarbeitung ist wie folgt:

1. Deklarieren der Zeiger `pszServerPrefix`, `pszTagPrefix` und `pszWindowPrefix` für die drei Präfixe
2. Initialisieren der Variablen `nServerPrefixLen`, `nTagPrefixLen` und `nTagLen`
Sie dienen als Zwischenspeicher für die Stringlänge der auszulesenden Präfixe.
3. Initialisieren der Variable `myTagName`
4. Auslesen von Serverpräfix, Variablenpräfix und Windowpräfix
5. **Fallunterscheidung**: Serverpräfix
 - **Kein** Serverpräfix zurückgegeben: es wird ein Text ausgegeben und die Funktion verlassen.
 - Serverpräfix zurückgegeben: es wird dessen Länge ermittelt und in der Variable `nServerPrefixLen` gespeichert.
6. Falls ein Variablenpräfix zurückgegeben wurde, wird dessen Länge ermittelt und in der Variable `nTagPrefixLen` gespeichert.
7. Ermitteln der Länge des Variablennamens und speichern in der Variable `nTagLen`.
8. **Fallunterscheidung**: Zulässige Länge für Variablenname
 - Zulässige Länge überschritten: es wird ein Text ausgegeben und die Funktion verlassen.
 - Zulässige Länge **nicht** überschritten: es wird der Variablenname, der für eine Client-Umgebung benötigt wird, zusammengesetzt.

1.1 Systemfunktionen (RT Professional)

```
{
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;
}
}
```

1.1.15 GetTag (RT Professional)

1.1.15.1 GetTag-Funktionen (RT Professional)

Beschreibung

Die GetTagXXX-Funktion ermittelt den Wert einer Variablen vom angegebenen Datentyp

Nur in C-Scripting verwendbar:

Die folgende Tabelle zeigt die verschiedenen GetTag-Funktionen zum Lesen des Variablenwerts:

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
BOOL	GetTagBit	Tag Tag_Name	Binäre Variable	Bool
BYTE	GetTagByte	Tag Tag_Name	Vorzeichenlos 8-Bit	USInt
char*	GetTagChar	Tag Tag_Name	Textvariable 8-Bit oder Textvariable 16-Bit	String oder WString
SYSTEM-TIME	GetTagDateTime	Tag Tag_Name	DTL	DateTime
double	GetTagDouble	Tag Tag_Name	Gleitkommazahl 64-Bit	LReal
DWORD	GetTagDWord	Tag Tag_Name	Vorzeichenlos 32-Bit	UDInt
float	GetTagFloat	Tag Tag_Name	Gleitkommazahl 32-Bit	Real
BOOL	GetTagRaw	Tag Tag_Name, BYTE* pValue, DWORD size	Rohdatentyp	Raw
signed char	GetTagSByte	Tag Tag_Name	Vorzeichenbehaftet 8-Bit	SInt
long int	GetTagSDWord	Tag Tag_Name	Vorzeichenbehaftet 32-Bit	DInt
short int	GetTagSWord	Tag Tag_Name	Vorzeichenbehaftet 16-Bit	Int
WORD	GetTagWord	Tag Tag_Name	Vorzeichenlos 16-Bit	UInt

Syntax

```
<Typ><Funktionsname><(Parameter)>;
```

Beispiel: BYTE GetTagByte (Tag Tag_Name);

Parameter

Tag_Name

Name der Variablen

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

Wert der Variablen im angegebenen Typ.

Die Systemfunktion "GetTagChar" gibt einen Zeiger auf die Zeichenkette zurück, die den Wert der Variablen enthält.

Die Systemfunktion "GetTagRaw" gibt TRUE oder FALSE zurück:

TRUE: Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE: Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagByte den Wert der Variablen gs_tag_byte aus und speichert ihn in der Variable bvalue.

```
{
BYTE bvalue;

//Get the current value of the tag
bvalue = GetTagByte("gs_tag_byte");

// print value
printf ("Value of gs_tag_byte: %d\r\n", bvalue);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.15.2 GetTagDateTime-Funktion (RT Professional)

Funktion

Ermittelt den Wert einer Variablen vom Datentyp "Datum/Uhrzeit".

Syntax

```
SYSTEMTIME GetTagDateTime(Tag Tag_Name);
```

Parameter

Tag_Name

Name der Variablen

Rückgabewert

Wert der Variablen im Datentyp "Datum/Uhrzeit".

1.1.15.3 GetTagMultiStateQCWait-Funktionen (RT Professional)**Beschreibung**

Die Werte, Status und Quality Codes mehrerer Variablen werden ermittelt und an den entsprechenden Adressen im angegebenen Format abgelegt. Die Werte werden explizit aus dem AS gelesen.

Zwei DWORD Arrays werden übergeben, in dessen Member sich nach dem Aufruf der Systemfunktion die Status und Quality Codes der einzelnen Variablen befinden. Die Arrays müssen so groß gewählt werden, dass für diese Status und Quality Codes genügend Speicherplatz zur Verfügung steht.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL GetTagMultiStateQCWait (DWORD* pdwState, DWORD* pdwQualityCode, const char* pFormat, ...);
```

Parameter**pdwState**

Feld, in dem nach dem Durchlauf der Systemfunktion der Status der einzelnen Variablen abgelegt wird.

pdwQualityCode

Feld, in dem nach dem Durchlauf der Systemfunktion der Quality Code der einzelnen Variablen abgelegt wird.

pFormat

Formatbeschreibung (Typ) für alle angeforderten Variablen, danach für jede Variable Name und Adresse des Wertes.

Rückgabewert**TRUE**

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagMultiStateQCWait den Wert der Variablen "gs_tag_XXX" aus und speichert diese Werte in den Variablen lvalue1, lvalue2 ... usw.

1. Präprozessor-Definition für "DATA_SIZE" (hier für 5 Variablen)
2. Anlegen der DWord-Felder
 - dwState Feld für Variablenstatus
 - dwQc Feld für Quality Codes
3. Definition der Variablen zum Zwischenspeichern
4. Ausführen der Funktion GetTagMultiStateQCWait
In die Adressen der Variablen wird der ausgelesene Wert geschrieben.

```
{
#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
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.15.4 GetTagMultiStateWait-Funktionen (RT Professional)

Beschreibung

Die Werte und Status mehrerer Variablen werden ermittelt und an den entsprechenden Adressen im angegebenen Format abgelegt. Die Werte werden explizit aus dem AS gelesen.

Der Systemfunktion muss ein DWORD Array übergeben werden, in dessen Member sich nach dem Aufruf der Systemfunktion die Status der einzelnen Variablen befinden. Das Array muss so groß gewählt werden, dass für die Status genügend Speicherplatz zur Verfügung steht.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL GetTagMultiStateWait(DWORD* pdwState, const char* pFormat);
```

Parameter

pdwState

Feld, in dem die Variablenstatus gespeichert werden

pFormat

Formatbeschreibung (Typ) für alle angeforderten Variablen und für jede Variable Name und Adresse des Wertes.

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagMultiStateWait den Wert der Variablen "gs_tag_XXX" aus und speichert diese Werte in den Variablen lvalue1, lvalue2 ... usw.

1. Präprozessor-Definition für "DATA_SIZE" (hier für 5 Variablen)
2. Anlegen des DWord-Feldes dwState für den Variablenstatus
3. Definition der Variablen zum Zwischenspeichern
4. Ausführen der Funktion GetTagMultiStateWait
In die Adressen der Variablen wird der ausgelesene Wert geschrieben.

1.1 Systemfunktionen (RT Professional)

```
{
#define DATA_SIZE 5
DWORD dwState[DATA_SIZE];

//define all Datas
BOOL lValue1;
long lValue2 ;
char* szValue3;
double dblValue4 ;
WORD lValue5 ;

//Set the tags
GetTagMultiStateWait(dwState,"%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
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.15.5 GetTagMultiWait-Funktionen (RT Professional)

Beschreibung

Die Werte mehrerer Variablen werden ermittelt und bei den entsprechenden Adressen im angegebenen Format abgelegt. Der Wert wird explizit aus dem AS gelesen. Der Speicher für den Variablenwert wird durch die Systemfunktion mit SysMalloc angelegt.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL GetTagMultiWait(const char* pFormat,...);
```

Parameter

pFormat

Formatbeschreibung für alle angeforderten Variablen und für jede Variable Name und Adresse des Werts.

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagMultiWait mehrere Variablen verschiedenen Typs.

1. Deklarieren von drei Variablen als Speicher für drei verschiedene Variablentypen
2. Deklarieren der boolschen Variablen ok zum Zwischenspeichern des Rückgabewertes (TRUE/FALSE)
3. Auslesen der drei Variablen und speichern der Werte in den zugehörigen Adressen. Der Rückgabewert der Funktion wird in der Variable ok gespeichert.
4. Ausgabe der drei Variablen mit vorangestelltem Variablentyp

```
DWORD dwVar1Value;  
char* szVar2Value;  
//memory for values allocated via SysMalloc  
  
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);
```

1.1.15.6 GetTagState-Funktionen (RT Professional)

Beschreibung

Die GetTagStateXXX-Funktion ermittelt den Wert einer Variablen vom angegebenen Datentyp. Zusätzlich wird der Status der Variablen zurückgegeben.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen GetTagStateXXX-Funktionen zum Lesen des Variablenwerts:

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
BOOL	GetTagBitState	Tag Tag_Name, PDWORD lp_dwstate	Binäre Variable	Bool
BYTE	GetTagByteState	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenlos 8-Bit	Byte
char*	GetTagCharState	Tag Tag_Name, PDWORD lp_dwstate	Textvariable 8-Bit oder Textvariable 16-Bit	String
double	GetTagDoubleState	Tag Tag_Name, PDWORD lp_dwstate	Gleitkommazahl 64-Bit	Double
DWORD	GetTagDWordState	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenlos 32-Bit	UInteger
float	GetTagFloatState	Tag Tag_Name, PDWORD lp_dwstate	Gleitkommazahl 32-Bit	Float
BOOL	GetTagRawState	Tag Tag_Name, BYTE* pValue, DWORD size, PDWORD lp_dwstate	Rohdatentyp	Raw
signed char	GetTagSByteState	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenbehaftet 8-Bit	Byte
long int	GetTagSDWordState	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenbehaftet 32-Bit	Integer
short int	GetTagSWordState	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenbehaftet 16-Bit	Short
WORD	GetTagWordState	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenlos 16-Bit	UShort

Syntax

<Typ><Funktionsname><(Parameter)>;

Beispiel: BOOL GetTagBitState (Tag Tag_Name, PDWORD lp_dwstate);

Parameter

Tag_Name

Name der Variablen

lp_dwstate

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Status der Variablen abgelegt wird.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

Wert der Variablen im angegebenen Typ.

Die Systemfunktion "GetTagCharState" gibt einen Zeiger auf den Wert der Variablen im Datentyp "char" zurück.

Die Systemfunktion "GetTagRawState" gibt TRUE oder FALSE zurück:

TRUE: Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE: Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagBitState den Wert der Variablen gs_tag_bit aus und speichert ihn in der Variable bValue.

Der Status wird in der Adresse der Variable dwState gespeichert.

Je nach Rückgabewert in bValue (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```
{
DWORD dwState;
BOOL bValue;

dwState = 0xFFFFFFFF;

//Get the tag value
//dwstate is the tag state
bValue = GetTagBitState("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
    ...
}
}
```

Siehe auch

Konstanten (Seite 202)

1.1.15.7 GetTagStateQC-Funktionen (RT Professional)

Beschreibung

Die GetTagStateQC-Funktion ermittelt den Wert einer Variablen vom angegebenen Datentyp. Zusätzlich werden der Status und der Quality Code der Variablen zurückgegeben.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen GetTagStateQC-Funktionen zum Lesen des Variablenwerts:

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
BOOL	GetTagBitStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Binäre Variable	Bool
BYTE	GetTagByteStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Vorzeichenlos 8-bit	UByte
char*	GetTagCharStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Textvariable 8-bit oder Textvariable 16-bit	String
double	GetTagDoubleStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Gleitkommazahl 64-bit	Double
DWORD	GetTagDWordStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Vorzeichenlos 32-bit	UInteger
float	GetTagFloatStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Gleitkommazahl 32-bit	Float
BOOL	GetTagRawStateQC	Tag Tag_Name, BYTE pValue[], DWORD size, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Rohdatentyp	Raw
signed char	GetTagSByteStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Vorzeichenbehaftet 8-bit	Byte
long int	GetTagSDWordStateQC	Tag Tag_Name, PDWORD Ip_dwstate, PDWORD pdwQualityCode	Vorzeichenbehaftet 32-bit	Integer

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
short int	GetTagSWordStateQC	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode	Vorzeichenbehaftet 16-bit	Short
WORD	GetTagWordStateQC	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode	Vorzeichenlos 16-bit	UShort

Syntax

<Typ><Funktionsname><(Parameter)>;

Beispiel: BOOL GetTagBitStateQC (Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Parameter

Tag_Name

Name der Variablen

lp_dwstate

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Status der Variablen abgelegt wird.

pdwQualityCode

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Quality Code der Variablen abgelegt wird.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

Wert der Variablen im angegebenen Typ.

Die Systemfunktion "GetTagCharStateQC" gibt einen Zeiger auf den Wert der Variablen im Datentyp "char" zurück.

Die Systemfunktion "GetTagRawStateQC" gibt TRUE oder FALSE zurück:

TRUE: Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE: Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagBitStateQC den Wert der Variablen gs_tag_bit aus und speichert ihn in der Variable ok.

Der Status und der Quality Code wird in den Adressen dwState und dwQC der Variablen abgelegt.

Je nach Rückgabewert in der Variable ok. (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```
{
DWORD dwState;
DWORD dwQC;
BOOL ok;

dwState = 0xFFFFFFFF;

//Get the tag value
//dwstate is the tag state
ok = GetTagBitStateQC("gs_tag_bit",&dwState,&dwQC);

//Create a string which includes the tag value
if (ok)
{
    // succeeded, print tag value
    printf ("Value at &dwState,&dwQC: %d\r\n", &dwState);
    printf ("Value at &dwQC: %d\r\n", &dwQC);
    ...
}
else
{
    // failed
    printf ( "Error - function failed." );
}
}
```

1.1.15.8 GetTagStateQCWait-Funktionen (RT Professional)

Beschreibung

Die GetTagStateQCWait-Funktion ermittelt den Wert einer Variablen vom angegebenen Datentyp. Der Wert wird explizit aus dem AS gelesen. Zusätzlich werden der Status und der Quality Code der Variablen zurückgegeben.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen GetTagStateQCWait-Funktionen zum Lesen des Variablenwerts:

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Da- tentyp
BOOL	GetTagBitStateQCWait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Binäre Variable	Bool
BYTE	GetTagByteStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Vorzeichenlos 8-bit	UByte
char*	GetTagCharStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Textvariable 8-bit oder Textvariable 16-bit	String
double	GetTagDoubleStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Gleitkommazahl 64-bit	Double
DWORD	GetTagDWordStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Vorzeichenlos 32-bit	UInteger
float	GetTagFloatStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Gleitkommazahl 32-bit	Float
BOOL	GetTagRawStateQC- Wait	Tag Tag_Name, BYTE pValue[], DWORD size, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Rohdatentyp	Raw
signed char	GetTagSByteStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Vorzeichenbehaftet 8-bit	Byte
long int	GetTagSDWordSta- teQCWait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Vorzeichenbehaftet 32-bit	Integer
short int	GetTagSWordStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Vorzeichenbehaftet 16-bit	Short
WORD	GetTagWordStateQC- Wait	Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityC- ode	Vorzeichenlos 16-bit	UShort

Syntax

<Typ><Funktionsname><(Parameter)>;

Beispiel: BOOL GetTagBitStateQC (Tag Tag_Name, PDWORD lp_dwstate, PDWORD pdwQualityCode);

Parameter

Tag_Name

Name der Variablen

lp_dwstate

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Status der Variablen abgelegt wird.

pdwQualityCode

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Quality Code der Variablen abgelegt wird.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

Wert der Variablen im angegebenen Typ.

Die Systemfunktion "GetTagCharStateQCWait" gibt einen Zeiger auf den Wert der Variablen im Datentyp "char" zurück.

Die Systemfunktion "GetTagRawStateQCWait" gibt TRUE oder FALSE zurück:

TRUE: Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE: Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagBitStateQCWait den Wert der Variablen gs_tag_bit aus und speichert ihn in der Variable bValue.

Der Status und der Quality Code wird in den Adressen von dwState und dwQC abgelegt.

Je nach Rückgabewert in bValue (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```

{
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
...
}
}

```

1.1.15.9 GetTagStateWait-Funktionen (RT Professional)

Beschreibung

Die GetTagStateWait-Funktion ermittelt den Wert einer Variablen vom angegebenen Datentyp. Der Wert wird explizit aus dem AS gelesen. Zusätzlich wird der Status der Variablen zurückgegeben.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen GetTagStateWait-Funktionen zum Lesen des Variablenwerts:

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
BOOL	GetTagBitStateWait	Tag Tag_Name, PDWORD lp_dwstate	Binäre Variable	Bool
BYTE	GetTagByteState- Wait	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenlos 8- Bit	UByte
char*	GetTagCharState- Wait	Tag Tag_Name, PDWORD lp_dwstate	Textvariable 8-Bit oder Textvariable 16-Bit	String
double	GetTagDoubleState- Wait	Tag Tag_Name, PDWORD lp_dwstate	Gleitkommazahl 64-Bit	Double

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
DWORD	GetTagDWordStateWait	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenlos 32-Bit	UInteger
float	GetTagFloatStateWait	Tag Tag_Name, PDWORD lp_dwstate	Gleitkommazahl 32-Bit	Float
BOOL	GetTagRawStateWait	Tag Tag_Name, BYTE* pValue, DWORD size, PDWORD lp_dwstate	Rohdatentyp	Raw
signed char	GetTagSByteStateWait	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenbehaftet 8-Bit	Byte
long int	GetTagSDWordStateWait	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenbehaftet 32-Bit	Integer
short int	GetTagSWordStateWait	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenbehaftet 16-Bit	Short
WORD	GetTagWordStateWait	Tag Tag_Name, PDWORD lp_dwstate	Vorzeichenlos 16-Bit	UShort

Syntax

<Typ><Funktionsname><(Parameter)>

Beispiel: BOOL GetTagBitStateWait (Tag Tag_Name, PDWORD lp_dwstate)

Parameter

Tag_Name

Name der Variablen

lp_dwstate

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Status der Variablen abgelegt wird.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

Wert der Variablen im angegebenen Typ.

Die Systemfunktion "GetTagCharStateWait" gibt Zeiger auf den Wert der Variablen im Datentyp "char" zurück.

Die Systemfunktion "GetTagRawState" gibt TRUE oder FALSE zurück.

TRUE: Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE: Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagBitStateWait den Wert der Variablen gs_tag_bit aus und speichert ihn in der Variable bValue.

Der Status wird in der Adresse der Variable dwState gespeichert.

Je nach Rückgabewert in bValue (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```
{
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
    ...
}
}
```

1.1.15.10 GetTagValue-Funktionen (RT Professional)

Beschreibung

Ermöglicht die Übergabe eines Wertes in Form eines Variants. Ermittelt den Zeiger auf die Ergebnisstruktur, die den Wert enthält.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL GetTagValue(LPDM_VARKEY IpdmVarKey, LPDM_VAR_UPDATE_STRUCT
Ipdmresult, LPCMN_ERROR IpdmError);
```

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Datentyp "DM_VARKEY"

lpdmresult

Zeiger auf den Wert vom Datentyp "DM_VAR_UPDATE_STRUCT"

lpdmError

Zeiger auf die Struktur, welche die Fehlerbeschreibung enthält

Rückgabewert

TRUE

Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ermittelt mit der Funktion GetTagValue den Wert in varKey. Je nach Rückgabewert in keyFound (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```
{
DM_VARKEY varKey;
DM_VAR_UPDATE_STRUCT result;
CMN_ERROR error;

BOOL keyFound;

//Get the tag value
keyFound = GetTagValue(&varKey, &result, &error);

if (keyFound)
{
    // print tag value
    printf ("Value of varKey: %d\r\n", &varKey);
    ...
}
else
{
    // failed
    printf ( "Error - function failed." );
    ...
}
}
```

1.1.15.11 GetTagValueStateQC-Funktionen (RT Professional)

Beschreibung

Ermöglicht die Übergabe eines Wertes in Form eines Variants. Ermittelt den Zeiger auf die Ergebnisstruktur, die den Wert enthält. Zusätzlich werden der Status und der Quality Code der Variablen zurückgegeben.

Die Systemfunktion kann nur in C-Scripting verwendet werden.

Syntax

```
BOOL GetTagValueStateQC (LPDM_VARKEY lpdmVarKey,  
LPDM_VAR_UPDATE_STRUCTEX lpdmresult, LPCMN_ERROR lpdmError);
```

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Datentyp "DM_VARKEY"

lpdmresult

Zeiger auf den Wert vom Datentyp "DM_VAR_UPDATE_STRUCTEX"

lpdmError

Zeiger auf die Struktur, welche die Fehlerbeschreibung enthält

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ermittelt mit der Funktion GetTagValueStateQC den Wert in varKey.

Der Status und der Quality Code wird in den Adressen von dwState und dwQC abgelegt.

Je nach Rückgabewert in keyFound (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```
{
DM_VARKEY varKey;
DM_VAR_UPDATE_STRUCTEX result;
CMN_ERROR error;
// zu klären (DM_VAR_UPDATE_STRUCTEX enthält bereits QualityCode)
DWORD dwState;
DWORD dwQC;

BOOL keyFound;

//Get the tag value
keyFound = GetTagValueStateQC(&varKey, &result, &error,&dwState,&dwQC);

if (keyFound)
{
// User defined code if the
// value of the tag is true
...
}
else
{
// User defined code if the
// value of the tag is false
...
}
}
```

1.1.15.12 GetTagValueStateQCWait-Funktionen (RT Professional)

Beschreibung

Ermöglicht die Übergabe eines Wertes in Form eines Variants. Ermittelt den Zeiger auf die Ergebnisstruktur, die den Wert enthält. Der Wert wird explizit aus dem AS gelesen. Zusätzlich werden der Status und der Quality Code der Variablen zurückgegeben.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL GetTagValueStateQCWait (LPDM_VARKEY lpdmVarKey,
LPDM_VAR_UPDATE_STRUCTEX lpdmresult, LPCMN_ERROR lpdmError);
```

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Datentyp "DM_VARKEY"

lpdmresult

Zeiger auf den Wert vom Datentyp "DM_VAR_UPDATE_STRUCTEX"

lpdmError

Zeiger auf die Struktur, welche die Fehlerbeschreibung enthält.

Rückgabewert**TRUE**

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagValueStateQCWait

Der Status und der Quality Code wird in den Adressen von dwState und dwQC abgelegt.

Je nach Rückgabewert in keyFound (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```
{
DM_VARKEY varKey;
DM_VAR_UPDATE_STRUCTEX result;
CMN_ERROR error;
// zu klären (DM_VAR_UPDATE_STRUCTEX enthält bereits QualityCode)
DWORD dwState;
DWORD dwQC;

BOOL keyFound;

//Get the tag value
keyFound = GetTagValueStateQC(&varKey, &result, &error,&dwState,&dwQC);

if (keyFound)
{
// User defined code if the
// value of the tag is true
...
}
else
{
// User defined code if the
// value of the tag is false
...
}
}
```

1.1.15.13 GetTagValueWait-Funktionen (RT Professional)

Beschreibung

Ermöglicht die Übergabe eines Wertes in Form eines Variants. Ermittelt den Zeiger auf die Ergebnisstruktur, die den Wert enthält. Der Wert wird direkt aus der AS gelesen.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL GetTagValueWait(LPDM_VARKEY lpdmVarKey, LPDM_VAR_UPDATE_STRUCT  
lpdmresult, LPCMN_ERROR lpdmError);
```

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Datentyp "DM_VARKEY"

lpdmresult

Zeiger auf den Wert vom Datentyp "DM_VAR_UPDATE_STRUCT"

lpdmError

Zeiger auf die Struktur, die die Fehlerbeschreibung enthält

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ermittelt mit der Funktion GetTagValueWait den Wert in varKey.

Je nach Rückgabewert in keyFound (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```

{
DM_VARKEY varKey;
DM_VAR_UPDATE_STRUCT result;
CMN_ERROR error;

BOOL keyFound;

//Get the tag value
keyFound = GetTagValueWait(&varKey, &result, &error);

if (keyFound)
{
// succeeded, key found
// print tag value
printf ("Value of varKey: %d\r\n", &varKey);
...
}
else
{
// failed
printf ( "Error - function failed." );
}
...
}

```

1.1.15.14 GetTagWait-Funktionen (RT Professional)

Beschreibung

Die GetTagWaitXXX-Funktion ermittelt den Wert einer Variablen vom angegebenen Datentyp. Der Wert wird explizit aus dem AS gelesen.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen GetTagWait-Funktionen zum Lesen des Variablenwerts:

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
BOOL	GetTagBitWait	Tag Tag_Name	Binäre Variable	Bool
BYTE	GetTagByteWait	Tag Tag_Name	Vorzeichenlos 8-Bit	UByte
char*	GetTagCharWait	Tag Tag_Name	Textvariable 8-Bit oder Textvariable 16-Bit	String
double	GetTagDoubleWait	Tag Tag_Name	Gleitkommazahl 64-Bit	Double
DWORD	GetTagDWordWait	Tag Tag_Name	Vorzeichenlos 32-Bit	UInteger
float	GetTagFloatWait	Tag Tag_Name	Gleitkommazahl 32-Bit	Float
BOOL	GetTagRawWait	Tag Tag_Name, BYTE* pValue, DWORD size	Rohdatentyp	Raw
char	GetTagSByteWait	Tag Tag_Name	Vorzeichenbehaftet 8-Bit	Byte

Typ	Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
long int	GetTagSDWordWait	Tag Tag_Name	Vorzeichenbehaftet 32-Bit	Integer
short int	GetTagSWordWait	Tag Tag_Name	Vorzeichenbehaftet 16-Bit	Short
WORD	GetTagWordWait	Tag Tag_Name	Vorzeichenlos 16-Bit	UShort

Syntax

<Typ><Funktionsname><(Parameter)>;

Beispiel: BYTE GetTagByteWait (Tag Tag_Name);

Parameter

Tag_Name

Name der Variablen

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

Rückgabewert

Wert der Variablen im angegebenen Typ.

Die Systemfunktion "GetTagCharWait" gibt einen Zeiger auf die Zeichenkette zurück, die den Wert der Variablen enthält.

Die Systemfunktion "GetTagRawWait" gibt TRUE oder FALSE zurück:

TRUE: Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE: Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagByteWait den Wert der Variablen gs_tag_byte aus und speichert ihn in der Variable bvalue.

```
{
BYTE bvalue;

//Get the current value of the tag
bvalue = GetTagByteWait("gs_tag_byte");

// print value
printf ("Value of gs_tag_byte: %d\r\n", bvalue);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.16 IncreaseTag (RT Professional)

Beschreibung

Addiert den angegebenen Wert zum Wert der Variablen.

$$X = X + a$$

Hinweis

Die Systemfunktion verwendet als Eingangswert und Ausgangswert dieselbe Variable. Wenn Sie diese Systemfunktion zum Umrechnen eines Wertes verwenden, müssen Sie mit einer Hilfsvariablen arbeiten. Der Hilfsvariablen können Sie mit der Systemfunktion "SetzeVariable" den Variablenwert zuweisen.

Wenn Sie die Systemfunktion an Ereignisse einer Meldung projektieren und die Variable wird im aktuellen Bild nicht verwendet, ist nicht sichergestellt, dass der tatsächliche Variablenwert in der Steuerung verwendet wird. Dies kann durch Setzen der Erfassungsart "Zyklisch fortlaufend" verbessert werden.

Verwendung in der Funktionsliste

ErhöheVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

IncreaseTag (Tag, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variable

Die Variable, zu welcher der angegebene Wert addiert wird.

Wert

Der Wert, der addiert wird.

Beispiel

Der folgende Programmcode erhöht den Wert der Variablen varX um den Wert in der Variable value. Der eingegebene Wert wird in der Variable old_value gesichert und zusammen mit dem neuen Wert varX ausgegeben.

```
{
BYTE varX;
BYTE value;

//user input
...
BYTE old_value = varX;

//Increase tag
IncreaseTag(varX, value);

//print original value and function result
printf ("User input: %i\r\n, Result of function IncreaseTag: %i\r\n", old_value, varX);
...
}
```

1.1.17 InquireLanguage (RT Professional)

Beschreibung

Ermittelt alle Sprachen, die in der Textbibliothek für die Laufzeit projiziert sind.

Mit Pointer to a counter geben Sie an, wo die Anzahl der ermittelten Sprachkennungen abgelegt werden soll.

Die Systemfunktion kann nur in C-Scripting verwendet werden.

Syntax

DWORD* InquireLanguage (DWORD* Pointer to a counter);

Parameter

Pointer to a counter

Zeiger auf die Anzahl der ermittelten Sprachkennungen.

Rückgabewert

Zeiger auf ein Feld, das die ermittelten Sprachkennungen enthält.

Es gelten folgende Zuordnungen (Sprachkennung hexadezimal):

symbolische Bezeichnung	Wert(hexadezimal)	Kürzel
LANG_ARABIC	0x0401	
LANG_AFRIKAANS	0x0436	
LANG_ALBANIAN	0x041C	
LANG_BASQUE	0x042D	
LANG_BULGARIAN	0x0402	
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	SVE
LANG_THAI	0x041E	

symbolische Bezeichnung	Wert(hexadezimal)	Kürzel
LANG_TURKISH	0x041F	TRK
LANG_UKRAINIAN	0x0422	

Beispiel

Der folgende Programmcode ermittelt mit der Funktion `InquireLanguage` die zur Laufzeit projizierten Sprachen und verarbeitet sie wie folgt:

1. Speichern der ermittelten Sprachkennung in der Variable `language`
2. Speichern der Anzahl der Sprachen in der Variable `count`
3. Formatierte Ausgabe der Sprachanzahl und Sprachkennungen

```
{
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++);
}
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.18 InverseLinearScaling (RT Professional)

Beschreibung

Weist der Variablen `X` einen Wert zu, der aus dem Wert der angegebenen Variablen `Y` über die lineare Funktion $X = (Y - b) / a$ berechnet wird.

Die Variablen X und Y dürfen nicht identisch sein. Diese Systemfunktion ist die Umkehrfunktion zur Systemfunktion "LineareSkalierung".

Hinweis

Die Variablen X und Y dürfen nicht identisch sein. Wenn Sie eine Variable auf sich selbst umrechnen wollen, müssen Sie eine Hilfsvariable verwenden.

Mit der Systemfunktion "SetzeVariable" können Sie den Wert der umzurechnenden Variablen der Hilfsvariablen zuweisen.

Verwendung in der Funktionsliste

InvertiereLineareSkalierung (X, Y, b, a)

Verwendung in benutzerdefinierten Funktionen

InvertLinearScaling (X, Y, b, a)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter**X**

Die Variable, welcher der berechnete Wert aus der linearen Gleichung zugewiesen wird.

Y

Die Variable, deren Wert zur Berechnung verwendet wird.

b

Der Wert, der subtrahiert wird.

a

Der Wert, durch den dividiert wird.

Beispiel

Der folgende Programmcode weist der Variable varX einen Wert mit der Funktion InverseLinearScaling zu.

```
{  
BYTE varX;  
BYTE Yvalue = 10;  
BYTE bvalue = 3;  
BYTE avalue = 4;  
  
//Inverse linear scaling  
InverseLinearScaling (varX, Yvalue, bvalue, avalue);  
  
printf ("varX = %d\r\n, varX);  
...  
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.19 InvertBit (RT Professional)

Beschreibung

Invertiert den Wert der angegebenen Variable vom Typ "Bool":

- Hat die Variable den Wert 1 (TRUE), wird sie auf 0 (FALSE) gesetzt.
- Hat die Variable den Wert 0 (FALSE), wird sie auf 1 (TRUE) gesetzt.

Verwendung in der Funktionsliste

InvertiereBit (Variable)

Verwendung in benutzerdefinierten Funktionen

InvertBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, dessen Bit gesetzt wird.

Beispiel

Der folgende Programmcode invertiert den Wert der booleschen Variable `b_value` und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `b_saved` aus.

```
{
BOOL b_value = 0;
BOOL b_saved = b_value;

//Invert variable
invertBit(b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
    ...
}
```

1.1.20 InvertBitInTag (RT Professional)

Beschreibung

Invertiert ein Bit in der angegebenen Variablen:

- Hat das Bit in der Variablen den Wert 1 (TRUE), wird es auf 0 (FALSE) gesetzt.
- Hat das Bit in der Variablen den Wert 0 (FALSE), wird es auf 1 (TRUE) gesetzt.

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie statt dessen die Systemfunktion "InvertiereBit".

Verwendung in der Funktionsliste

InvertiereBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

InvertBitInTag (Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Tag

Die Variable, in der das angegebene Bit gesetzt wird.

Bit

Die Nummer des Bits, das gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in einer Variablen von rechts nach links gezählt. Die Zählung beginnt mit 0.

Beispiel

Der folgende Programmcode invertiert ein Bit an der angegebenen Position `bitposition` in der Variable `bvalue` und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `bsaved` aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Invert bit in bitposition
InvertBitInTag (bvalue, bitposition);
//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

1.1.21 IsUserAuthorized (RT Professional)

Beschreibung

Überprüft, ob ein Bediener die angegebene Berechtigung besitzt.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL IsUserAuthorized (DWORD AuthorizationNumber);
```

Parameter

AuthorizationNumber

Die Berechtigung (numerisch), die überprüft werden soll.

Rückgabewert

TRUE

Der Bediener besitzt die angegebene Berechtigung.

FALSE

Der Bediener besitzt die angegebene Berechtigung nicht.

Beispiel

Der folgende Programmcode überprüft die Berechtigung des Benutzers mit der Funktion `IsUserAuthorized` und schreibt den Wert in die boolsche Variable `ok`.

```
{
BOOL ok;
DWORD authnumber;

//Check authorization and return value
ok = IsUserAuthorized (authnumber);

//error handling
if(ok)
{
    // user authorized
    printf ( "User is authorized." );
}
else
{
    // user not authorized
    printf ( "Authorization failed." );
}
...
}
```

Der Rückgabewert wird ausgegeben.

1.1.22 LinearScaling (RT Professional)

Beschreibung

Weist der Variablen Y einen Wert zu, der aus dem Wert der angegebenen Variablen X über die lineare Funktion $Y = (a * X) + b$ berechnet wird.

Die Umkehrfunktion ist die Systemfunktion "InvertiereLineareSkalierung".

Hinweis

Die Variablen X und Y dürfen nicht identisch sein. Wenn Sie eine Variable auf sich selbst umrechnen wollen, müssen Sie eine Hilfsvariable verwenden.

Mit der Systemfunktion "SetzeVariable" können Sie den Wert der umzurechnenden Variablen der Hilfsvariablen zuweisen.

Verwendung in der Funktionsliste

LineareSkalierung (Y, a, X, b)

Verwendung in benutzerdefinierten Funktionen

LinearScaling (Y, a, X, b)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Y

Die Variable, welcher der berechnete Wert aus der linearen Gleichung zugewiesen wird.

a

Der Wert, mit dem multipliziert wird.

X

Die Variable, deren Wert zur Berechnung verwendet wird.

b

Der Wert, der addiert wird.

Beispiel

Der folgende Programmcode weist mit der Funktion LinearScaling der Variable Yvar einen Wert zu.

```
{
BYTE Yvar;
BYTE Xvalue = 10;
BYTE bvalue = 3;
BYTE avalue = 4;

// linear scaling
LinearScaling ( Yvar, avalue, Xvalue, bvalue);

printf ("Yvar = %d\r\n", Yvar);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.23 ReportJob (RT Professional)

Beschreibung

Je nach Wert des Parameters Name of method wird ein Druckauftrag oder die Vorschau zu einem Druckauftrag gestartet.

Nur in C-Scripting verwendbar.

Syntax

```
void ReportJob(LPCSTR Print job name, LPCSTR Name of method)
```

Parameter

Print job name

Zeiger auf den Namen des Druckauftrags.

Name of method

Legt fest, ob der Druckauftrag oder die Vorschau zu einem Druckauftrag gestartet wird:

- PRINTJOB: Druckauftrag wird gestartet.
- PREVIEW: Vorschau zum Druckauftrag wird gestartet

Beispiel

Der folgende Programmcode führt je Inhalt der Variable printmethod eine Druckvorschau oder einen Ausdruck aus.

```
{
char* pszPrintjobName;
char* printmethod;

//Get the print method
printf("Input PRINTJOB for printing or PREVIEW for a quick view");
scanf ("%s",&printmethod);

//Print job or show preview
ReportJob(&PrintjobName, printmethod);

//error handling
if(printmethod=="PRINTJOB")
{
// message for printing completed
printf("printing done");
...
}
else
{
// User defined code if the
// job is a preview or failed
...
}
}
```

1.1.24 ResetBit (RT Professional)

Beschreibung

Setzt den Wert einer Variablen vom Typ "Bool" auf 0 (FALSE).

Verwendung in der Funktionsliste

RücksetzeBit (Variable)

Verwendung in benutzerdefinierten Funktionen

ResetBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable vom Typ BOOL, die auf 0 (FALSE) gesetzt wird.

Beispiel

Der folgende Programmcode setzt den Wert der booleschen Variable b_value mit der Funktion ResetBit auf 0 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert b_saved aus.

```
{
BOOL b_value = 1;
BOOL b_saved = b_value;

//Reset bit
ResetBit (b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

1.1.25 ResetBitInTag (RT Professional)

Beschreibung

Setzt ein Bit in der angegebenen Variablen auf 0 (FALSE).

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie stattdessen die Systemfunktion "RücksetzeBit".

Verwendung in der Funktionsliste

RücksetzeBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

ResetBitInTag (Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in der ein Bit auf 0 (FALSE) gesetzt wird.

Bit

Die Nummer des Bits, das auf 0 (FALSE) gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in der angegebenen Variablen unabhängig von der verwendeten Steuerung von rechts nach links gezählt. Die Zählung beginnt mit 0.

Beispiel

Der folgende Programmcode setzt ein Bit an der angegebenen Position bitposition der Variablen bvalue auf 0 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert bsaved aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Reset bit in bitposition
ResetBitInTag (bvalue, bitposition);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

1.1.26 SetBit (RT Professional)

Beschreibung

Setzt den Wert einer Variablen vom Typ "Bool" auf 1 (TRUE).

Verwendung in der Funktionsliste

SetzeBit (Variable)

Verwendung in benutzerdefinierten Funktionen

SetBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable vom Typ BOOL, die auf 1 (TRUE) gesetzt wird.

Beispiel

Der folgende Programmcode setzt den Wert der booleschen Variable b_value mit der Funktion SetBit auf 1 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert b_saved aus.

```
{
BOOL b_value = 0;
BOOL b_saved = b_value;

//Set bit
SetBit (b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

1.1.27 SetBitInTag (RT Professional)

Beschreibung

Setzt ein Bit in der angegebenen Variablen auf 1 (TRUE).

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie statt dessen die Systemfunktion "SetzeBit".

Verwendung in der Funktionsliste

SetzeBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

SetBitInTag(Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in der ein Bit auf 1 (TRUE) gesetzt wird.

Bit

Die Nummer des Bits, das auf 1 (TRUE) gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in der angegebenen Variablen unabhängig von der verwendeten Steuerung von rechts nach links gezählt. Die Zählung beginnt mit 0.

Hinweis

Eine notwendige Bedingung für eine zuverlässige Funktionalität ist eine garantierte Aktualisierung der verwendeten Variablen mit den aktuellen Prozesswerten. Projektieren Sie die Variable deswegen in einem E/A-Feld oder die Systemfunktion an einem Bildobjekt z. B. einer Schaltfläche.

Wenn Sie die Systemfunktion an ein kurzzeitiges Ereignis z. B. Kommen einer Meldung projiziert haben, dann erreichen Sie aktuelle Prozesswerte nur, wenn Sie die Variable auf dauerhaft lesen setzen.

Beispiel

Der folgende Programmcode setzt ein Bit an der angegebenen Position bitposition der Variablen bvalue auf 1 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert bsaved aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Reset bit in bitposition
SetBitInTag (bvalue, bitposition);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

1.1.28 SetLanguageByLocaleID (RT Professional)

Beschreibung

Ändert die Spracheinstellung im Runtime.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL SetLanguageByLocaleID (DWORD dwLocaleID);
```

Parameter

dwLocaleID
Sprachkennung der einzustellenden Sprache.

Rückgabewert

TRUE
Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE
Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetLanguage die aktuelle RuntimeSprache auf Deutsch und speichert den Rückgabewert in der Variable ok.

```
{
BOOL ok;
DWORD old_language;
DWORD new_language;

//Get the current language and save it
old_language = GetLanguageByLocaleID ();

//Set the current language to German
ok = SetLanguage(0x0407);

//Get the current language and save it
new_language = GetLanguageByLocaleID ();

//error handling
if(ok)
{
    // succeeded
    printf ( "RT language is now German." );
}
else
{
    // failed
    printf ( "RT language was not updated." );
}
//print language code
printf ("Former language code: %d\r\n", old_language);
printf ("Current language code: %d\r\n", new_language);
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

Siehe auch

GetLanguageByLocaleID (Seite 34)

1.1.29 SetProp (RT Professional)

1.1.29.1 SetPropBOOL (RT Professional)

Beschreibung

Setzt eine Objekteigenschaft mit dem Datentyp "BOOL".

Nur in C-Scripting verwendbar.

Syntax

BOOL SetPropBOOL(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property, BOOL Value)

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Value

Wert, welcher der Objekteigenschaft vom Datentyp "BOOL" zugewiesen wird.

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetPropBool die Eigenschaft des Objekts gs_graph_iofield auf "Sichtbar". Der Rückgabewert wird in der Variable ok gespeichert.

```
{
BOOL ok;

//Set the visibility TRUE
ok = SetPropBOOL("gs_graph_iofield","IOField1","Visible",TRUE);

//error handling
if(ok)
{
    // succeeded
    printf ( "IO field is visible." );
}
else
{
    // failed
    printf ( "Error - visibility not set" );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.29.2 SetPropChar (RT Professional)

Beschreibung

Setzt eine Objekteigenschaft mit dem Datentyp "char".
Nur in C-Scripting verwendbar.

Syntax

BOOL SetPropChar(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property, char* Value)

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Value

Wert, welcher der Objekteigenschaft vom Datentyp "char" zugewiesen wird.

Rückgabewert**TRUE**

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetPropChar die Eigenschaft Tooltiptext des Objekts gs_graph_iofield auf den Wert "Tooltiptext 1". Der Rückgabewert wird in der Variable ok gespeichert.

```
{
BOOL ok;

//Set the property Tooltiptext
ok = SetPropChar("Screen_1","gs_graph_iofield","ToolTipText","Tooltiptext 1");

//error handling
if(ok)
{
    // succeeded
    printf ( "Property of Tooltiptext is now Tooltiptext 1." );
}
else
{
    // failed
    printf ( "Error - property not set" );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.29.3 SetPropDouble (RT Professional)**Beschreibung**

Setzt eine Objekteigenschaft mit dem Datentyp "Double".

Nur in C-Scripting verwendbar.

Syntax

BOOL SetPropDouble(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property, double Value)

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Value

Wert, welcher der Objekteigenschaft vom Datentyp "double" zugewiesen wird.

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode greift auf eine Objekteigenschaft im Bild screenName zu. In diesem Beispiel setzt die Funktion SetPropDouble die Eigenschaft (Radius) des Kreises "Circle_1" auf den Wert 10. Der Rückgabewert wird in der Variable ok gespeichert.

```
{
BOOL ok;

//Set the property of circle
ok = SetPropDouble(screenName,"Circle_1","Radius",10);

//error handling
if(ok)
{
    // succeeded
    printf ( "Radius was set." );
}
else
{
    // failed
    printf ( "Error - radius not set" );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.29.4 SetPropLong (RT Professional)

Beschreibung

Setzt eine Objekteigenschaft mit dem Datentyp "long".

Nur in C-Scripting verwendbar.

Syntax

BOOL SetPropLong(LPCTSTR Screen name, LPCTSTR Object, LPCTSTR Name of the property, long Value)

Parameter

Screen name

Name des Bildes.

Object

Name des Objekts. Wenn sich der Aufruf der Funktion auf eine Eigenschaft des Bild-Objekts bezieht, muss der Parameter Object = NULL gesetzt werden.

Name of the Property

Name der Objekteigenschaft.

Value

Wert, welcher der Objekteigenschaft vom Datentyp "long" zugewiesen wird.

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropLong die Vordergrundfarbe eines Objekts: Im Bild "Screen_1" wird die Eigenschaft "ForeColor" des darin enthaltenen Objekts "Button1" auf den Wert 65333 (rot) gesetzt. Der Rückgabewert wird in der Variable ok gespeichert.

```
{
BOOL ok;

//Set the property Tooltiptext
ok = SetPropLong("Screen_1","Button1","ForeColor",65333);

//error handling
if(ok)
{
    // succeeded
    printf ( "Color was set." );
}
else
{
    // failed
    printf ( "Error - color not set" );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.30 SetPropertyByConstant (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft als Zeichenkette fest.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchKonstante (Bildname, Objekt, Name der Eigenschaft, Wert)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByConstant

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes ändern wollen, muss der Parameter "Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

```
SetPropertyByConstant("<Bildname>", "", "<Eigenschaftsname>", "<Wert>")
```

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft geändert wird.

Name der Eigenschaft

Name der Eigenschaft, die geändert wird.

Wert

Der Wert, welcher der Eigenschaft zugewiesen wird.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByConstant die Eigenschaft eines Objekts: Im Bild "Trends" wird die Eigenschaft "ToolBarButtonClick" des darin enthaltenen Objekts "Control_1" auf den Wert 26 gesetzt.

```
{  
  
//Name of the picture: Trends  
//Name of the f(t) trend view control: Control_1  
  
SetPropertyByConstant ("Trends", "Control_1", "ToolBarButtonClick", "26");  
  
// User defined code  
...  
}
```

Beispiel: Bildeigenschaft ändern

Der folgende Programmcode ändert mit der Funktion SetPropertyByConstant die Eigenschaft eines Bilds: Im Bild "Trends" wird die Eigenschaft "BackColor" auf den Wert 255 gesetzt.

```
{  
  
//Name of the picture: Trends  
  
SetPropertyByConstant ("Trends", "Trends", "BackColor", "255");  
  
// User defined code  
...  
}
```

Alternativ verwenden Sie anstelle des zweiten Parameters (Objekt) das Kennwort NULL oder einen Leerzeichen-String.

1.1.31 SetPropertyByProperty (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft durch eine andere Objekteigenschaft fest.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchEigenschaft (Bildname, Objekt, Name der Eigenschaft, Ziel Bildname, Ziel Objekt, Ziel Eigenschaftsname)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByProperty (Screen_name, Screen_object, Property_name, Source_screen_name, Source_screen_object, Source_property_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes durch eine andere Bildeigenschaft festlegen wollen, müssen die Parameter "Objekt" und "Ziel Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

```
SetPropertyByProperty  
("<Bildname>", "", "<Eigenschaftsname>", "<Ziel_Bildname>", "", "<Ziel_Eigenschaftsname>")
```

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft an das Zielobjekt übertragen wird.

Name der Eigenschaft

Name der Eigenschaft, an das Zielobjekt übertragen wird.

Ziel Bildname

Name des Bildes, in dem das Zielobjekt enthalten ist.

Ziel Objekt

Name des Zielobjekts, an das die Eigenschaft übertragen wird.

Ziel Eigenschaftsname

Name der Eigenschaft, die geändert wird.

Beispiel

Der folgende Programmcode überträgt mit der Funktion SetPropertyByProperty die Eigenschaft "ToolBarButtonClick" des Objekts "Control_1" im Ausgangsbild "Trend_1" : auf die entsprechende Eigenschaft im Zielbild "Trend_2".

```
{  
  
//Name of source picture: Trend_1  
//Name of target picture: Trend_2  
//Name of the f(t) trend view control: Control_1  
  
SetPropertyByProperty ("Trend_1", "Control_1", "ToolBarButtonClick", "Trend_2",  
"Control_2", "ToolBarButtonClick");  
  
// User defined code  
...  
}
```

1.1.32 SetPropertyByTag (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft durch einen Variablenwert fest.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchVariable (Bildname, Objekt, Name der Eigenschaft, Variablenname)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByTag (Screen_name, Screen_object, Property_name, Tag_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes durch einen Variablenwert festlegen wollen, muss der Parameter "Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

SetPropertyByTag ("**<Bildname>**", "", "**<Eigenschaftsname>**", "**<Variablenname>**")

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft durch den Variablenwert gesetzt wird.

Name der Eigenschaft

Name der Eigenschaft, die durch den Variablenwert gesetzt wird.

Variablenname

Name der Variablen, die den Wert der Eigenschaft enthält.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByTag die Eigenschaft eines Objekts: Beim Klicken auf das Objekt wird der Objektname und das Bild, in dem sich das Objekt befindet übergeben. Die Überschrift CaptionText im Bildfenster erhält den Wert der Variable HMI_value_1.

```
{  
  
SetPropertyByTag (screenName, objectName, "CaptionText", "HMI_value_1");  
  
// User defined code  
...  
}
```

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByTag die Eigenschaft eines Objekts: Im Bild "Trends" wird die Eigenschaft "ToolBarButtonClick" des darin enthaltenen Objekts "Control_1" auf den Wert 26 gesetzt.

```
{  
  
//Name of the picture: Trends  
//Name of the f(t) trend view control: Control_1  
  
SetPropertyByConstant ("Trends", "Control_1", "ToolBarButtonClick", "26");  
  
// User defined code  
...  
}
```

1.1.33 SetPropertyByTagIndirect (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft durch eine Variable fest. Die Variable enthält den Variablennamen, der die Objekteigenschaft festlegt.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchVariableIndirekt (Bildname, Objekt, Name der Eigenschaft, Variablenname)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByTagIndirect (Screen_name, Screen_object, Property_name, Tag_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes indirekt durch eine Variable festlegen wollen, muss der Parameter "Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

```
SetPropertyByTagIndirect ("<Bildname>", "", "<Eigenschaftsname>", "<Variablenname>")
```

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft durch den Variablenwert gesetzt wird.

Name der Eigenschaft

Name der Eigenschaft, die durch den Variablenwert gesetzt wird.

Variablenname

Name der Variablen, die wiederum den Namen der Variablen enthält, welche die Objekteigenschaft festlegt.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByTagIndirect die Eigenschaft eines Objekts: .

```
{
SetPropertyByTagIndirect (GetParentScreen(screenName), GetParentScreenWindow(screenName),
"ScreenName", "HMI_value_1");

// User defined code
...
}
```

1.1.34 SetTag (RT Professional)**1.1.34.1 SetTag-Funktionen (RT Professional)****Beschreibung**

Die SetTagXXX-Funktion setzt den Wert einer Variablen vom angegebenen Datentyp.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen SetTag-Funktionen zum Setzen des Variablenwerts:

Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
SetTagBit	Tag Tag_Name, short int value	Binäre Variable	Bool
SetTagByte	Tag Tag_Name, BYTE value	Vorzeichenlos 8-bit	USInt
SetTagDateTime	Tag Tag_Name, SYSTEMTIME va- lue	DTL	DateTime
SetTagChar	Tag Tag_Name, LPSTR value	Textvariable 8-bit oder Textvari- able 16-bit	String oder WString
SetTagDouble	Tag Tag_Name, double value	Gleitkommazahl 64-bit	LReal
SetTagDWord	Tag Tag_Name, DWORD value	Vorzeichenlos 32-bit	UDInt
SetTagFloat	Tag Tag_Name, float value	Gleitkommazahl 32-bit	Real
SetTagRaw	Tag Tag_Name, BYTE* pValue, DWORD size	Rohdatentyp	Raw
SetTagSByte	Tag Tag_Name, char value	Vorzeichenbehaftet 8-bit	SInt

Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
SetTagSDWord	Tag Tag_Name, long int value	Vorzeichenbehaftet 32-bit	DInt
SetTagSWord	Tag Tag_Name, short int value	Vorzeichenbehaftet 16-bit	Int
SetTagWord	Tag Tag_Name, WORD value	Vorzeichenlos 16-bit	UInt

Syntax

BOOL<Funktionsname><(Parameter)>;

Beispiel: BOOL SetTagBit (Tag Tag_Name, short int value);

Parameter

Tag_Name

Name der Variablen

value

Wert der Variablen im angegebenen Datentyp.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTagBit den Wert der Variablen gs_tag_bit auf TRUE und speichert den Rückgabewert in der Variable ok.

```
{
BOOL ok;
BOOL bvalue;

//Set the tag to true
ok = SetTagBit("gs_tag_bit", TRUE);
//error handling
if(ok)
{
    // succeeded
    printf ( "Function has run through.\r\n" );
bvalue = GetTagBit("gs_tag_bit");
    printf ("Value of gs_tag_bit: %d\r\n", bvalue);
}
else
{
    // failed
    printf ( "Error - function failed." );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.34.2 SetTagDateTime (RT Professional)

Funktion

Setzt den Wert einer Variablen vom Datentyp "Datum/Uhrzeit".

Syntax

```
BOOL SetTagDateTime(Tag Tag_Name, SYSTEMTIME value);
```

Parameter

Tag_Name

Name der Variablen

value

Wert der Variablen im Datentyp "Datum/Uhrzeit".

Rückgabewert

TRUE

Die Funktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

1.1.34.3 SetTagMultiStateWait-Funktionen (RT Professional)

Beschreibung

Setzt die Werte mehrerer Variablen. Die Systemfunktion wird erst beendet, nachdem das AS die Übernahme des Wertes zurückgemeldet hat.

Die Systemfunktion kann nur in C-Scripting verwendet werden.

Der Systemfunktion muss ein DWORD Array übergeben werden, in dessen Member sich nach dem Aufruf der Systemfunktion die Status der einzelnen Variablen befinden. Das Array muss so groß gewählt werden, dass für diese Status genügend Speicherplatz zur Verfügung steht.

Syntax

```
BOOL SetTagMultiStateWait(DWORD* pdwState, const char* pFormat,...);
```

Parameter

pdwState

Feld, in dem die Variablenstatus gespeichert werden

pFormat

Formatbeschreibung für alle angeforderten Variablen und für jede Variable Name und Wert

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTagMultiStateWait den Wert mehrerer Variablen.

1. Anlegen eines DWord-Arrays mit der benötigten Größe (Anzahl der Variablen)
2. Anlegen der Variablen, deren Werte mit der Funktion SetTagMultiStateWait auf die WinCC-Variablen übertragen werden
3. Beschreiben der WinCC-Variablen mit den Werten der eben deklarierten Variablen:
 - gs_tag_bit mit dem Wert der Variable "lValue1"
 - gs_tag_SByte mit dem Wert der Variable in der Adresse "&lValue2"
 - usw.

```
{
#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);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.34.4 SetTagMultiWait-Funktionen (RT Professional)

Beschreibung

Die Werte mehrerer Variablen werden im angegebenen Format gesetzt. Die Systemfunktion wird erst beendet, nachdem das AS die Übernahme des Wertes zurückgemeldet hat

Nur in C-Scripting verwendbar.

Syntax

```
BOOL SetTagMultiWait(const char* pFormat,...);
```

Parameter

pFormat

Formatbeschreibung für alle angeforderten Variablen und für jede Variable Name und Wert

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetTagMultiWait den Wert mehrerer Variablen. Der Rückgabewert wird in der Variable ok gespeichert.

```
{
BOOL ok;
//memory for values allocated via SysMalloc
DWORD dwVar1Value;
char* szVar2Value;
double dbVar3Value;

//settings
ok=SetTagMultiWait("%d%s%f", "Ernie_word", 16,
  "Ernie_char", "Hallo Welt",
  "Ernie_double", 55.4711);

//error handling
if(ok)
{
  // succeeded
  printf ( "Function has run through.\r\n" );

  // Get values and print
  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);
}
else
{
  // failed
  printf ( "Error - function failed." );
}
...
}
```

Beispiel

Der folgende Programmcode liest mit der Funktion GetTagMultiWait mehrere Variablen verschiedenen Typs.

1. Deklarieren von drei Variablen als Speicher für drei verschiedene Variablentypen
2. Deklarieren der booleschen Variablen ok zum Zwischenspeichern des Rückgabewertes (TRUE/FALSE)
3. Auslesen der drei Variablen und speichern der Werte in den zugehörigen Adressen. Der Rückgabewert der Funktion wird in der Variable ok gespeichert.
4. Ausgabe der drei Variablen mit vorangestelltem Variablentyp

1.1 Systemfunktionen (RT Professional)

```
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);
```

1.1.34.5 SetTagState-Funktionen (RT Professional)

Beschreibung

Setzt den Wert einer Variablen vom angegebenen Datentyp. Zusätzlich wird der Status der Variablen zurückgegeben.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen SetTagStateXXX-Funktionen zum Setzen des Variablenwerts:

Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
SetTagBitState	Tag Tag_Name, short int value, PDWORD lp_dwstate	Binäre Variable	Bool
SetTagByteState	Tag Tag_Name, BYTE value, PDWORD lp_dwstate	Vorzeichenlos 8-bit	UByte
SetTagCharState	Tag Tag_Name, LPSTR value, PDWORD lp_dwstate	Textvariable 8-bit oder Textvariable 16-bit	String
SetTagDoubleState	Tag Tag_Name, double value, PDWORD lp_dwstate	Gleitkommazahl 64-bit	Double
SetTagDWordState	Tag Tag_Name, DWORD value, PDWORD lp_dwstate	Vorzeichenlos 32-bit	UInteger
SetTagFloatState	Tag Tag_Name, float value, PDWORD lp_dwstate	Gleitkommazahl 32-bit	Float
SetTagRawState	Tag Tag_Name, BYTE* pValue, DWORD size, PDWORD lp_dwstate	Rohdatentyp	Raw

Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
SetTagSByteState	Tag Tag_Name, signed char value, PDWORD Ip_dwstate	Vorzeichenbehaftet 8-bit	Byte
SetTagSDWordState	Tag Tag_Name, long int value, PDWORD Ip_dwstate	Vorzeichenbehaftet 32-bit	Integer
SetTagSWordState	Tag Tag_Name, short int value, PDWORD Ip_dwstate	Vorzeichenbehaftet 16-bit	Short
SetTagWordState	Tag Tag_Name, WORD value, PDWORD Ip_dwstate	Vorzeichenlos 16-bit	UShort

Syntax

BOOL <Funktionsname><(Parameter)>;

Beispiel: BOOL SetTagBitState (Tag Tag_Name, short int value, PDWORD Ip_dwstate);

Parameter

Tag_Name

Name der Variablen

value

Wert der Variablen im angegebenen Datentyp.

Ip_dwstate

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Status der Variablen abgelegt wird.

pFormat

Formatbeschreibung für alle angeforderten Variablen und für jede Variable Name und Wert.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTagBitState den Wert der Variablen gs_tag_bit auf TRUE und speichert den Rückgabewert in der Variable ok. "&dwstate" ist die Adresse der Variablen, in die der Variablenstatus abgelegt wird.

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

```
{
DWORD dwstate;
BOOL ok;

//Load dwState with default values
dwstate = 0xFFFFFFFF;
//Set the value of the tag to TRUE
//dwstate is the tag state
ok = SetTagBitState("gs_tag_bit",TRUE,&dwstate);

//error handling
if(ok)
{
    // succeeded
    printf ( "Function has run through.\r\n" );
    printf ("Status of gs_tag_bit: %d\r\n", dwstate);
}
else
{
    // failed
    printf ( "Error - function failed." );
}
...
}
```

1.1.34.6 SetTagStateWait-Funktionen (RT Professional)

Beschreibung

Setzt den Wert einer Variablen vom angegebenen Datentyp. Die Systemfunktion wird erst beendet, nachdem das AS die Übernahme des Wertes zurückgemeldet hat. Zusätzlich wird der Status der Variablen zurückgegeben.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen SetTagStateWait-Funktionen zum Setzen des Variablenwerts:

Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
SetTagBitStateWait	Tag Tag_Name, short int value, PDWORD lp_dwstate	Binäre Variable	Bool
SetTagByteStateWait	Tag Tag_Name, BYTE value, PDWORD lp_dwstate	Vorzeichenlos 8-bit	UByte
SetTagCharStateWait	Tag Tag_Name, LPSTR value, PDWORD lp_dwstate	Textvariable 8-bit oder Textvariable 16-bit	String
SetTagDoubleStateWait	Tag Tag_Name, double value, PDWORD lp_dwstate	Gleitkommazahl 64-bit	Double
SetTagDWordStateWait	Tag Tag_Name, DWORD value, PDWORD lp_dwstate	Vorzeichenlos 32-bit	UInteger
SetTagFloatStateWait	Tag Tag_Name, float value, PDWORD lp_dwstate	Gleitkommazahl 32-bit	Float
SetTagRawStateWait	Tag Tag_Name, BYTE* pValue, DWORD size, PDWORD lp_dwstate	Rohdatentyp	Raw
SetTagSByteStateWait	Tag Tag_Name, char value, PDWORD lp_dwstate	Vorzeichenbehaftet 8-bit	Byte
SetTagSDWordStateWait	Tag Tag_Name, long int value, PDWORD lp_dwstate	Vorzeichenbehaftet 32-bit	Integer
SetTagSWordStateWait	Tag Tag_Name, short int value, PDWORD lp_dwstate	Vorzeichenbehaftet 16-bit	Short
SetTagWordStateWait	Tag Tag_Name, WORD value, PDWORD lp_dwstate	Vorzeichenlos 16-bit	UShort

Syntax

BOOL<Funktionsname><(Parameter)>;

Beispiel: BOOL SetTagBitStateWait (Tag Tag_Name, short int value, PDWORD lp_dwstate);

Parameter

Tag_Name

Name der Variablen

value

Wert der Variablen im angegebenen Datentyp.

lp_dwstate

Zeiger auf ein DWORD, in dem nach dem Durchlauf der Systemfunktion der Status der Variablen abgelegt wird.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTagBitStateWait den Wert der Variablen `gs_tag_bit` auf TRUE und speichert den Rückgabewert in der Variable `ok`. "`&dwstate`" ist die Adresse der Variablen, in die der Variablenstatus abgelegt wird.

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

```
{
DWORD dwstate;
BOOL ok;

//Load dwState with default values
dwstate = 0xFFFFFFFF;
//Set the value of the tag to TRUE
//dwstate is the tag state
ok = SetTagBitStateWait("gs_tag_bit",TRUE,&dwstate);

//error handling
if(ok)
{
    // succeeded
    printf ( "Function has run through.\r\n" );
    printf ("Status of gs_tag_bit: %d\r\n", dwstate);
}
else
{
    // failed
    printf ( "Error - function failed." );
}
...
}
```

1.1.34.7 SetTagValue-Funktionen (RT Professional)

Beschreibung

Ermöglicht die Übergabe eines Werts in Form eines Variants und setzt den Zeiger auf den Wert vom Datentyp "Variant".

Nur in C-Scripting verwendbar.

Syntax

BOOL SetTagValue (LPDM_VARKEY lpdmVarKey, LPVARIANT lpdmValue, PDWORD dwState, LPCMN_ERROR lpdmError);

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Datentyp "DM_VARKEY"

lpdmValue

Zeiger auf den Wert vom Datentyp "Variant". Die Beschreibung des Datentyps VARIANT finden Sie in der einschlägigen Fachliteratur.

lpdmError

Zeiger auf die Struktur, welche die Fehlerbeschreibung enthält

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode übergibt mit der Funktion SetTagValue den Wert in varKey.

Je nach Rückgabewert in keyFound (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```
{
// tags for setting the value
DM_VARKEY varKey;
LPVARIANT value;
LPCMN_ERROR error1;

// tags for getting the value
DM_VAR_UPDATE_STRUCT result;
CMN_ERROR error;

BOOL keyFound;

ok = SetTagValue(&varKey, &value, &error1);

if (keyFound)
{
// succeeded, get the new value
GetTagValue(&varKey, &result, &error);
// print tag value
printf ("Value of varKey: %d\r\n", &varKey);
...
}
else
{
// failed
printf ("Error - function failed." );
...
}
}
```

1.1.34.8 SetTagValueWait-Funktionen (RT Professional)

Beschreibung

Ermöglicht die Übergabe eines Wertes in Form eines Variants und setzt den Zeiger auf den Wert vom Datentyp "Variant". Die Systemfunktion wird erst beendet, nachdem das AS die Übernahme des Wertes zurückgemeldet hat.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL SetTagValueWait(LPDM_VARKEY lpdmVarKey, LPVARIANT lpdmValue, PDWORD dwState, LPCMN_ERROR lpdmError);
```

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Datentyp "DM_VARKEY"

lpdmValue

Zeiger auf den Wert vom Datentyp "Variant". Die Beschreibung des Datentyps VARIANT finden Sie in der einschlägigen Fachliteratur.

dwState

Status der Variablen, der nach Durchlaufen der Systemfunktion zurückgeliefert wird

lpdmError

Zeiger auf die Struktur, welche die Fehlerbeschreibung enthält

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode übergibt mit der Funktion SetTagValueWait den Wert in varKey.

Je nach Rückgabewert in keyFound (TRUE/FALSE) wird ein spezifischer Code ausgeführt.

```

{
// tags for setting the value
DM_VARKEY varKey;
LPVARIANT value;
LPCMN_ERROR error1;

// tags for getting the value
DM_VAR_UPDATE_STRUCT result;
CMN_ERROR error;

BOOL keyFound;

ok = SetTagValueWait(&varKey, &value, &error1);

if (keyFound)
{
// succeeded, get the new value
GetTagValueWait(&varKey, &result, &error);
// print tag value
printf ("Value of varKey: %d\r\n", &varKey);
...
}
else
{
// failed
printf ( "Error - function failed." );
...
}
}

```

1.1.34.9 SetTagWait-Funktionen (RT Professional)

Beschreibung

Setzt den Wert einer Variablen vom angegebenen Datentyp. Die Systemfunktion wird erst beendet, nachdem das AS die Übernahme des Wertes zurückgemeldet hat.

Nur in C-Scripting verwendbar.

Die folgende Tabelle zeigt die verschiedenen SetTagWait-Funktionen zum Setzen des Variablenwerts:

Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
SetTagBitWait	Tag Tag_Name, short int value	Binäre Variable	Bool
SetTagByteWait	Tag Tag_Name, BYTE value	Vorzeichenlos 8-bit	UByte
SetTagCharWait	Tag Tag_Name, char* value	Textvariable 8-bit oder Textvariable 16-bit	String

Funktionsname	Parameter	PLC Datentyp	HMI Datentyp
SetTagDouble- Wait	Tag Tag_Name, double value	Gleitkommazahl 64-bit	Double
SetTagDWord- Wait	Tag Tag_Name, DWORD value	Vorzeichenlos 32-bit	UInteger
SetTagFloat- Wait	Tag Tag_Name, float va- lue	Gleitkommazahl 32-bit	Float
SetTagRaw- Wait	Tag Tag_Name, BYTE* pValue, DWORD size	Rohdatentyp	Raw
SetTagSByte- Wait	Tag Tag_Name, char va- lue	Vorzeichenbehaftet 8-bit	Byte
SetTagSDWord- Wait	Tag Tag_Name, long int value	Vorzeichenbehaftet 32-bit	Integer
SetTagSWord- Wait	Tag Tag_Name, short int value	Vorzeichenbehaftet 16-bit	Short
SetTagWord- Wait	Tag Tag_Name, WORD value	Vorzeichenlos 16-bit	UShort

Syntax

BOOL <Funktionsname><(Parameter)>;

Beispiel: BOOL SetTagBitWait (Tag Tag_Name, short int value);

Parameter

Tag_Name

Name der Variablen

value

Wert der Variablen im angegebenen Datentyp.

pValue

Zeiger auf ein Byte-Feld, das den Wert der Rohdatenvariablen enthält

size

Größe des Byte-Feldes in Byte

Rückgabewert

TRUE

Die Systemfunktion selbst wurde fehlerfrei durchlaufen.

Es wird allerdings nicht geprüft, ob auch die Variable fehlerfrei geschrieben werden konnte.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTagBitWait den Wert der Variablen gs_tag_bit auf TRUE und speichert den Rückgabewert in der Variable ok.

```
{
BOOL ok;
BOOL bvalue;

//Set the tag to true
ok = SetTagBitWait("gs_tag_bit", TRUE);
//error handling
if(ok)
{
    // succeeded
    printf ( "Function has run through.\r\n" );
    bvalue = GetTagBitWait("gs_tag_bit");
    printf ("Value of gs_tag_bit: %d\r\n", bvalue);
}
else
{
    // failed
    printf ( "Error - function failed." );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.34.10 SetTag (RT Professional)

Beschreibung

Weist der angegebenen Variablen einen neuen Wert zu.

Hinweis

Abhängig vom Variablentyp können Sie mit dieser Systemfunktion Zeichenketten und Zahlen zuweisen.

Verwendung in der Funktionsliste

SetzeVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

SetTag (Tag, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, welcher der angegebene Wert zugewiesen wird.

Wert

Der Wert, welcher der angegebenen Variablen zugewiesen wird.

Hinweis

Die Systemfunktion "SetzeVariable" wird nur nach Aufbau einer Verbindung ausgeführt.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTag den Wert der Variablen gs_tag_bit auf TRUE und speichert den Rückgabewert in der Variable ok.

```
{
BOOL ok;
BOOL bvalue;

//Set the tag to true
ok = SetTag("gs_tag_bit", TRUE);
//error handling
if(ok)
{
    // succeeded
    printf ( "Function has run through.\r\n" );
    bvalue = GetTagBit("gs_tag_bit");
    printf ("Value of gs_tag_bit: %d\r\n", bvalue);
}
else
{
    // failed
    printf ( "Error - function failed." );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.34.11 SetTagByProperty (RT Professional)

Beschreibung

Legt einen Variablenwert durch den Wert einer Objekteigenschaft fest. Die Änderung wird zusätzlich im Meldesystem protokolliert

Verwendung in der Funktionsliste

SetzteVariableDurchEigenschaft (Variablenname, Bildname, Objekt, Name der Eigenschaft, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagByProperty (Tag_name, Screen_name, Screen_object, Property_name, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der Variablen, deren Wert durch die Objekteigenschaft festgelegt wird.

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft den Variablenwert liefert.

Name der Eigenschaft

Name der Eigenschaft, die den Variablenwert liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode gibt beim Klick auf ein Kombinationsfeld den Wert des ausgewählten Textes zurück.

```
{  
char* rt_value;  
  
SetTagByProperty (rt_value, screenName, objectName, "SelectedText",  
hmiWithoutOperatorEvent);  
  
...  
}
```

1.1.34.12 SetTagByTagIndirect (RT Professional)

Beschreibung

Legt den Variablenwert durch den Wert einer indirekten Variablen fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableDurchVariableIndirekt (Variablenname, Variablenname, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagByTagIndirect (Tag_name, Source_tag_name, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der Variable, deren Wert durch eine indirekte Variable festgelegt wird.

Variablenname

Name der indirekten Variablen, die den Variablenwert liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode gibt beim Klick auf eine Taste den Wert der Variable "@LocalMachineName" zurück.

```
{
char* rt_value;

SetTagByTagIndirect (rt_value, "@LocalMachineName", hmiWithoutOperatorEvent);

...
}
```

1.1.34.13 SetTagIndirect (RT Professional)

Beschreibung

Legt den indirekten Namen für eine Variable fest.

Verwendung in der Funktionsliste

SetzeVariableIndirekt (Variablenname (Ausgabe), LpValue)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirect (Tag_name, Value, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname (Ausgabe)

Name der Variablen, in die der Variablenname geschrieben wird.

LpValue

Name der Variablen, der in die Variable geschrieben wird.

Beispiel

Der folgende Programmcode überträgt beim Klick auf eine Taste den Wert der Variable "value" auf die Variable "result".

```
{
BYTE result;
BYTE value;

SetTagIndirect ("result", "value", hmiWithoutOperatorEvent);
...
}
```

1.1.34.14 SetTagIndirectByProperty (RT Professional)

Beschreibung

Legt einen Variablennamen durch den Wert einer Objekteigenschaft fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableIndirektDurchEigenschaft (Variablenname, Bildname, Bildobjekt, Name der Eigenschaft, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirectByProperty (Tag_name, Screen_name, Screen_object, Property_name, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der Variablen, deren Variablenname durch die Objekteigenschaft festgelegt wird.

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Bildobjekt

Name des Objekts, dessen Eigenschaft den Variablennamen liefert.

Name der Eigenschaft

Name der Eigenschaft, die den Variablennamen liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode setzt beim Klick auf die Taste objectName die Variable "rt_value_property" auf den Wert der Eigenschaft "FlashingEnabled" .

```
{
  Int rt_value_property;
  SetTagIndirectByProperty ("rt_value_property", screenName, objectName, "FlashingEnabled",
  hmiWithoutOperatorEvent);
  ...
}
```

1.1.34.15 SetTagIndirectByTagIndirect (RT Professional)

Beschreibung

Legt den Variablenwert durch den Wert einer indirekten Variablen fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableIndirektDurchVariableIndirekt (Variablenname, Variablenname, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirectByTagIndirect (Tag_name, Source_tag_name, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der indirekten Variable, deren Wert durch eine indirekte Variable festgelegt wird.

Variablenname

Name der indirekten Variablen, die den Variablenwert liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode setzt beim Klick auf die Taste objectName die Variable "rt_value" auf den Wert der Variable "value" .

```
{  
Int rt_value;  
Int value;  
  
SetTagIndirectByTagIndirect ("rt_value", "value", hmiWithoutOperatorEvent);  
...  
}
```

1.1.34.16 SetTagIndirectWithOperatorInputAlarm (RT Professional)

Beschreibung

Legt den indirekten Namen für eine Variable fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableIndirektMitBedienmeldung (Variablenname (Ausgabe), LpValue)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirectWithOperatorEvent

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname (Ausgabe)

Name der Variablen, in die der Variablenname geschrieben wird.

LpValue

Name der Variablen, der in die Variable geschrieben wird.

1.1.34.17 SetTagWithOperatorEvent (RT Professional)

Beschreibung

Legt den Wert für eine Variable fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableMitBedienmeldung (Variablenname (Ausgabe), LpValue)

Verwendung in benutzerdefinierten Funktionen

SetTagWithOperatorEvent (Tag_name, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname (Ausgabe)

Name der Variablen, deren Wert festgelegt wird.

LpValue

Der Wert, der in die Variable geschrieben wird.

Beispiel

Der folgende Programmcode überträgt beim Klick auf eine Taste den Wert der Variable "value" auf die Variable "result" .

```
{  
BYTE result;  
BYTE value;  
  
SetTagWithOperatorEvent ("result", "value");  
...  
}
```

1.1.35 StartProgram (RT Professional)

Beschreibung

Startet das angegebene Programm mit den angegebenen Parametern.

Die Systemfunktion kann nur in C-Scripting verwendet werden

Syntax

```
void StartProgram(Program_name,Program_parameter,Display_mode,  
Wait_for_program_end);
```

Parameter

Program_name

Pfad und Namen des zu startenden Programms.

Program_parameters

Parameter, die für den Start verwendet werden sollen. Informationen zu den möglichen Parametern finden Sie in der Beschreibung des zu startenden Programms.

Display_mode

Legt fest, in welchem Anzeigemodus das Programm gestartet wird:

0 (hmiShowNormal) = Anzeige in Fenster

1 (hmiShowMinimized) = Anzeige in minimiertem Fenster

2 (hmiShowMaximized) = Anzeige in maximiertem Fenster

3 (hmiShowMinimizedAndInactive) = Anzeige in inaktivem, minimierten Fenster

Wait_for_program_end

Der Parameter wird von WinCC Runtime Professional nicht ausgewertet.

Beispiel

Der folgende Programmcode startet das Programm calc.exe im minimiertem Fenster.

```
{  
  BOOL Wait_for_program_end;  
  float number;  
  
  //start the program calc.exe  
  StartProgram("C:\\Winnt\\system32\\calc.exe", number, hmiShowMinimized,  
  Wait_for_program_end);  
  ...  
}
```

1.1.36 StopRuntime (RT Professional)**Beschreibung**

Beendet die Runtime-Software und damit das laufende Projekt am Bediengerät.

Verwendung in der Funktionsliste

StoppeRuntime (Modus)

Verwendung in benutzerdefinierten Funktionen

StopRuntime (Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob nach dem Beenden von Runtime auch das Betriebssystem heruntergefahren wird:

0 (hmiStopRuntime) = Runtime: Betriebssystem wird nicht heruntergefahren

1 (hmiStopRuntimeAndOperatingSystem) = Runtime und Betriebssystem: Betriebssystem wird heruntergefahren (bei WinCE nicht möglich)

Beispiel

Der folgende Programmcode fährt Runtime und Betriebssystem herunter.

```
{  
  
//Stop runtime and shutdown  
StopRuntime (hmiStopRuntimeAndOperationSystem);  
  
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

1.1.37 StoreScreen (RT Professional)

Beschreibung

Speichert das aktuelle Bild. Dieses Bild kann mit der Systemfunktion ActivateStoredScreen geöffnet werden.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL StoreScreen();
```

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Beispiel

Der folgende Programmcode schreibt den Rückgabewert der Funktion StoreScreen in die Variable screen_stored und ruft das gespeicherte Bild auf, falls es fehlerfrei gespeichert wurde.

```
{
BOOL screen_stored;
screen_stored = StoreScreen();
//user defined code
...
//error handling
if(screen_stored)
{
    // succeeded
    ActivateStoredScreen();
    printf ( "Stored screen is now activated.\r\n" );
}
else
{
    // failed
    printf ( "Error - no screen stored." );
}
...
}
```

Siehe auch

ActivateStoredScreen (Seite 28)

1.1.38 TriggerOperatorEvent (RT Professional)

Beschreibung

Die Systemfunktion TriggerOperatorEvent löst eine Bedienmeldung aus.

Syntax

```
int TriggerOperatorEvent( DWORD dwFlags, DWORD dwMsgNum, char* lpszObjectName,
DWORD dwMyTextID, double doValueOld, double doValueNew, char* pszComment );
```

Parameter

dwFlags

FLAG_COMMENT_PARAMETER (0x001): Legt fest, dass die Eingabe des Kommentars über den Parameter erfolgt.

FLAG_COMMENT_DIALOG (0x003): Legt fest, dass die Eingabe des Kommentars über einen Dialog erfolgt.

FLAG_TEXTID_PARAMETER (0x100): Legt fest, dass der Kommentars über eine Text-ID festgelegt wird

dwMsgNum

Nummer der Bedienmeldung die ausgelöst wird.

lpzObjectName

Zeiger auf den Namen der Variablen mit dem alten und dem neuen Wert.

dwMyTextID

ID des Textes, der als Kommentar verwendet werden soll.

doValueOld

Alter Wert.

doValueNew

Neuer Wert.

pszComment

Zeiger auf den Text, der als Kommentar verwendet werden soll.

Rückgabewert

TRUE

Die Systemfunktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

1.1.39 UA (Recipe) (RT Professional)

1.1.39.1 uaAddArchive (RT Professional)

Beschreibung

Hinzufügen einer neuen Rezeptur. Dies entspricht der Projektierung einer neuen Rezeptur mit dem Editor "Rezepturen".

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaAddArchive (  
    UAHCONFIG hConfig,  
    UACONFIGARCHIVE* pArchive )
```

Parameter

UAHCONFIG hConfig

Konfigurations-Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

UACONFIGARCHIVE* pArchive

Zeiger auf Puffer zum Speicher der Rezeptur-Konfiguration

Rückgabewert

Index der neuen Rezeptur. Ein Fehler entspricht -1.

Siehe auch

uaQueryConfiguration (Seite 168)

1.1.39.2 uaAddField (RT Professional)

Beschreibung

Hinzufügen eines neuen Datenfelds.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaAddField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGFIELD* pField )
```

Parameter

UAHCONFIG hConfig,

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,
Archive-Index (0 bis (uaGetNumArchives()-1))

UACONFIGFIELD* pArchive
Zeiger auf Puffer der Feld-Konfiguration

Rückgabewert

Index des neuen Felds. Ein Wert von -1 zeigt einen Fehler an.

1.1.39.3 uaArchiveClose (RT Professional)

Beschreibung

Die Verbindung zur aktuellen Rezeptur wird aufgehoben.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveClose (  
    UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive
Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE
Erfolgreiches Schließen der Rezeptur

FALSE
Fehler

1.1.39.4 uaArchiveDelete (RT Professional)

Beschreibung

Löscht die Daten aus einer Rezeptur. Die projektierte Rezeptur bleibt aber erhalten.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveDelete (  
    UAHARCHIVE hArchive,  
    LPCSTR pszWhere )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPCSTR pszWhere

In diesem String steht der SQL- Selektionsausdruck. Dieser legt fest, welche Datensätze gelöscht werden sollen. Der Ausdruck ist der gleiche wie in der SQL-Anweisung "DELETE FROM <archive> WHERE pszWhere".

Achtung ! Wenn dieser String leer ist, wird die gesamte Rezeptur gelöscht.

Returnwert

TRUE

Erfolgreiches Löschen der Rezeptur

FALSE

Fehler

1.1.39.5 uaArchiveExport (RT Professional)

Beschreibung:

Exportiert die aktuelle Rezeptur in ein Archiv im CSV-Format.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveExport (  
    UAHARCHIVE hArchive,  
    LPCSTR pszDestination,  
    LONG lType,  
    LONG lOptions )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit UaQueryArchive oder UaQueryArchiveByName eingerichtet.

LPCSTR pszDestination

Dateiname des Zielarchivs

LONG lType

Datenformat des Zielarchivs. Es stehen zwei Formate zur Verfügung:

- UA_FILETYPE_DEFAULT = 0: Default Dateiformat = CSV
- UA_FILETYPE_CSV = 1: CSV Dateiformat

LONG lOptions

Für spätere Erweiterungen reserviert. Muss 0 sein.

Rückgabewert

TRUE

Erfolgreiches Exportieren der Rezeptur

FALSE

Fehler

1.1.39.6 uaArchiveGetCount (RT Professional)

Beschreibung

Liest die Anzahl der Datensätze.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaArchiveGetCount(  
    UAHARCHIVE hArchive,  
    LONG * plCount )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG plCount

Zeiger auf eine Variable, wo die Anzahl der Datensätze abgelegt werden soll.

Rückgabewert

Anzahl der Datensätze.

0 = Archiv leer oder Fehler aufgetreten. Abfrage mittels uaGetLastError() erforderlich.

1.1.39.7 uaArchiveGetFieldLength (RT Professional)

Beschreibung

Liest die Länge eines Felds im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaArchiveGetFieldLength(  
    UAHARCHIVE hArchive,  
    LONG lField )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

Rückgabewert

Länge des aktuellen Feldes

1.1.39.8 uaArchiveGetFieldName (RT Professional)

Beschreibung

Liest den Namen eines Felds im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
VOID uaArchiveGetFieldName (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LPCSTR pszName,  
    LONG cMaxLen )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

LPCSTR pszName

Feldname

LONG cMaxLen

Maximale Länge

1.1.39.9 uaArchiveGetFields (RT Professional)

Beschreibung

Liest die Anzahl der projizierten Datenfelder, wobei die Felder "ID", "Letzter Benutzer" und "Letzter Zugriff" mitgezählt werden. In den Runtimeaufrufen werden die Indizes der projizierten Felder mit 1 bis N angegeben. Das Feld ID hat den Index 0. Die Felder "Letzter Benutzer" und "Letzter Zugriff" schließen sich am Ende der projizierten Felder an.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaArchiveGetFields (  
    UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

Anzahl der projizierten Felder.

1.1.39.10 uaArchiveGetFieldType (RT Professional)**Beschreibung**

Liest den Typ eines Felds im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaArchiveGetFieldType (  
    UAHARCHIVE hArchive,  
    LONG lField )
```

Parameter**UAHARCHIVE hArchive**

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

Rückgabewert

Typ des aktuellen Felds

Die symbolischen Bezeichnungen für die Feldtypen sind:

UA_FIELDTYPE_INTEGER

UA_FIELDTYPE_DOUBLE

UA_FIELDTYPE_STRING

UA_FIELDTYPE_DATETIME

1.1.39.11 uaArchiveGetFieldValueDate (RT Professional)**Beschreibung**

Liest Datum und Uhrzeit eines Felds im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveGetFieldValueDate (
    UAHARCHIVE hArchive,
    LONG lField,
    LPSYSTEMTIME pstDateTime )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

LPSYSTEMTIME pstDateTime

Zeiger auf Variable vom Typ SYSTEMTIME

Rückgabewert

TRUE

Erfolgreiches Lesen von Datum und Uhrzeit

FALSE

Fehler

1.1.39.12 uaArchiveGetFieldValueDouble (RT Professional)

Beschreibung

Liest Double Wert eines Felds im aktuellen Datensatz.

Die Systemfunktion kann nur in C-Scripting verwendet werden.

```
BOOL uaArchiveGetFieldValueDouble (
    UAHARCHIVE hArchive,
    LONG lField,
    double* pdValue )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

double* pdValue

Zeiger auf Variable für den aktuellen Feldinhalt

Rückgabewert

TRUE

Erfolgreiches Lesen von Feldwert

FALSE

Fehler

1.1.39.13 uaArchiveGetFieldValueFloat (RT Professional)

Beschreibung

Liest Float Wert eines Felds im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveGetFieldValueFloat (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    FLOAT* pfValue )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

FLOAT* pfValue

Zeiger auf Float Variable für den aktuellen Feldinhalt

Rückgabewert

TRUE

Erfolgreiches Lesen von Feldwert

FALSE

Fehler

1.1.39.14 uaArchiveGetFieldValueLong (RT Professional)

Beschreibung

Liest Long Integer Wert eines Felds im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveGetFieldValueLong (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LONG* pdValue )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

LONG* pdValue

Zeiger auf Long Variable für den aktuellen Feldinhalt

Rückgabewert

TRUE

Erfolgreiches Lesen von Feldwert

FALSE

Fehler

1.1.39.15 uaArchiveGetFieldValueString (RT Professional)

Beschreibung

Liest String eines Felds im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveGetFieldValueString (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LPSTR pszString,  
    LONG cMaxLen )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste Feld adressiert.

LPCSTR pszString

Feldwert als String

LONG cMaxLen

Maximale Länge des Strings

Rückgabewert

TRUE

Erfolgreiches Lesen von Feldwert

FALSE

Fehler

1.1.39.16 uaArchiveGetFilter (RT Professional)

Beschreibung

Liest Filter des aktuellen Datensatzes. Weiteres erfahren Sie im Anhang, Kapitel "SQL-Anweisungen".

Nur in C-Scripting verwendbar.

Syntax

```
VOID uaArchiveGetFilter (  
    UAHARCHIVE hArchive,  
    LPSTR pszFilter,  
    LONG cMaxLen )
```

Die Systemfunktion kann nur in C-Scripting verwendet werden.

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPSTR pszFilter

Gelesener Filter

LONG cMaxLen

Maximale Länge

1.1.39.17 uaArchiveGetID (RT Professional)

Beschreibung

uaArchiveGetID liest die Kennung (ID) der Rezeptur.

Die ID der Rezeptur dient internen Zwecken und kann ungleich sein mit der in der in der Rezeptur angezeigten Nummer.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaArchiveGetID (  
    UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

ID der Rezeptur.

1.1.39.18 uaArchiveGetName (RT Professional)**Beschreibung**

Liest den Namen der Rezeptur.

Nur in C-Scripting verwendbar.

Syntax

```
VOID uaArchiveGetName (
    UAHARCHIVE hArchive,
    LPSTR pszName,
    LONG cMaxLen )
```

Parameter**UAHARCHIVE hArchive**

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPSTR pszName

Zeiger auf Puffer für Rezepturname

LONG cMaxLen

Maximale Länge

Beispiel

```
char Abfuellung [40];
uaArchiveGetName( hArchive, Abfuellung, 39 );
```

1.1.39.19 uaArchiveGetSor (RT Professional)**Beschreibung**

uaArchiveGetSort liest die Sortierung der Rezeptur.

Nur in C-Scripting verwendbar.

Syntax

```
VOID uaArchiveGetSort (  
    UAHARCHIVE hArchive,  
    LPSTR pszSort,  
    LONG cMaxLen )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPCSTR pszSort

Sortierung

LONG cMaxLen

Maximale Länge

1.1.39.20 uaArchiveImport (RT Professional)

Beschreibung

uaArchivImport importiert eine Rezeptur mit CSV-Datenformat. Die Struktur der Zielrezeptur muss identisch sein mit der importierten Rezeptur.

Syntax

```
BOOL uaArchiveImport (  
    UAHARCHIVE hArchive,  
    LPCSTR pszSource,  
    LONG lType,  
    LONG lOptions )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPCSTR pszSource

Dateiname des Quellarchivs

LONG lType

Datenformat des Quellarchivs. Es stehen zwei Formate zur Verfügung:

UA_FILETYPE_DEFAULT = 0: Default Dateiformat = CSV

UA_FILETYPE_CSV = 1: CSV Dateiformat

LONG lOptions

Für spätere Erweiterungen reserviert. Muss 0 sein.

Rückgabewert

TRUE

Erfolgreiches Importieren der Rezeptur

FALSE

Fehler

1.1.39.21 uaArchiveInsert (RT Professional)**Beschreibung**

Fügt den lokalen Datensatzpuffer in die aktuelle Datenbank ein. Damit sinnvolle Daten im neuen Datensatz stehen, müssen die Felder des lokalen Datensatzpuffers vor dem Aufruf von uaArchiveInsert mit den Systemfunktionen "uaArchiveSetFieldValue..." beschrieben werden. Die interne Spalte "ID" muss mit der Systemfunktion "uaArchiveSetFieldValueLong" in den aktuellen Datensatz geschrieben werden. Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveInsert (  
    UAHARCHIVE hArchive )
```

Parameter**UAHARCHIVE hArchive**

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Erfolgreiches Einfügen des Datensatzes.

1.1.39.22 uaArchiveMoveFirst (RT Professional)

Beschreibung

Gehe zum ersten Datensatz.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveMoveFirst (  
    UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Erfolgreiches Springen in der Rezeptur

FALSE

Fehler

1.1.39.23 uaArchiveMoveLast (RT Professional)

Beschreibung

Gehe zum letzten Datensatz.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveMoveLast (  
    UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert**TRUE**

Erfolgreiches Springen in der Rezeptur

FALSE

Fehler

1.1.39.24 uaArchiveMoveNext (RT Professional)**Beschreibung**

Gehe zum nächsten Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveMoveNext (  
    UAHARCHIVE hArchive )
```

Parameter**UAHARCHIVE hArchive**

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert**TRUE**

Erfolgreiches Springen in der Rezeptur

FALSE

Fehler

1.1.39.25 uaArchiveMovePrevious (RT Professional)**Beschreibung**

Gehe zum vorherigen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveMovePrevious (
```

```
UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Erfolgreiches Springen in der Rezeptur

FALSE

Fehler

1.1.39.26 uaArchiveOpen (RT Professional)

Beschreibung

uaArchiveOpen muss vor allen RT-Funktionen aufgerufen werden (z. B. uaArchiveMoveFirst, uaArchiveMoveLast, uaArchiveMoveNext, uaArchiveMovePrevious, uaArchiveDelete, uaArchiveUpdate, uaArchiveInsert, uaArchiveGetID, uaArchiveGetFields, uaArchiveGetFieldType, uaArchiveGetFieldValueDate, uaArchiveGetFieldValueDouble, uaArchiveGetFieldValueFloat, uaArchiveGetFieldValueLong, uaArchiveGetFieldValueString, uaArchiveSetFieldValueDate, uaArchiveSetFieldValueDouble, uaArchiveSetFieldValueFloat, uaArchiveSetFieldValueLong und uaArchiveSetFieldValueString).

Hinweis

Rezepturen sortieren und filtern

Die Systemfunktionen "uaArchiveSetSort" und "uaArchiveSetFilter" können Sie auf eine Rezeptur anwenden, ohne die Rezeptur mit "uaArchiveOpen" zu öffnen.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveOpen (  
UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Erfolgreiches Öffnen der Rezeptur

FALSE

Fehler

1.1.39.27 uaArchiveReadTagValues (RT Professional)

Beschreibung

Liest die aktuellen Werte aus den Feldvariablen.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveReadTagValues (  
    UAHARCHIVE hArchive,  
    LONG* pnFields,  
    LONG cFields,  
    LONG lOptions )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG* pnFields

Reserviert für zukünftige Anwendungen (NULL)

LONG cFields

Anzahl der übergebenen Feldindizes (Größe des Arrays pnFields).

Reserviert für zukünftige Anwendungen (0)

LONG lOptions

Reserviert für zukünftige Anwendungen (0)

Für alle anderen Werte von lOptions werden die Daten an der Position des Zeigers eingefügt.

Rückgabewert

TRUE

Erfolgreiches Lesen in der Rezeptur

FALSE

Fehler

1.1.39.28 uaArchiveReadTagValuesByName (RT Professional)

Beschreibung

Liest die Variablenwerte in den aktuellen Daten

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveReadTagValuesByName (  
    UAHARCHIVE hArchive,  
    LPCSTR pszFields,  
    LONG lOptions )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPCSTR pszFields

Reserviert für zukünftige Anwendungen (NULL)

LONG lOptions

Reserviert für zukünftige Anwendungen (0)

Rückgabewert

TRUE

Erfolgreiches Lesen in der Rezeptur

FALSE

Fehler

1.1.39.29 uaArchiveRequery (RT Professional)

Beschreibung

Nach dem Aufruf von uaArchiveSetFilter und uaArchiveSetSort muss die Rezeptur mit uaArchiveRequery neu geladen werden.

Hinweis

Rezepturen sortieren und filtern

Die Systemfunktionen "uaArchiveSetSort" und "uaArchiveSetFilter" können Sie auf eine Rezeptur anwenden, ohne die Rezeptur mit "uaArchiveOpen" zu öffnen. In diesem Fall müssen Sie die Systemfunktion "uaArchiveRequery" nicht aufrufen.

Rufen Sie uaArchiveRequery auch in folgenden Fällen auf:

- Wenn Sie Eingaben über die Rezepturanzeige gemacht haben.
- Wenn Sie Eingaben im Editor "Rezepturen" gemacht haben, die in die Tabellenzelle übernommen werden sollen.

Syntax

```
BOOL uaArchiveRequery(  
UAHARCHIVE hArchive )
```

Nur in C-Scripting verwendbar.

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Erfolgreiches Requery

FALSE

Fehler

1.1.39.30 uaArchiveSetFieldValueDate (RT Professional)

Beschreibung

Schreibt Datum und Uhrzeit in ein Feld im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveSetFieldValueDate (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LPSYSTEMTIME pstDateTime )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei das erste projektierte Feld mit lField = 1 adressiert wird. Mit lField = 0 wird das ID-Feld adressiert.

LPSYSTEMTIME pstDateTime

Datum und Uhrzeit

Rückgabewert

TRUE

Erfolgreiches Schreiben von Datum und Uhrzeit

FALSE

Fehler

1.1.39.31 uaArchiveSetFieldValueDouble (RT Professional)

Beschreibung

Schreibt einen Double Wert in ein Feld im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveSetFieldValueDouble (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    double dValue )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das ID-Feld adressiert.

double dValue

Feldwert

Rückgabewert

TRUE

Erfolgreiches Schreiben von Feldwert

FALSE

Fehler

1.1.39.32 uaArchiveSetFieldValueFloat (RT Professional)

Beschreibung:

Schreibt einen Float Wert in ein Feld im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveSetFieldValueFloat (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    float fValue )
```

Parameter:

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das ID-Feld adressiert.

float fValue

Feldwert

Rückgabewert:

TRUE

Erfolgreiches Schreiben von Feldwert

FALSE

Fehler

1.1.39.33 uaArchiveSetFieldValueLong (RT Professional)

Beschreibung

Schreibt einen Long Integer Wert in ein Feld im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveSetFieldValueLong (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LONG dValue )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das ID-Feld adressiert.

LONG dValue

Feldwert

Rückgabewert

TRUE

Erfolgreiches Schreiben von Feldwert

FALSE

Fehler

1.1.39.34 uaArchiveSetFieldValueString (RT Professional)

Beschreibung

Schreibt einen String in ein Feld im aktuellen Datensatz.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveSetFieldValueString (  
    UAHARCHIVE hArchive,  
    LONG lField,  
    LPCSTR pszString )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das ID-Feld adressiert.

LPCSTR pszString

Feldwert

Rückgabewert

TRUE

Erfolgreiches Schreiben von Feldwert

FALSE

Fehler

1.1.39.35 uaArchiveSetFilter (RT Professional)

Beschreibung

Setzt den Filter. Sie können die Systemfunktion aufrufen, ohne dass Sie die Rezeptur mit "uaArchiveOpen" geöffnet haben.

Hinweis

Wenn Sie die Rezeptur mit "uaArchiveOpen" geöffnet haben, laden Sie die Rezeptur nach dem Filtern mit "uaArchiveRequery" neu.

Nur in C-Scripting verwendbar.

Syntax

```
VOID uaArchiveSetFilter (  
    UAHARCHIVE hArchive,  
    LPSTR pszFilter )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPSTR pszFilter

Zu setzenden Filter

1.1.39.36 uaArchiveSetSort (RT Professional)

Beschreibung

Setzt Sortierung der Rezeptur. Sie können die Systemfunktion aufrufen, ohne dass Sie die Rezeptur mit "uaArchiveOpen" geöffnet haben.

Hinweis

Wenn Sie die Rezeptur mit "uaArchiveOpen" geöffnet haben, laden Sie die Rezeptur nach dem Sortieren mit "uaArchiveRequery" neu.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveSetSort (  
    UAHARCHIVE hArchive,  
    LPSTR pszSort )
```

Die Systemfunktion kann nur in C-Scripting verwendet werden.

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPCSTR pszSort

Sortierung

Rückgabewert

TRUE

Erfolgreiches Setzen der Sortierung

FALSE

Fehler

1.1.39.37 uaArchiveUpdate (RT Professional)

Beschreibung

Aktualisiert die geöffnete Rezeptur. Es werden alle Datenänderungen einer Rezeptur in die Datenbank übernommen. Die Projektierung der Rezeptur bleibt unverändert.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveUpdate (  
    UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Erfolgreiches Aktualisieren der Rezeptur

FALSE

Fehler "Update_failed" = 106

Dieser Fehler tritt bei einer Konsistenz-Verletzung auf. Beispiel: In ein Feld ist das Flag "Feld muss einen Wert besitzen" gesetzt, aber im Feld ist kein Wert abgelegt.

1.1.39.38 uaArchiveWriteTagValues (RT Professional)

Beschreibung

Schreibt die Werte des aktuellen Datensatzes in die Variablen.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveWriteTagValues (  
    UAHARCHIVE hArchive,  
    LONG* pnFields,  
    LONG cFields,  
    LONG lOptions )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LONG* pnFields

Reserviert für zukünftige Anwendungen (NULL)

LONG cFields

Reserviert für zukünftige Anwendungen (0)

LONG lOptions

Reserviert für zukünftige Anwendungen (0)

Returnwert

TRUE

Erfolgreiches Lesen in der Rezeptur

FALSE

Fehler

1.1.39.39 uaArchiveWriteTagValuesByName (RT Professional)

Beschreibung

Schreibt die Werte des aktuellen Datensatzes in die Variablen. Der Zugriff erfolgt aufgrund des Rezeptur- und Feldnamens.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaArchiveWriteTagValuesByName (  
    UAHARCHIVE hArchive,  
    LPCSTR pszFields,  
    LONG lOptions )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

LPCSTR pszFields

Reserviert für zukünftige Anwendungen (NULL)

LONG lOptions

Reserviert für zukünftige Anwendungen (0)

Rückgabewert

TRUE

Erfolgreiches Lesen in der Rezeptur

FALSE

Fehler

1.1.39.40 uaConnect (RT Professional)

Beschreibung

Verbindung zu Rezepturen aufbauen (Runtime).

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaConnect (  
    UAHCONNECT* phConnect )
```

Parameter

UAHCONNECT* phConnect

Zeiger auf Handle zur neu angeschlossenen Rezeptur.

Rückgabewert

TRUE

Erfolgreicher Anschluss einer Rezeptur

FALSE

Fehler

1.1.39.41 uaDisconnect (RT Professional)

Beschreibung

Falls eine Verbindung zu Rezepturen (Runtime) besteht, wird diese aufgelöst.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaDisconnect (  
    UAHCONNECT hConnect )
```

Parameter

UAHCONNECT hConnect

Handle zur angeschlossenen Rezeptur (Runtime). Der Handle wird mit uaConnect eingerichtet.

Rückgabewert

TRUE

Erfolgreiches Abkoppeln einer Rezeptur

FALSE

Fehler

1.1.39.42 uaGetArchive (RT Professional)

Beschreibung

Liest die Rezeptur-Konfiguration.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaGetArchive (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGARCHIVE* pArchive )
```

Parameter

UAHCONFIG hConfig,

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,

Archive-Index (0 bis (uaGetNumArchives()-1))

UACONFIGARCHIVE* pArchive

Zeiger auf Puffer zum Empfangen der Rezeptur-Konfiguration

Rückgabewert

TRUE

Erfolgreicher Zugriff auf die Rezeptur

FALSE

Fehler

1.1.39.43 uaGetField (RT Professional)

Beschreibung:

Liest die Feld-Konfiguration.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaGetField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    long lField,  
    UACONFIGFIELD* pField )
```

Parameter:

UAHCONFIG hConfig,

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,

Archive-Index (0 bis (uaGetNumArchives()-1))

long lField,

Die Feldnummer, wobei lField = 0 das erste Feld adressiert.

UACONFIGFIELD* pArchive

Zeiger auf Puffer zum Empfangen der Feld-Konfiguration

Returnwert:

TRUE

Erfolgreicher Zugriff auf die Rezeptur

FALSE

Fehler

1.1.39.44 uaGetLastError (RT Professional)

Beschreibung

Die Systemfunktionen der WinCC Skriptsprache liefern einen BOOL-Wert zurück, wobei TRUE einer fehlerfreien Bearbeitung entspricht. Falls FALSE returniert wird, kann mit "uaGetLastError()" und "uaGetLastHResult()" der Fehler der letzten Systemfunktion gelesen werden.

Nr in C-Scripting verwendbar.

Falls uaGetLastError() erst aufgerufen wird, nachdem schon mehrere Systemfunktionen abgearbeitet wurden, gibt uaGetLastError() den zuletzt aufgetretenen Fehler zurück. Um Gewissheit zu bekommen, bei welcher Systemfunktion der Fehler auftrat, sollte bei jedem returnieren von FALSE einer Systemfunktion "uaGetLastError()" und "uaGetLastHResult()" aufgerufen werden.

Beispiel:

```

if ( uaArchiveGetFieldValueLong ( hArchive, Index, &IntValue ) ==
TRUE )
printf( "Field Value = %u\n", IntValue );
else
printf("Error calling uaArchiveGetFieldValueLong: %d / %08lx\n",
uaGetLastError(), uaGetLastHResult());

```

Bei Systemfunktionen, die keinen Rückgabewert liefern (VOID) soll in jedem Fall eine Abfrage mittels `uaGetLastError()` erfolgen.

Beispiel:

```

uaArchiveGetFilter(hArchive, pszFilter, cMaxLen);
    INT nUAError = uaGetLastError ( );
    if ( UA_ERROR_SUCCESS != nUAError)
    {
        printf( "Filter = [%s]\n", pszFilter );
    }
    else
    {
        printf("Error calling uaArchiveGetFilter: %d, hr=0x%08lx\n",
nUAError, uaGetLastHResult());
    }

INT uaGetLastError()

```

Returnwert

Fehlerstatus der zuletzt ausgeführten Systemfunktion. Folgende Fehler können von `uaGetLastError()` zurückgegeben werden:

```

UA_ERROR_SUCCESS
UA_ERROR_GENERIC
UA_ERROR_CONNECT_FAILED
UA_ERROR_OPEN_FAILED
UA_ERROR_CLOSE_FAILED
UA_ERROR_REQUERY_FAILED
UA_ERROR_MOVE_FAILED
UA_ERROR_INSERT_FAILED
UA_ERROR_UPDATE_FAILED
UA_ERROR_DELETE_FAILED

```

UA_ERROR_IMPORT_FAILED
UA_ERROR_EXPORT_FAILED
UA_ERROR_READ_FAILED
UA_ERROR_WRITE_FAILED
UA_ERROR_GET_FAILED
UA_ERROR_SET_FAILED
UA_ERROR_INVALID_NAME
UA_ERROR_INVALID_TYPE
UA_ERROR_INVALID_NUMRECS
UA_ERROR_INVALID_COMMTYPE
UA_ERROR_INVALID_LENGTH
UA_ERROR_INVALID_PRECISION
UA_ERROR_NULL_POINTER
UA_ERROR_INVALID_POINTER
UA_ERROR_INVALID_HANDLE
UA_ERROR_INVALID_INDEX
UA_ERROR_SERVER_UNKNOWN

Diese Fehler-Konstanten sowie die Predefines der User Archives Routinen befinden sich in CCUACAPI.H.

1.1.39.45 uaGetLastHResult (RT Professional)

Beschreibung

Liest den zuletzt aufgetretenen COM Fehler. Diese Systemfunktion dient vor allem der Diagnose von Inkompabilitäten bei der verwendeten COM-Implementation beziehungsweise zum Auffinden von Registrierungsfehlern und Kommunikationsstörungen.

Diese Systemfunktion ist grundsätzlich zusätzlich zum UAGetLastError zu verwenden, wenn eine User Archive Systemfunktion (z.B uaConnect) einen Fehler mit "FALSE" signalisiert.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaGetLastHResult ()
```

Rückgabewert

Zuletzt aufgetretener COM Fehler

1.1.39.46 uaGetNumArchives (RT Professional)

Beschreibung

Liest die Anzahl der momentan konfigurierten Rezepturen.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaGetNumArchives (  
    UAHCONFIG hConfig )
```

Parameter

UAHCONFIG hConfig

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

Rückgabewert

Anzahl momentan konfigurierter Rezepturen. Im Fehlerfall wird -1 zurückgegeben.

1.1.39.47 uaGetNumFields (RT Professional)

Beschreibung

Liefert die Anzahl der projizierten Felder. Die Felder "ID", "Letzter Benutzer" und "Letzter Zugriff" werden nicht mitgezählt. In den Konfigurationsaufrufen werden die Indizes mit 0 bis uaGetNumFields() -1 angegeben.

Nur in C-Scripting verwendbar.

Syntax

```
LONG uaGetNumFields (  
    UAHCONFIG hConfig,  
    long lArchive )
```

Parameter

UAHCONFIG hConfig,

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,
Archive-Index (0 bis (uaGetNumArchives()-1))

Rückgabewert

Anzahl der projizierten Felder. Im Fehlerfall wird -1 zurückgegeben.

1.1.39.48 uaQueryArchive (RT Professional)

Beschreibung

Verbindung zur Rezeptur für den Runtimebetrieb herstellen. UaQueryArchive erstellt den Handle UAHARCHIVE.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaQueryArchive (  
    UAHCONNECT hConnect,  
    LONG lArchive,  
    UAHARCHIVE* phArchive )
```

Parameter

UAHCONNECT hConnect

Handle der angeschlossenen Rezeptur (Runtime). Der Handle wird mit uaConnect eingerichtet.

LONG lArchive

ID des Archivs, das verbunden werden soll (0... uaGetNumArchives() -1)

UAHARCHIVE* phArchive

Zeiger auf Handle der Rezeptur

Rückgabewert

TRUE

Erfolgreiche Erstellung des Handles zur Rezeptur.

FALSE

Fehler

Bemerkung

Wenn Sie User Archives-Funktionen in einem Client-Projekt verwenden, das auf redundante Serverpaare schaut, kann beim Wechsel des Masters die User Archives-Verbindung nicht automatisch auf den neuen Master umgeschaltet werden. In diesem Fall liefern alle User Archives-Aufrufe den LastError UA_ERROR_SERVER_UNKNOWN = 1004, worauf die Anwenderprogramme einen neuen uaQueryArchive() bzw. uaQueryArchiveByName() und uaArchiveOpen() durchführen müssen.

1.1.39.49 uaQueryArchiveByName (RT Professional)

Beschreibung

Verbindung zu einer Rezeptur über den Rezepturnamen für den Runtimebetrieb herstellen. UaQueryArchiveByName erstellt den Handle UAHARCHIVE zur Rezeptur.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaQueryArchiveByName (  
    UAHCONNECT hConnect,  
    LPCSTR pszName,  
    UAHARCHIVE* phArchive )
```

Parameter

UAHCONNECT hConnect

Handle der angeschlossenen Rezeptur (Runtime). Der Handle wird mit uaConnect eingerichtet.

LPCSTR pszName

Name der Rezeptur. Bei einem Client-Projekt können Sie hier noch ein Serverpräfix mit '::' als Trennzeichen dem Rezepturnamen voranstellen, wenn ein anderer als der Default-Server verwendet wird.

UAHARCHIVE* phArchive

Zeiger auf Handle der Rezeptur

Rückgabewert

TRUE

Erfolgreiche Erstellung des Handles zur Rezeptur

FALSE

Fehler

Bemerkung

Wenn Sie User Archives-Funktionen in einem Client-Projekt verwenden, das auf redundante Serverpaare schaut, kann beim Wechsel des Masters die User Archives-Verbindung nicht automatisch auf den neuen Master umgeschaltet werden. In diesem Fall liefern alle User Archives-Aufrufe den LastError UA_ERROR_SERVER_UNKNOWN = 1004, worauf die Anwenderprogramme einen neuen uaQueryArchive() bzw. uaQueryArchiveByName() und uaArchiveOpen() durchführen müssen.

1.1.39.50 uaQueryConfiguration (RT Professional)

Beschreibung

Erstelle Verbindung zur Rezeptur für die Konfiguration.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaQueryConfiguration (  
    UAHCONFIG* phConfig )
```

Parameter

UAHCONFIG* phConfig,
Zeiger auf Handle der Rezeptur.

Rückgabewert

TRUE
Erfolgreicher Zugriff auf Rezeptur
FALSE
Fehler

Siehe auch

uaAddArchive (Seite 130)

1.1.39.51 uaReleaseArchive (RT Professional)

Beschreibung

Löst Verbindung zur aktuellen Rezeptur.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaReleaseArchive (  
    UAHARCHIVE hArchive )
```

Parameter

UAHARCHIVE hArchive

Handle der Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Erfolgreiche Verbindungsauflösung zur Rezeptur.

FALSE

Fehler

Bemerkung

Der Handle "hArchive" ist bei erfolgreicher Abkopplung auf "NULL" zu setzen, damit bei einer weiteren Verwendung des nicht mehr gültigen Handles der Fehler "UA_ERROR_INVALID_HANDLE" mit Sicherheit sofort kommt, ohne dass die jeweilige Funktion für längere Zeit in den COM-Schnittstellen verhungert.

1.1.39.52 uaReleaseConfiguration (RT Professional)

Beschreibung

Löse Verbindung zu Rezepturen (Konfiguration).
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaReleaseConfiguration (  
    UAHCONFIG hConfig,
```

```
BOOL bSave )
```

Parameter

UAHCONFIG hConfig

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

BOOL bSave

Sichere Konfigurationsänderungen vor Lösen der Verbindung zu Rezepturen für die Konfiguration.

TRUE = Änderungen sichern, FALSE = Änderungen verwerfen

Achtung: Änderungen sichern (bSave = TRUE) darf nur stattfinden, wenn Runtime nicht aktiv ist !Ob Runtime aktiv ist kann durch Aufruf von ualsActive() geprüft werden.

Rückgabewert

TRUE

Erfolgreiche Verbindungsauflösung

FALSE

Fehler

1.1.39.53 uaRemoveAllArchives (RT Professional)

Beschreibung

Löscht alle Rezepturen, die nicht in Sichten verwendet werden.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaRemoveAllArchives  
( UAHCONFIG hConfig )
```

Parameter

UAHCONFIG hConfig

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

Rückgabewert**TRUE**

Erfolgreiches Löschen

FALSE

Fehler

Bemerkung

Im Anschluss können Sie mit "uaGetNumArchives()" abfragen, ob alle Rezepturen gelöscht wurden.

1.1.39.54 uaRemoveAllFields (RT Professional)**Beschreibung**

Entfernt alle Felder.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaRemoveAllFields (  
    UAHCONFIG hConfig,  
    long lArchive )
```

Parameter**UAHCONFIG hConfig,**

Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,

Archiv-Index (0 bis (uaGetNumArchives()-1))

Rückgabewert**TRUE**

Erfolgreiches Löschen von Feldern

FALSE

Fehler

1.1.39.55 uaRemoveArchive (RT Professional)

Beschreibung

uaRemoveArchive löscht die komplette projektierte Rezeptur.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaRemoveArchive (  
    UAHCONFIG hConfig,  
    long lArchive )
```

Parameter

UAHCONFIG hConfig,
Konfigurations-Handle der Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,
Archive-Index (0 bis (uaGetNumArchives()-1))

Rückgabewert

TRUE
Erfolgreiches Löschen der Rezeptur.

FALSE
Fehler

1.1.39.56 uaRemoveField (RT Professional)

Beschreibung

Entfernt ein Feld.
Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaRemoveField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    long lField )
```

Parameter

UAHCONFIG hConfig,

Konfigurations-Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,

Archive-Index (0 bis (uaGetNumArchives()-1))

long lField,

Die Feldnummer, wobei lField = 0 das erste Feld adressiert.

Rückgabewert

TRUE

Erfolgreiches Löschen des Feldes

FALSE

Fehler

1.1.39.57 uaSetArchive (RT Professional)

Beschreibung:

Setzt Konfiguration einer Rezeptur.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaSetArchive (  
    UAHCONFIG hConfig,  
    long lArchive,  
    UACONFIGARCHIVE* pArchive  
)
```

Parameter

UAHCONFIG hConfig,

Konfigurations-Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,

Archive-Index (0 bis (uaGetNumArchives()-1))

UACONFIGARCHIVE* pArchive

Zeiger auf Puffer pArchive mit der Rezeptur-Konfiguration.

Rückgabewert

TRUE

Erfolgreicher Zugriff auf Rezeptur.

FALSE

Es ist ein Fehler aufgetreten.

1.1.39.58 uaSetField (RT Professional)

Beschreibung

Setzt Feld-Konfiguration.

Nur in C-Scripting verwendbar.

Syntax

```
BOOL uaSetField (  
    UAHCONFIG hConfig,  
    long lArchive,  
    long lField,  
    UACONFIGFIELD* pField )
```

Parameter

UAHCONFIG hConfig,

Konfigurations-Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

long lArchive,

Archive-Index (0 bis (uaGetNumArchives()-1))

long lField,

Die Feldnummer, wobei lField = 0 das erste Feld adressiert.

UACONFIGFIELD* pField

Zeiger auf Puffer der Feld-Konfiguration

Rückgabewert

TRUE

Erfolgreicher Zugriff auf Rezeptur.

FALSE

Fehler

1.2 C-bib (RT Professional)

1.2.1 ctype-Funktionen (RT Professional)

Funktionsübersicht

Folgende ctype-Funktionen sind verfügbar:

- long int isalnum (long int x);
- long int isalpha (long int x);
- long int isdigit (long int x);
- long int isgraph (long int x);
- long int islower (long int x);
- long int isprint (long int x);
- long int ispunct (long int x);
- long int isspace (long int x);
- long int isupper (long int x);
- long int isxdigit (long int x);
- long int tolower (long int x);
- long int toupper (long int x);

Eine Beschreibung der ctype-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.2.2 Funktionsgruppe c_bib (RT Professional)

Einleitung

Die Funktionsgruppe c_bib enthält C-Funktionen aus der C-Bibliothek und ist unterteilt in:

- ctype
- math
- memory
- stdio
- stdlib
- string
- time

stdio selbst ist noch einmal unterteilt in:

- char_io
- directio
- error
- file
- file_pos
- output

Die Beschreibung dieser Funktionen finden Sie in der einschlägigen Fachliteratur.

Besonderheiten von Funktionen in WinCC

Die Funktion localtime hat bei der Datumsausgabe folgendes Verhalten:

- Die Zählung der Monate beginnt mit 0.
- Die Jahre werden ab dem Jahr 1900 gezählt, beginnend mit 0.

Die Funktionen der C-Bibliothek printf(), sprintf(), fprintf() können in WinCC nur 360 Zeichen verarbeiten.

1.2.3 math-Funktionen (RT Professional)

Funktionsübersicht

Folgende math-Funktionen sind verfügbar:

- double acos (double x);
- double asin (double x);
- double atan (double x);

- `double atan2 (double x, double y);`
- `double ceil (double x);`
- `double cos (double x);`
- `double cosh (double x);`
- `double exp (double x);`
- `double fabs (double x);`
- `double floor (double x);`
- `double fmod (double x, double y);`
- `double frexp (double x, long int* y);`
- `double ldexp (double x, long int y);`
- `double log (double x);`
- `double log10 (double x);`
- `double modf (double x, double* y);`
- `double pow (double x, double y);`
- `double sin (double x);`
- `double sinh (double x);`
- `double sqrt (double x);`
- `double tan (double x);`
- `double tanh (double x);`

Eine Beschreibung der math-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.2.4 memory-Funktionen (RT Professional)

Funktionsübersicht

Folgende memory-Funktionen sind verfügbar:

- `long int memcmp (const void* cs, const void* ct, size_t n);`
- `void* memchr (const void* cs, long int c, size_t n);`
- `void* memcpy (void* s, const void* ct, size_t n);`
- `void* memmove (void* s, const void* ct, size_t n);`
- `void* memset (void* s, long int c, size_t n);`

Eine Beschreibung der memory-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.2.5 multibyte-Funktionen (RT Professional)

Funktionsübersicht

Folgende multibyte-Funktionen sind verfügbar:

- `int _ismbcalnum(unsigned int c)`
- `int _ismbcalpha(unsigned int c)`
- `int _ismbcdigit(unsigned int c)`
- `int _ismbcgraph(unsigned int c)`
- `int _ismbclower(unsigned int c)`
- `int _ismbcprint(unsigned int c)`
- `int _ismbcpunct(unsigned int c)`
- `int _ismbcspace(unsigned int c)`
- `int _ismbcupper(unsigned int c)`
- `int _mbncmp(const unsigned char *string1, const unsigned char *string2)`
- `int _mbsncmp(const unsigned char *string1, const unsigned char *string2, size_t count)`
- `int _mbsrchr(const unsigned char *string, unsigned int c)`
- `size_t _mbscspn(const unsigned char *string, const unsigned char *strCharSet)`
- `size_t _mbssp(const unsigned char *string, const unsigned char *strCharSet)`
- `size_t _mbstrlen(const char *string)`
- `size_t _mbslen(const unsigned char *string)`
- `unsigned char *_mbscat(unsigned char *strDestination, const unsigned char *strSource)`
- `unsigned char *_mbschr(const unsigned char *string, unsigned int c)`
- `unsigned char *_mbscpy(unsigned char *strDestination, const unsigned char *strSource)`
- `unsigned char *_mbsdec(const unsigned char *start, const unsigned char *current)`
- `unsigned char *_mbsinc(const unsigned char *current)` `size_t _mbclen(const unsigned char *c);`
- `unsigned char *_mbsncat(unsigned char *strDest, const unsigned char *strSource, size_t count)`
- `unsigned char *_mbsncpy(unsigned char *strDest, const unsigned char *strSource, size_t count)`
- `unsigned char *_mbspbrk(const unsigned char *string, const unsigned char *strCharSet)`
- `unsigned char *_mbsstr(const unsigned char *string, const unsigned char *strCharSet)`
- `unsigned char *_mbstok(unsigned char *strToken, const unsigned char *strDelimiter)`

Eine Beschreibung der multibyte-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.2.6 stdio-Funktionen (RT Professional)

Funktionsübersicht

Folgende stdio-Funktionen sind verfügbar:

- `char* fgets (char* s, long int n, FILE* stream);`
- `char* tmpnam (char* s);`
- `FILE* fopen (const char* name, const char* mode);`
- `FILE* freopen (const char* filename, const char* mode, FILE* stream);`
- `FILE* tmpfile ();`
- `fprintf();`
- `long int fclose (FILE* stream);`
- `long int feof (FILE* stream);`
- `long int ferror (FILE* stream);`
- `long int fflush (FILE* stream);`
- `long int fgetc (FILE* stream);`
- `long int fgetpos (FILE* stream, fpos_t* ptr);`
- `long int fputc (long int c, FILE* stream);`
- `long int fputs (const char* s, FILE* stream);`
- `long int fseek (FILE* stream, long int offset, long int origin);`
- `long int fsetpos (FILE* stream, const fpos_t* ptr);`
- `long int ftell (FILE* stream);`
- `long int getc (FILE* stream);`
- `long int putc (long int c, FILE* stream);`
- `long int remove (const char* filename);`
- `long int rename (const char* oldname, const char* newname);`
- `long int setvbuf (FILE* stream, char* buf, long int mode, size_t size);`
- `long int ungetc (long int c, FILE* stream);`
- `long int vfprintf (FILE* stream, const char* format, va_list arg);`
- `long int vsprintf (char* s, const char* format, va_list arg);`
- `printf();`

- `size_t fread (void* ptr, size_t size, size_t nobj, FILE* stream);`
- `size_t fwrite (void* ptr, size_t size, size_t nobj, FILE* stream);`
- `void clearerr (FILE* stream);`
- `void rewind (FILE* stream);`
- `void setbuf (FILE* stream, char* buf);`

Eine Beschreibung der stdio-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.2.7 stdlib-Funktionen (RT Professional)

Funktionsübersicht

Folgende stdlib-Funktionen sind verfügbar:

- `char* getenv (const char* name);`
- `div_t div (long int num, long int denom);`
- `double atof (const char* s);`
- `double strtod (const char* s, char** endp);`
- `ldiv_t ldiv (long int num, long int denom);`
- `long int abs (long int n);`
- `long int atoi (const char* s);`
- `long int atol (const char* s);`
- `long int labs (long int n);`
- `long int rand ();`
- `long int srand (unsigned long int seed);`
- `long int strtol (const char* s, char** endp, long int base);`
- `long int system (const char* s);`
- `unsigned long int strtoul (const char* s, char** endp, long int base);`
- `void abort ();`
- `void* bsearch (const void* key, const void* base, size_t n, size_t size, long int(* cmp) (const`
- `void* calloc (size_t nobj, size_t size);`
- `void exit (long int status);`
- `void free (void* p);`
- `void* keyval, const void* datum));`
- `void* malloc (size_t size);`

- `void qsort (void* base, size_t n, size_t size, long int* cmp, const void* , const void*);`
- `void* realloc (void* p, size_t size);`

Eine Beschreibung der stdlib-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.2.8 string-Funktionen (RT Professional)

Funktionsübersicht

Folgende string-Funktionen sind verfügbar:

- `char* strcat (char* s, const char* ct);`
- `char* strchr (const char* cs, long int c);`
- `char* strcpy (char* s, const char* ct);`
- `char* strerror (size_t n);`
- `char* strncat (char* s, const char* ct, size_t n);`
- `char* strncpy (char* s, const char* ct, size_t n);`
- `char* strpbrk (const char* cs, const char* ct);`
- `char* strrchr (const char* cs, long int c);`
- `char* strstr (const char* cs, const char* ct);`
- `char* strtok (char* s, const char* ct);`
- `long int strcmp (const char* cs, const char* ct);`
- `long int strncmp (const char* cs, const char* ct, size_t n);`
- `size_t strcspn (const char* cs, const char* ct);`
- `size_t strlen (const char* cs);`
- `size_t strspn (const char* cs, const char* ct);`

Eine Beschreibung der string-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.2.9 time-Funktionen (RT Professional)

Funktionsübersicht

Folgende time-Funktionen sind verfügbar:

- char* asctime (const struct tm* tp);
- char* ctime (const time_t* tp);
- clock_t clock ();
- double difftime (time_t time2, time_t time1);
- size_t strftime (char* s, size_t smax, const char* fmt, const struct tm* tp);
- struct tm* gmtime (const time_t* tp);
- struct tm* localtime (const time_t* tp);
- time_t mktime (struct tm* tp);
- time_t time (time_t* tp);

Eine Beschreibung der time-Funktionen finden Sie in der Fachliteratur zur Programmiersprache C.

Die Funktionen können nur in C-Skripten verwendet werden.

1.3 Strukturdefinition (RT Professional)

1.3.1 Strukturdefinition CCAPErrrorExecute (RT Professional)

```
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  
} CCAPErrrorExecute;
```

dwErrorCode1	dwErrorCode2	bCycle	szApplicationName	szFunctionName	szTagName	lpParamSize	dwParamSize	dwCycle	pError	Beschreibung
1007001	0	x	x	x		x	x			Exception in der Aktion
1007001	1	x	x	x		x	x			Exception beim Zugriff auf das Rückgabe-ergebnis
1007001	4097	x	x	x		x	x			Stapelüberlauf beim Ausführen der Aktion
1007001	4098	x	x	x		x	x			Division durch 0 in der Aktion
1007001	4099	x	x	x		x	x			Zugriff auf ein nicht vorhandenes Symbol in der Aktion
1007001	4100	x	x	x		x	x			Access violation in der Aktion
1007004	0	x	x	x						Funktion ist nicht bekannt
1007005	1	x	x							Aktion enthält keinen P-Code
1007005	2	x	x							Funktionsname ist fehlerhaft
1007005	4	x	x	x		x	x			Typ des Rückgabewerts ist ungültig
1007005	32768ff	x	x	x		x	x			Fehler im Ciss Compiler beim Laden der Aktion
1007006	0	x	x	x	x	x	x	x		Variable ist nicht definiert
1007006	1	x	x	x	x	x	x	x		Variable timeout
1007006	2	x	x	x	x	x	x	x	x	Variable kann im gewünschten Format nicht geliefert werden
1007006	3	x	x	x	x	x	x	x	x	Variable liefert Statusverletzung, in CMN_ERROR.dwError1 steht der Status
1007007	1	x	x	x		x	x		x	Fehler bei PDLRTGet-Prop
1007007	2	x	x	x		x	x		x	Fehler bei PDLRTSet-Prop
1007007	3	x	x	x		x	x		x	Fehler bei DM-Aufruf

Fehlerstruktur

Die Funktion OnErrorExecute nutzt die Fehlerstruktur zur Auswertung bzw. zur Ausgabe von Fehlermeldungen, wenn dies in der Spalte pError mit einem "x" gekennzeichnet ist.

1.3.2 Strukturdefinition CCAPTime (RT Professional)

```
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

Das Strukturelement dwCode gibt Auskunft über den Aufruf von OnTime:

dwCode = 113	Aufruf mit Zeitbestimmung für jede Aktion
dwCode = 114	Aufruf mit Zeitüberwachung für eine Aktion

dwFlags

Das Strukturelement dwFlags gibt Auskunft über die Art der Ausgabe:

dwFlags = TRUE	Die Ausgabe der Ergebnisse erfolgt in eine Datei
dwFlags = FALSE	Die Ausgabe der Ergebnisse erfolgt in das Diagnosefenster

1.3.3 Strukturdefinition CMN_ERROR (RT Professional)

```
struct CMNERRORSTRUCT {
    DWORD    dwError1,
    DWORD    dwError2,
    DWORD    dwError3,
    DWORD    dwError4,
    DWORD    dwError5;
    TCHAR    szErrorText[MAX_ERROR_LEN];
}
CMN_ERROR
```

Beschreibung

In der erweiterten Fehlerstruktur befinden sich der Fehlercode und ein Fehlertext für den aufgetretenen Fehler. Jede Applikation kann die Fehlerstruktur zur Auswertung bzw. zur Ausgabe von Fehlermeldungen nutzen.

Members

dwError1 .. dwError5

Diese Einträge können durch die API-Funktionen beliebig verwendet werden.

In den API-Beschreibungen ist jeweils beschrieben, welche Werte die Einträge im Fehlerfall jeweils enthalten. Soweit nichts anderes angegeben ist, stehen die Fehlercodes in dwError1.

szErrorText

Puffer für textuelle Beschreibung der Fehlerursache

Der Inhalt wird aus den Ressourcen ermittelt und ist deshalb sprachabhängig.

1.3.4 Strukturdefinition DM_TYPEREF (RT Professional)

```
typedef struct {
    DWORD dwType;
    DWORD dwSize;
    char szTypeName[MAX_DM_TYPE_NAME + 1];
}
DM_TYPEREF;
```

Members**dwType**

Spezifiziert den Typ der Variablen

dwType	PLC Datentyp	HMI Datentyp
DM_VARTYPE_BIT	Binäre Variable	Bool
DM_VARTYPE_SBYTE	Vorzeichenbehafteter 8-Bit Wert	Byte
DM_VARTYPE_BYTE	Vorzeichenloser 8-Bit Wert	UByte
DM_VARTYPE_SWORD	Vorzeichenbehafteter 16-Bit Wert	Short
DM_VARTYPE_WORD	Vorzeichenloser 16-Bit Wert	UShort
DM_VARTYPE_SDWORD	Vorzeichenbehafteter 32-Bit Wert	Integer
DM_VARTYPE_DWORD	Vorzeichenloser 32-Bit Wert	UInteger
DM_VARTYPE_FLOAT	Gleitkommazahl 32-Bit IEEE 754	Float
DM_VARTYPE_DOUBLE	Gleitkommazahl 64-Bit IEEE 754	Double
DM_VARTYPE_TEXT_8	Textvariable 8-Bit Zeichensatz	Char
DM_VARTYPE_TEXT_16	Textvariable 16-Bit Zeichensatz	String
DM_VARTYPE_RAW	Rohdatentyp	Raw
DM_VARTYPE_STRUCT	Strukturvariable	Struct
DM_VARTYPE_TEXTREF	Text-Referenz-Variable	String

dwSize

Gibt die Länge des Datentyps in Byte an.

szTypeName

Enthält bei Strukturvariablen den Namen des Strukturtyps

1.3.5 Strukturdefinition DM_VAR_UPDATE_STRUCT (RT Professional)

```
typedef struct {
  DM_TYPEREF dmTypeRef;
  DM_VARKEY dmVarKey;
  VARIANT dmValue;
  DWORD dwState;
}
DM_VAR_UPDATE_STRUCT;
```

Members**dmTypeRef**

Enthält Angaben zum Datentyp. Bei zyklischen Anforderungen wird in diese Struktur aus Performancegründen nichts eingetragen.

dmVarKey

Spezifiziert die zu bearbeitenden Variablen.

dmValue

Wert der Variablen.

Bei Zugriffen auf den Wert des VARIANTS muss zwischen dem Namen des VARIANTS und dem Namen des Members ein ".u." eingefügt werden.

Beispiel

```
// Variant versorgen
myVariant.vt = VT_I4;
myVariant.u.lVal = 233;
```

Die Beschreibung des Datentyps VARIANT finden Sie in der einschlägigen Fachliteratur. Der VARIANT dmValue muss vor Erstbenutzung mit VariantInit() initialisiert und nach Benutzung mit VariantClear(&dmValue) wieder freigegeben werden. Die Struktur DM_VAR_UPDATE_STRUCT darf aus diesem Grund nicht mit ZeroMemory() oder memset() gelöscht werden.

dwState

Kennzeichnet den Status der Variablen.

1.3.6 Strukturdefinition DM_VAR_UPDATE_STRUCTEX (RT Professional)

```
typedef struct {
    DM_TYPEREF dmTypeRef;
    DM_VARKEY dmVarKey;
    VARIANT dmValue;
    DWORD dwState;
    DWORD dwQualityCode;
}
DM_VAR_UPDATE_STRUCTEX;
```

Members

dmTypeRef

Enthält Angaben zum Datentyp. Bei zyklischen Anforderungen wird in diese Struktur aus Performancegründen nichts eingetragen.

dmVarKey

Spezifiziert die zu bearbeitenden Variablen.

dmValue

Wert der Variablen.

Bei Zugriffen auf den Wert des VARIANTS muss zwischen dem Namen des VARIANTS und dem Namen des Members ein ".u." eingefügt werden.

Beispiel

```
// Variant versorgen
myVariant.vt = VT_I4;
myVariant.u.lVal = 233;
```

Die Beschreibung des Datentyps VARIANT finden Sie in der einschlägigen Fachliteratur. Der VARIANT dmValue muss vor Erstbenutzung mit VariantInit() initialisiert und nach Benutzung mit VariantClear(&dmValue) wieder freigegeben werden. Die Struktur DM_VAR_UPDATE_STRUCTEX darf aus diesem Grund nicht mit ZeroMemory() oder memset() gelöscht werden.

dwState

Kennzeichnet den Status der Variablen.

dwQualityCode

Kennzeichnet den QualityCode der Variablen.

1.3.7 Strukturdefinition DM_VARKEY (RT Professional)

```
typedef struct {
    DWORD dwKeyType;
    DWORD dwID;
    char szName[ MAX_DM_VAR_NAME + 1 ];
    LPVOID lpvUserData;
}
DM_VARKEY;
```

Members

dwKeyType

Gibt an, ob die Variable über eine Schlüssel-ID oder über ihren Namen angesprochen werden soll:

DM_VARKEY_ID Spezifizierung über Schlüssel ID

DM_VARKEY_NAME Spezifizierung über Variablennamen

dwID

Enthält die Schlüssel-ID der Variablen, wenn dwKeyType entsprechend gesetzt ist

szName

Enthält den Namen der Variablen, wenn dwKeyType entsprechend gesetzt ist.

lpvUserData

Zeiger auf applikationsspezifische Daten

1.3.8 Strukturdefinition LINKINFO (RT Professional)

```
typedef struct {
LINKTYPE LinkType;
DWORD dwCycle;
TCHAR szLinkName[256];
}
LINKINFO;
```

Members

LinkType

LinkType sind Enumerationskonstanten, die in der Datei "Trigger.h" definiert sind. Diese sind mit dem Befehl #include "Trigger.h" und der entsprechenden Enumerationskonstanten in Ihr Script einzubinden.

BUBRT_LT_NOLINK	0	keine Verknüpfung
BUBRT_LT_VARIABLE_DIRECT	1	direkte Variable
BUBRT_LT_VARIABLE_INDIRECT	2	indirekte Variable
BUBRT_LT_ACTION	3	C-Aktion
BUBRT_LT_ACTION_WIZARD	4	Dynamic Dialog
BUB_LT_DIRECT_CONNECTION	5	Direktverbindung
BUBRT_LT_ACTION_WIZARD_INPROC	6	Dynamic Dialog

Bei der Funktion SetLink dürfen nur die Enumerationskonstanten BUBRT_LT_VARIABLE_DIREKT und BUBRT_LT_VARIABLE_INDIRECT verwendet werden. Bei der Funktion GetLink können alle aufgeführten Enumerationskonstanten zurückgeliefert werden.

dwCycle

Zykluszeit der Aktualisierung

255	Bildzyklus
235	Fensterzyklus
0	Bei Änderung
1	250ms

2	500ms
3	1s
4	2s
5	5s
6	10s
7	1min
8	5min
9	10min
10	1h
11-1 5	Anwenderzyklus 1-5

szLinkName

Variablenname

1.3.9 Strukturdefinition MSG_FILTER_STRUCT (RT Professional)

```
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;
```

Beschreibung

In dieser Struktur werden die Kriterien angegeben.

Members

dwFilter

Die Filterbedingungen werden über die folgenden Konstanten aus der Datei "m_global.h" festgelegt:

MSG_FILTER_DATE_FROM	Datum von
MSG_FILTER_DATE_TO	Datum bis
MSG_FILTER_TIME_FROM	Uhrzeit von
MSG_FILTER_TIME_TO	Uhrzeit bis
MSG_FILTER_NR_FROM	Meldungsnummer von
MSG_FILTER_NR_TO	Meldungsnummer bis
MSG_FILTER_CLASS	Meldungsklassen
MSG_FILTER_STATE	Meldungsstatus
MSG_FILTER_AG_FROM	AG Nummer von
MSG_FILTER_AG_TO	AG Nummer bis
MSG_FILTER_AGSUB_FROM	AG Subnummer von
MSG_FILTER_AGSUB_TO	AG Subnummer bis
MSG_FILTER_TEXT	Meldungstexte
MSG_FILTER_PVALUE	Prozesswerte

MSG_FILTER_COUNTER_FROM	Interner Meldungsähler von
MSG_FILTER_COUNTER_TO	Interner Meldungsähler bis
MSG_FILTER_PROCESSMSG	Prozessmeldungen
MSG_FILTER_SYSMMSG	Systemmeldungen
MSG_FILTER_BEDMSG	Bedienmeldungen
MSG_FILTER_DATE	Datum von bis
MSG_FILTER_TIME	Uhrzeit von bis
MSG_FILTER_NR	Meldungsnummer von bis
MSG_FILTER_VISIBLEONLY	Sichtbare Meldungen anzeigen
MSG_FILTER_HIDDENONLY	Ausgeblendete Meldungen anzeigen

st

Datum / Uhrzeit von - bis

Dabei ist st[0] Beginnzeitpunkt (von), st[1] Endzeitpunkt (bis)

Belegen Sie diese Felder für die Kriterien: MSG_FILTER_DATE, MSG_FILTER_DATE_FROM, MSG_FILTER_DATE_TO, MSG_FILTER_TIME, MSG_FILTER_TIME_FROM, bzw. MSG_FILTER_TIME_TO

Wird eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt, so ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

dwMsgNr

Meldungsnummer von - bis

Dabei ist dwMsgNr[0] Anfangsnr. (von), dwMsgNr[1] Endnr. (bis)

Belegen Sie diese Felder für die Kriterien: MSG_FILTER_NR, MSG_FILTER_NR_FROM bzw. MSG_FILTER_NR_TO

dwMsgClass

Meldungsklassen bitcodiert

Belegen Sie dieses Feld für das Kriterium: MSG_FILTER_CLASS

dwMsgType

Meldungsart je Meldeklasse, bitcodiert

Belegen Sie dieses Feld für das Kriterium: MSG_FILTER_CLASS

dwMsgState

Meldungsstatus bitcodiert

Belegen Sie dieses Feld für das Kriterium: MSG_FILTER_STATE

wAGNr

AGNr von - bis

Belegen Sie diese Felder für die Kriterien: MSG_FILTER_AG_FROM bzw. MSG_FILTER_AG_TO

wAGSubNr

AGSubNr von - bis

Belegen Sie dieses Feld für die Kriterien: MSG_FILTER_AGSUB_FROM bzw. MSG_FILTER_AGSUB_TO

dwArchivMode

Archivierung/ Protokollierung

Muss mit 0 belegt werden.

szTB

Texte der Textblöcke

Belegen Sie diese Felder für das Kriterium: MSG_FILTER_TEXT

dwTB

Textblöcke aktiv (von - bis, bitcodiert)

Belegen Sie dieses Feld für das Kriterium: MSG_FILTER_TEXT

dPValue

Prozesswerte von - bis

Belegen Sie diese Felder für das Kriterium: MSG_FILTER_PVALUE

dwPValue

Prozesswerte aktiv (von - bis bitcodiert)

Belegen Sie dieses Feld für das Kriterium: MSG_FILTER_PVALUE

dwMsgCounter

Interner Meldunzähler von - bis

Belegen Sie diese Felder für die Kriterien: MSG_FILTER_COUNTER_FROM, MSG_FILTER_COUNTER_TO

dwQuickSelect

Schnellselektion für Stunde, Tag, Monat

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

Belegen Sie dieses Feld für das Kriterium: MSG_FILTER_QUICKSELECT

LOWORD Typ:

MSG_FILTER_QUICK_MONTH	Schnellselektion letzte n Monate
MSG_FILTER_QUICK_DAYS	Schnellselektion letzte n Tage
MSG_FILTER_QUICK_HOUR	Schnellselektion letzte n Stunden

HIWORD Anzahl: 1...n

Der Endzeitpunkt der Schnellselektion bezieht sich auf die aktuelle Systemzeit des lokalen Rechners. Der Startzeitpunkt wird $n * (\text{Monate, Tage, Stunden zurückgerechnet})$.

1.3.10 Strukturdefinition MSG_RTDATA_STRUCT (RT Professional)

```
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

Zustand einer Meldung

MSG_STATE_COME	0x00000001	Meldung gekommen
MSG_STATE_GO	0x00000002	Meldung gegangen
MSG_STATE_QUIT	0x00000003	Meldung quittiert

MSG_STATE_LOCK	0x00000004	Meldung gesperrt
MSG_STATE_UNLOCK	0x00000005	Meldung entsperrt
MSG_STATE_QUIT_SYSTEM	0x00000010	Meldung quittiert durch System
MSG_STATE_QUIT_EMERGENCY	0x00000011	Not - Quittierung
MSG_STATE_QUIT_HORN	0x00000012	Quittierung zentraler Melder
MSG_STATE_COMEGO	0x00000013	Meldung gekommen und gegangen, nur in der Anzeige "Aktuelle Meldungen"
MSG_STATE_UPDATE	0x00010000	Bit für Meldungsupdate
MSG_STATE_RESET	0x00020000	Bit für Meldungsreset
MSG_STATE_SUMTIME	0x00040000	Bit für Sommerzeit aktiv
MSG_STATE_INSTANCE	0x00080000	Bit für Instanzmeldung (n Meldungen einer Nr.)

dwMsgNr

Meldungsnummer

stMsgTime

Datum / Uhrzeit: Telegrammzeit in Abhängigkeit von der aufrufenden Funktion

dwTimeDiff

Zeitdauer kommend / Telegrammzeit in Sekunden

dwCounter

Interner Meldungsähler

dwFlags

Meldungsflags in der Datenbank

MSG_FLAG_SUMTIME	0x00000001	Sommerzeit aktiv
MSG_FLAG_COMMENT	0x00000002	Meldung hat Kommentar
MSG_FLAG_ARCHIV	0x00000004	Archivieren
MSG_FLAG_PROTOCOL	0x00000008	Protokollieren
MSG_FLAG_TEXTVALUES	0x00000010	Meldung hat Textbegleitwerte
MSG_FLAG_TIMEINVALID	0x00000020	Bit für ungültigen Datum/Uhrzeit-Stempel
MSG_FLAG_INSTANCE	0x00000040	Kennzeichnung von Instanzmeldungen (185269)

wPValueUsed

benutzte Prozesswerte, bitcodiert. Jedes Bit darf nur in einem der beiden Strukturelemente "wPValueUsed" oder "wTextValueUsed" gesetzt werden. Ein Begleitwert kann entweder Zahl oder Text sein.

wTextValueUsed

verwendete Textwerte, bitcodiert. Jedes Bit darf nur in einem der beiden Strukturelemente "wPValueUsed" oder "wTextValueUsed" gesetzt werden. Ein Begleitwert kann entweder Zahl oder Text sein.

Runtime API (RT Professional)

2.1 Runtime API (RT Professional)

Einleitung

Runtime API beschreibt die offen gelegte Programmierschnittstelle von WinCC. Über die API-Funktionen nutzen Sie in eigenen Anwendungen die internen Funktionen von WinCC und greifen auf Daten von HMI-Variablen oder in den Archivdaten zu.

Hinweis

Siemens übernimmt keine Haftung oder Gewährleistung dafür, dass die über die API-Schnittstellen transportierten Daten und Informationen kompatibel zu Software Dritter sind.

Wir weisen ausdrücklich darauf hin, dass es bei unsachgemäßer Nutzung der API-Schnittstelle zu Datenverlust oder Produktionsausfall kommen kann.

Beispiele:

- MSRTCreateMsgPlus(): Meldung erzeugen
- DMGetValue(): Variablenwert ermitteln.

Voraussetzung

- Programmierumgebung ist installiert, z. B. MS Visual Studio
- WinCC Runtime Professional ist installiert.

Einsatz

Die API-Funktionen setzen Sie an folgenden Stellen ein:

- Innerhalb von WinCC: In benutzerdefinierte C-Funktionen und lokalen C-Skripten.
- Außerhalb von WinCC: In Windows-Applikationen, die in der Programmiersprache C/C++ erstellt sind. Um Runtime API in den Programmiersprachen C# oder VB.net zu nutzen, müssen Sie eine entsprechende Umsetzung programmieren.

Lieferumfang

Runtime API besteht aus den folgenden Komponenten:

- Dokumentation der API-Funktionen
- Beispiele in C und C++
- Include-Dateien und Lib-Dateien

Beispiele in C und C++

Beispiele finden Sie im Verzeichnis "%ProgramFiles%\SIEMENS\Automation\SCADA-RT_V11\WINCC\samples"

Include-Dateien und Lib-Dateien

Um lauffähigen Code zu generieren benötigen Sie die Include-Dateien und Lib-Dateien. Sie finden diese Dateien im Verzeichnis:

- %ProgramFiles%\SIEMENS\Automation\SCADA-RT_V11\WinCC\include
- %ProgramFiles%\SIEMENS\Automation\SCADA-RT_V11\WinCC\lib

2.2 Funktionen der Datenhaltung (RT Professional)

2.2.1 Grundlagen (RT Professional)

2.2.1.1 Übersicht über die Strukturen (RT Professional)

Übersicht

DM_CONNECTION_DATA (Seite 214)	Verbindungsdaten
DM_CONNKEY (Seite 215)	Verbindungsdaten auflisten (Filterstruktur)
DM_CYCLE_INFO (Seite 216)	Aktualisierungszyklen
DM_DATA_SERVICE (Seite 217)	Datentransportkanal
DM_DIRECTORY_INFO (Seite 218)	Pfad- und Dateinamen der Projektierungsdaten
DM_DLGOPTIONS (Seite 219)	Dialogvorgaben
DM_MACHINE_TABLE (Seite 221)	Formatkonvertierung
DM_PROJECT_INFO (Seite 222)	Projektinformationen
DM_SD_TARGET_APP (Seite 226)	Spezifikation von Applikationen
DM_SD_TARGET_MACHINE (Seite 225)	Rechnerinformationen
DM_SEND_DATA_STRUCT (Seite 223)	Spezifikation von Partnern beim Datenaustausch
DM_TYPEREF (Seite 227)	Variablentypreferenz
DM_VAR_UPDATE_STRUCTEX (Seite 230)	Variablenwerte ermitteln
DM_VAR_UPDATE_STRUCT (Seite 228)	Variablenwerte ermitteln
DM_VARFILTER (Seite 233)	Selektion von Variablen
DM_VARGRP_DATA (Seite 235)	Informationen zu einer Variablengruppe
DM_VARGRPKEY (Seite 236)	Spezifikation einer Variablengruppe
DM_VARIABLE_DATA4 (Seite 239)	mit Informationen zu einer Variablen
DM_VARIABLE_DATA (Seite 237)	mit Informationen zu einer Variablen
DM_FORMAT_INFO (Seite 220)	Formatkonvertierung
DM_VARKEY (Seite 241)	Spezifikation einer Variablen

DM_VARLIMIT (Seite 244)	Variablengrenzwerte
MCP_NEWVARIABLE_DATA_4 (Seite 247)	Variablendefinition
MCP_NEWVARIABLE_DATA_5 (Seite 249)	Variablendefinition
MCP_NEWVARIABLE_DATA_EX4 (Seite 254)	Variablendefinition
MCP_NEWVARIABLE_DATA_EX (Seite 252)	Variablendefinition
MCP_NEWVARIABLE_DATA (Seite 245)	Variablendefinition
MCP_VARIABLE_COMMON_EX (Seite 258)	Variablenbeschreibung
MCP_VARIABLE_COMMON (Seite 256)	Variablenbeschreibung
MCP_VARIABLE_LIMITS_EX (Seite 264)	Variablengrenzwerte
MCP_VARIABLE_LIMITS5 (Seite 262)	Variablengrenzwerte
MCP_VARIABLE_LIMITS (Seite 260)	Variablengrenzwerte
MCP_VARIABLE_PROTOCOL_EX (Seite 267)	Grenzwertbehandlung einer Variablen
MCP_VARIABLE_PROTOCOL (Seite 266)	Grenzwertbehandlung einer Variablen
MCP_VARIABLE_SCALES (Seite 268)	Variablenskalisierung

2.2.1.2 Übersicht über die Funktionen (RT Professional)

Übersicht

DM_ENUM_CYCLES_PROC (Seite 291)	Aktualisierungszyklen auflisten (Callback)
DM_ENUM_FORMATS_PROC (Seite 288)	Formatkonvertierungen auflisten (Callback)
DM_NOTIFY_PROC (Seite 281)	Benachrichtigungsfunktion
DMActivateRTProject (Seite 269)	Projekt aktivieren
DMChangeDataLocale (Seite 277)	Sprachumschaltung (Applikationen benachrichtigen)
DMConnect (Seite 278)	Verbindung zum Datenmanager aufnehmen
DMDeactivateRTProject (Seite 284)	Runtime-Projekt deaktivieren
DMDisconnect (Seite 285)	Verbindung zum Datenmanager abbauen
DMEnumNumberFormats (Seite 286)	Formatkonvertierungen auflisten
DMEnumUpdateCycles (Seite 289)	Aktualisierungszyklen auflisten
DMExitWinCCEX (Seite 293)	WinCC mit Vorgaben beenden
DMExitWinCC (Seite 292)	WinCC beenden
DMGetConnectionState (Seite 297)	Verbindung zum Datenmanager abfragen
DMGetHotkey (Seite 299)	Hotkey-ID ermitteln
DMGetMachineInfo (Seite 300)	Anlaufparameter ein Applikation abfragen
DMGetMachineTable (Seite 301)	Rechnerliste abfragen

Siehe auch

DMGetDataLocale (Seite 298)

2.2.1.3 Quality Codes von HMI-Variablen (RT Professional)

Einleitung

Der Quality Code wird dazu benötigt, den Status und die Qualität einer Variablen zu prüfen. Im angezeigten Quality Code ist die Qualität der gesamten Wert-Übertragung und Wert-Verarbeitung der jeweiligen HMI-Variable zusammengefasst. So ist am Quality Code u.a. erkennbar, ob der aktuelle Wert ein Start- oder Ersatzwert ist.

Die Quality Codes sind priorisiert. Treten gleichzeitig mehrere Codes auf, so wird der schlechteste Quality Code angezeigt.

Auswertung der Quality Codes

Sie können den Quality Code auf unterschiedliche Weise auswerten:

- Auswertung mit VB-Funktionen
- Auswertung mit C-Funktionen
- Auswertung über das Ereignis "Quality Code geändert" eines E/A-Feldes

Hinweis

Um bei Prozessvariablen die gesamte Wert-Übertragung und Wert-Verarbeitung im Quality Code einzubeziehen, muss das angeschlossene Automatisierungssystem den Quality Code unterstützen. Bei der Projektierung der PLC-Variablen im AS ist der notwendige Speicherbedarf für den Quality Code zu beachten. Der Quality Code benötigt beispielsweise in einem AS der S7-Familie ein zusätzliches Byte, angehängt an den Prozesswert. Um Fehler zu vermeiden, muss dieses Byte bei der Projektierung einer Variablen, z.B. am Ende eines Datenbaustein, berücksichtigt werden.

Aufbau

Der Quality Code hat folgende binäre Struktur:

QQSSSSL

Q: Qualität

S: Sub-Status der Qualität

L: Limits. Dieser Wert ist optional.

Hinweis

Die in der Tabelle "Qualität" dargestellten Quality Code sind Grundwerte der Qualitätsstufen. Durch die Nutzung von Sub-Status und Limit entstehen Zwischenwerte oberhalb der entsprechenden Qualitätsstufe.

Qualität

Die ersten zwei Stellen geben die Qualität der Variablen an.

	Q	Q	S	S	S	S	L	L
	2	2	2	2	2	2	2	2
	7	6	5	4	3	2	1	0
Bad - The value is not useful	0	0	-	-	-	-	-	-
Uncertain - The quality of the value is less than normal, but the value may still be useful.	0	1	-	-	-	-	-	-
Good (Non-Cascade) - The quality of the value is good. Possible alarm conditions may be indicated by the sub-status.	1	0	-	-	-	-	-	-
Good (Cascade) - The value may be used in control.	1	1	-	-	-	-	-	-

Sub-Status

Die Qualität allein reicht nicht aus. Sub Stati unterteilen die einzelnen Qualitäten. Der Quality Code ist binär codiert. Zur Analyse des Quality Code muss der Wert in seine hexadezimale Darstellung umgewandelt werden.

Quality Code von Variablen

Entnehmen Sie die realisierten Quality Code der nachfolgenden Tabelle. Die Tabelle beginnt mit dem schlechtesten Quality Code und endet mit dem besten Quality Code. Dabei besitzt der beste Quality Code die niedrigste Priorität und der schlechteste Quality Code die höchste Priorität. Wenn bei einer Variablen in der Verarbeitungskette mehrere Stati zeitgleich auftreten, dann wird der schlechteste Code weitergegeben.

Code (Hex)	Qualität		Q	Q	S	S	S	S	L	L
0x23	Bad	Device passivated - Diagnostic alerts inhibited	0	0	1	0	0	0	1	1
0x3F	Bad	Function check - Local override	0	0	1	1	1	1	1	1
0x1C	Bad	Out of Service - The value is not reliable because the block is not being evaluated, and may be under construction by a configurer. Set if the block mode is O/S.	0	0	0	1	1	1	-	-
0x73	Uncertain	Simulated value - Start	0	1	1	1	0	0	1	1
0x74	Uncertain	Simulated value - End	0	1	1	1	0	1	-	-
0x84	Good (Non-Cascade)	Active Update event - Set if the value is good and the block has an active Update event.	1	0	0	0	0	1	-	-
0x24	Bad	Maintenance alarm - More diagnostics available.	0	0	1	0	0	1	-	-
0x18	Bad	No Communication, with no usable value - Set if there has never been any communication with this value since it was last "Out of Service".	0	0	0	1	1	0	-	-
0x14	Bad	No Communication, with last usable value - Set if this value had been set by communication, which has now failed.	0	0	0	1	0	1	-	-
0x0C	Bad	Device Failure - Set if the source of the value is affected by a device failure.	0	0	0	0	1	1	-	-
0x10	Bad	Sensor failure	0	0	0	1	0	0	-	-

2.2 Funktionen der Datenhaltung (RT Professional)

Code (Hex)	Qualität		Q	Q	S	S	S	S	L	L
0x08	Bad	Not Connected - Set if this input is required to be connected and is not connected.	0	0	0	0	1	0	-	-
0x04	Bad	Configuration Error - Set if the value is not useful because there is some inconsistency regarding the parameterization or configuration, depending on what a specific manufacturer can detect.	0	0	0	0	0	1	-	-
0x00	Bad	non-specific - There is no specific reason why the value is bad. Used for propagation.	0	0	0	0	0	0	-	-
0x28	Bad	Process related - Substitute value	0	0	1	0	1	0	-	-
0x2B	Bad	Process related - No maintenance	0	0	1	0	1	0	1	1
0x68	Uncertain	Maintenance demanded	0	1	1	0	1	0	-	-
0x60	Uncertain	Simulated value - Set when the process value is written by the operator while the block is in manual mode.	0	1	1	0	0	0	-	-
0x64	Uncertain	Sensor calibration	0	1	1	0	0	1	-	-
0x5C	Uncertain	Configuration error	0	1	0	1	1	1	-	-
0x58	Uncertain	Sub-normal	0	1	0	1	1	0	-	-
0x54	Uncertain	Engineering Unit Range Violation - Set if the value lies outside of the set of values defined for this parameter. The Limits define which direction has been exceeded.	0	1	0	1	0	1	-	-
0x50	Uncertain	Sensor conversion not accurate	0	1	0	1	0	0	-	-
0x4B	Uncertain	Substitute (constant)	0	1	0	0	1	0	1	1
0x78	Uncertain	Process related - No maintenance	0	1	1	1	1	0	-	-
0x4C	Uncertain	Initial Value - Value of volatile parameters during and after reset of the device or of a parameter.	0	1	0	0	1	1	-	-
0x48	Uncertain	Substitute value - Predefined value is used instead of the calculated one. This is used for fail safe handling.	0	1	0	0	1	0	-	-
0x44	Uncertain	Last Usable Value - Whatever was writing this value has stopped doing so. This is used for fail safe handling.	0	1	0	0	0	1	-	-
0x40	Uncertain	Non-specific - There is no specific reason why the value is uncertain. Used for propagation.	0	1	0	0	0	0	-	-
0xE0	Good (Cascade)	Initiate Fail Safe (IFS) - The value is from a block that wants its downstream output block (e.g. AO) to go to Fail Safe.	1	1	1	0	0	0	-	-
0xD8	Good (Cascade)	Local Override (LO) - The value is from a block that has been locked out by a local key switch or is a Complex AO/DO with interlock logic active. The failure of normal control must be propagated to a function running in a host system for alarm and display purposes. This also implies "Not Invited".	1	1	0	1	1	0	-	-
0xD4	Good (Cascade)	Do Not Select (DNS) - The value is from a block which should not be selected, due to conditions in or above the block.	1	1	0	1	0	1	-	-
0xCC	Good (Cascade)	Not Invited (NI) - The value is from a block which does not have a target mode that would use this input.	1	1	0	0	1	1	-	-
0xC8	Good (Cascade)	Initialization Request (IR) - The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or the mode is wrong.	1	1	0	0	1	0	-	-

Code (Hex)	Qualität		Q	Q	S	S	S	S	L	L
0xC4	Good (Cascade)	Initialization Acknowledge (IA) - The value is an initialized value from a source (cascade input, remote-cascade in, and remote-output in parameters).	1	1	0	0	0	1	-	-
0xC0	Good (Cascade)	OK - No error or special condition is associated with this value.	1	1	0	0	0	0	-	-
0xA0	Good (Non-Cascade)	Initiate fail safe	1	0	1	0	0	0	-	-
0x98	Good (Non-Cascade)	Unacknowledged Critical Alarm - Set if the value is good and the block has an unacknowledged Alarm with a priority greater than or equal to 8.	1	0	0	1	1	0	-	-
0x94	Good (Non-Cascade)	Unacknowledged Advisory Alarm - Set if the value is good and the block has an unacknowledged Alarm with a priority less than 8.	1	0	0	1	0	1	-	-
0x90	Good (Non-Cascade)	Unacknowledged Update event - Set if the value is good and the block has an unacknowledged Update event.	1	0	0	1	0	0	-	-
0x8C	Good (Non-Cascade)	Active Critical Alarm - Set if the value is good and the block has an active Alarm with a priority greater than or equal to 8.	1	0	0	0	1	1	-	-
0x88	Good (Non-Cascade)	Active Advisory Alarm - Set if the value is good and the block has an active Alarm with a priority less than 8.	1	0	0	0	1	0	-	-
0xA8	Good (Non-Cascade)	Maintenance demanded	1	0	1	0	1	0	-	-
0xA4	Good (Non-Cascade)	Maintenance required	1	0	1	0	0	1	-	-
0xBC	Good (Non-Cascade)	Function check - Local override	1	0	1	1	1	1	-	-
0x80	Good (Non-Cascade)	OK - No error or special condition is associated with this value.	1	0	0	0	0	0	-	-

Limit

Die Limits können die Quality Codes noch weiter unterteilen. Limits sind optional.

	Q	Q	S	S	S	S	L	L
O.K. - The value is free to move.	-	-	-	-	-	-	0	0
Low limited - The value has acceded its low limits.	-	-	-	-	-	-	0	1
High limited - The value has acceded its high limits.	-	-	-	-	-	-	1	0
Constant (high and low limited) - The value cannot move, no matter what the process does.	-	-	-	-	-	-	1	1

Quality Codes bei der Kommunikation mit OPC

Bei einer Kommunikation über "OPC" werden die von OPC nicht unterstützten Quality Codes umgesetzt.

Quality Code in WinCC	Quality Code nach OPC
0x48	0x40
0x4C	0x40
0x5C	0x40
0x60	0x40
0x80...0xD4	0xC0
0xD8	0xC0

2.2.1.4 Konstanten (RT Professional)

Allgemeine Definitionen

MAX_DM_OHIO_MACHINES	64	maximale Anzahl von PCs in einem Projekt
MAX_DM_OHIO_APPLICATIONS	32	maximale Anzahl lokaler Client-Applikationen
MAX_DM_UPDATE_CYCLES	15	maximale Anzahl Aktualisierungszyklen
MAX_DM_SYSTEM_CYCLES	10	davon 10 Systemzyklen
MAX_DM_USER_CYCLES	5	und 5 Anwenderzyklen
MAX_DM_VAR_NAME	128	maximale Länge eines Variablennamens
MAX_DM_TYPE_NAME	128	maximale Länge eines Datentypnamens
MAX_DM_GROUP_NAME	64	maximale Länge eines Variablengruppennamens
MAX_DM_CYCLE_NAME	64	maximale Länge eines Zykluszeittexts
MAX_DM_FORMAT_NAME	64	maximale Länge einer Formatvorschrift
MAX_DM_SCALE_NAME	64	maximale Länge einer Skalierungsvorschrift
MAX_DM_SCALE_PARAM_NAME	64	maximale Länge einer Skalierungsparameterbeschreibung
MAX_DM_MEMBER_NAME	32	maximale Länge für Mitglieder zusammengesetzter Typen
MAX_DM_INFOTEXT_LEN	255	maximale Länge eines Info Texts
MAX_DM_SHIFT_NAME_LEN	32	maximale Länge eines Schichtnamens
MAX_DM_SHIFTS	16	maximale Anzahl Schichten pro Tag
MAX_DM_SHIFT_HOLYDAYS	30	maximale Anzahl der Feiertage im Schichtplan
MAX_DM_SHIFT_HOLYNAME	64	maximale Länge des Namens eines Feiertages
MAX_DM_SERVICE_NAME	32	maximale Länge eines Servicenamens
MAX_DM_APP_NAME	32	maximale Länge eines log. Applikationsnamens
MAX_DM_DSN_NAME	32	maximale Länge Data Source Name der Datenbank
MAX_DM_UNIT_NAME	65	maximale Länge einer Unit
MAX_DM_CONNECTION_NAME	32	maximale Länge einer Verbindung
MAX_DM_VAR_SPECIFIC	25	maximale Länge des spezifischen Teils einer Variablen im GAPI

MAX_DM_CON_SPECIFIC	128	maximale Länge des spezifischen Teils einer Verbindung im GAPI
MAX_DM_CON_COMMON	128	maximale Länge des spezifischen Teils einer Verbindung im GAPI
MAX_DM_VARTYPE_TEXT_LEN	255	maximale Länge einer Textvariablen

Variablentypen

DM_VARTYPE_BIT	1	Binäre Variable
DM_VARTYPE_SBYTE	2	Vorzeichenbehafteter 8-Bit Wert
DM_VARTYPE_BYTE	3	Vorzeichenloser 8-Bit Wert
DM_VARTYPE_SWORD	4	Vorzeichenbehafteter 16-Bit Wert
DM_VARTYPE_WORD	5	Vorzeichenloser 16-Bit Wert
DM_VARTYPE_SDWORD	6	Vorzeichenbehafteter 32-Bit Wert
DM_VARTYPE_DWORD	7	Vorzeichenloser 32-Bit Wert
DM_VARTYPE_FLOAT	8	Gleitkommazahl 32-Bit IEEE 754
DM_VARTYPE_DOUBLE	9	Gleitkommazahl 64-Bit IEEE 754
DM_VARTYPE_TEXT_8	10	Textvariable 8-Bit Zeichensatz
DM_VARTYPE_TEXT_16	11	Textvariable 16-Bit Zeichensatz
DM_VARTYPE_RAW	12	Rohdatentyp
DM_VARTYPE_STRUCT	14	Strukturvariable
DM_VARTYPE_TEXTREF	18	Text-Referenz aus Textbib

Eigenschaftsflags

DM_SIMULATION_MODE	0x00000001	Variable wird simuliert (keine Schreibzugriffe zum Prozess)
DM_INTERNAL_VAR	0x00000002	interne Variable
DM_EXTERNAL_VAR	0x00000004	externe Variable
DM_HAS_MIN_LIMIT	0x00000008	Variable hat feste Untergrenze
DM_HAS_MAX_LIMIT	0x00000010	Variable hat feste Obergrenze
DM_HAS_DEFAULT_VALUE	0x00000020	Variable hat Ersatzwert
DM_HAS_STARTUP_VALUE	0x00000040	Variable hat Initialisierungswert
DM_USE_DEFAULT_ON_STARTUP	0x00000080	Ersatzwert beim Systemstart eintragen
DM_USE_DEFAULT_ON_MAX	0x00000100	Ersatzwert beim Überschreiten der Obergrenze eintragen
DM_USE_DEFAULT_ON_MIN	0x00000200	Ersatzwert beim Unterschreiten der Untergrenze eintragen
DM_USE_DEFAULT_ON_COMM_ERROR	0x00000400	Ersatzwert bei Verbindungsfehler
DM_WRITE_ACCESS_APPLICATION	0x00000800	Applikationen dürfen die Variable beschreiben
DM_WRITE_ACCESS_PROCESS	0x00001000	Variable darf vom Prozess beschrieben werden

DM_INTERNAL_VAR_LOKAL	0x00002000	Interne Variable, Aktualisierung 1 = Rechnerlokal / 0 = Projektweit
DM_INVISIBLE	0x00004000	Variable im ControlCenter an der Oberfläche nicht anzeigen
DM_EXTERNAL_LOCK	0x00008000	Variable von extern erzeugt und nur von extern löschtbar

Protokollierungs-flags

DM_NOTIFY_MAX_LIMIT	0x00000001	Protokolleintrag bei Erreichen der Obergrenze
DM_NOTIFY_MIN_LIMIT	0x00000002	Protokolleintrag bei Erreichen der Untergrenze
DM_NOTIFY_FORMAT_ERROR	0x00000004	Protokolleintrag bei Wandlungsfehler
DM_NOTIFY_ACCESS_FAULT	0x00000008	Protokolleintrag bei unzulässigen Schreibzugriff
DM_NOTIFY_APPLICATION_WRITE	0x00000010	Protokolleintrag bei Schreibzugriff Applikation
DM_NOTIFY_PROCESS_WRITE	0x00000020	Protokolleintrag bei Schreibzugriff Prozess

Statusflags

Der Variablenstatus einzelner WinCC Variablen kann in Runtime überwacht werden. Der Variablenstatus beinhaltet auch Verletzungen der projektierten Messbereichsgrenzen sowie den Status der Kopplung zwischen WinCC und dem Automatisierungsgerät.

Der Quality Code informiert über die Qualität einer Variablen, unabhängig davon, wo dieser Code gebildet wurde. Dabei wird der Status der gesamten Wertübertragung und Wertverarbeitung berücksichtigt.

Wenn im System z. B. eine Verletzung des Messbereichs an der unteren Grenze auftritt, wird immer Quality Code "0x55" gemeldet. Diese Messbereichsverletzung kann sowohl im Datenmanager von WinCC als auch im Feldgerät aufgetreten sein. Mit dem Variablenstatus können Sie herausfinden, ob diese Messbereichsverletzung in WinCC aufgetreten ist oder bereits vor der Übergabe des Werts an WinCC.

Wenn der Variablenstatus z. B. mit dem Code 0x0010 eine Grenzwertverletzung meldet, dann wurde die in WinCC projektierte Bereichsgrenze unterschritten. Wenn der Variablenstatus keine Grenzwertverletzung meldet, dann enthielt bereits der an WinCC übergebene Quality Code die Grenzwertverletzung.

	0x0000	kein Fehler
DM_VARSTATE_NOT_ESTABLISHED	0x0001	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	0x0002	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	0x0004	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	0x0008	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	0x0010	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	0x0020	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	0x0040	Formatgrenze unterschritten

DM_VARSTATE_CONVERSION_ERROR	0x0080	Anzeige Wandlungsfehler (in Verbindung mit DM_VARSTATE_..._RANGE)
DM_VARSTATE_STARTUP_VALUE	0x0100	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	0x0200	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	0x0400	Adressierungsfehler im Kanal
DM_VARSTATE_INVALID_KEY	0x0800	Variable nicht gefunden / nicht vorhanden
DM_VARSTATE_ACCESS_FAULT	0x1000	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	0x2000	Timeout / keine Rückmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	0x4000	Server ist down

Benachrichtigungsklassen

DM_NOTIFY_CLASS_ERROR	0x00000001	Benachrichtigungscode enthält Fehlerkennung
DM_NOTIFY_CLASS_WARNING	0x00000002	Benachrichtigungscode enthält Warnung
DM_NOTIFY_CLASS_DATA	0x00000003	Benachrichtigungscode enthält Art der Daten

Benachrichtigungscode's

DM_NOTIFY_SHUTDOWN	0x00000001	Klasse Error : Datenmanager wird beendet
DM_NOTIFY_PROCESSNET_ERROR	0x00000002	Klasse Error : Fehler auf Prozessbus
DM_NOTIFY_SYSNET_ERROR	0x00000003	Klasse Error : Fehler auf Systembus
DM_NOTIFY_QUEUE_50_PERCENT	0x00000001	Klasse Warning : Füllstand der Applikationsqueue 50%
DM_NOTIFY_QUEUE_60_PERCENT	0x00000002	Klasse Warning : Füllstand der Applikationsqueue 60%
DM_NOTIFY_QUEUE_70_PERCENT	0x00000003	Klasse Warning : Füllstand der Applikationsqueue 70%
DM_NOTIFY_QUEUE_80_PERCENT	0x00000004	Klasse Warning : Füllstand der Applikationsqueue 80%
DM_NOTIFY_QUEUE_90_PERCENT	0x00000005	Klasse Warning : Füllstand der Applikationsqueue 90%
DM_NOTIFY_QUEUE_OVERFLOW	0x00000006	Klasse Warning : Applikationsqueue Überlauf
DM_NOTIFY_CYCLES_CHANGED	0x00000010	Klasse Warning, => Zyklen neu einlesen
DM_NOTIFY_MACHINES_CHANGED	0x00000011	Klasse Warning, Rechnerliste
DM_NOTIFY_PROJECT_OPENED	0x00000012	Klasse Warning, Projekt wird geladen
DM_NOTIFY_PROJECT_CLOSED	0x00000013	Klasse Warning, Projekt wird geschlossen
DM_NOTIFY_SYSTEM_LOCALE	0x00000014	Sprachumschaltung (Ressourcen) vornehmen
DM_NOTIFY_DATA_LOCALE	0x00000015	Sprachumschaltung (Projektierungsdaten) vornehmen
DM_NOTIFY_PROJECT_RUNTIME	0x00000016	Projekt im Runtime-Modus aktiviert
DM_NOTIFY_PROJECT_EDIT	0x00000017	Runtime-Modus für Projekt deaktiviert
DM_NOTIFY_HOTKEY_CHANGE	0x00000018	The Hotkeys have changed

DM_NOTIFY_APPLICATION_DATA	0x00000001	Klasse Data : Von Applikation verschickte Daten
DM_NOTIFY_VARIABLE_DATA	0x00000002	Klasse Data : Variablendaten
DM_NOTIFY_SERVERDOWN	0x00000022	
DM_NOTIFY_SERVERUP	0x00000023	

Rechnertyp

Rechnertyp	0	Server
	1	Client

Registrierte Messages

DM_OHIOLANGUAGE "WM_OHIOLANGUAGE"

interne Systemvariablen

DM_VAR_CURRENT_LANGUAGE "OHIOCurrentLanguage"

2.2.1.5 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

DM_E_SYS_ERROR	0x10000000	Falls in dwError1 der Fehler-Struktur CMN_ERROR dieses Bit gesetzt ist, dann enthält dwError2 den Systemfehlercode.
DM_E_OK	0x00000000	Kein Fehler
DM_E_CANCEL	0x00000001	Benutzer hat in einem Dialog "Abbrechen" gewählt
DM_E_FILE	0x00000002	Fehler bei Dateioperation
DM_E_UPDATE	0x00000003	Projekt wird gerade aktualisiert
DM_E_NO_RT_PRJ	0x00000004	Kein Projekt im Runtime Mode
DM_E_NOT_SUPPORTED	0x00000005	Angeforderter Dienst nicht verfügbar
DM_E_ALREADY_CONNECTED	0x00000006	Verbindung zum Datenmanager besteht bereits
DM_E_NOT_CONNECTED	0x00000007	Keine Verbindung zum Datenmanager
DM_E_INVALID_TAID	0x00000008	Ungültige Transaktions-ID
DM_E_INVALID_KEY	0x00000009	Variable nicht gefunden
DM_E_INVALID_TYPE	0x0000000A	ungültiger Variablentyp
DM_E_MAX_LIMIT	0x0000000B	Oberer Grenzwert der Variablen überschritten

DM_E_MIN_LIMIT	0x0000000C	Unterer Grenzwert der Variablen unterschritten
DM_E_MAX_RANGE	0x0000000D	Oberer Grenzwert der Formatkonvertierung überschritten
DM_E_MIN_RANGE	0x0000000E	Unterer Grenzwert der Formatkonvertierung unterschritten
DM_E_ACCESS_FAULT	0x0000000F	unerlaubter Schreibzugriff auf eine Variable
DM_E_TIMEOUT	0x00000010	Timeout-Fehler
DM_E_ALREADY_EXIST	0x00000011	anzulegendes Objekt existiert bereits
DM_E_PARAM	0x00000012	Ungültiger Parameter
DM_E_INV_PRJ	0x00000013	angegebenes Projekt nicht gefunden / geladen
DM_E_UNKNOWN	0x00000014	Unbekannter Fehler
DM_E_OOM	0x00000015	out of memory
DM_E_NOT_CREATED	0x00000016	Projekt konnte nicht erzeugt werden
DM_E_MACHINE_NOT_FOUND	0x00000017	Rechner nicht gefunden
DM_E_NO_INFO_FOUND	0x00000018	keine Startinformationen gefunden
DM_E_INTERNAL	0x00000019	Interner Verarbeitungsfehler
DM_E_INVALID_LOCALE	0x0000001A	Fehlerhafte lokale ID
DM_E_COMMUNICATION	0x0000001B	Fehlerhafte lokale ID
DM_E_DONT_EXIST	0x0000001C	Objekt existiert nicht z.B.: wurde versucht eine Variable anzulegen, und die angegebene Verbindung existiert nicht
DM_E_ALREADY_ACTIVATED	0x0000001D	Projekt befindet sich im Runtime-Modus z.B.: Wenn versucht wird ein Projekt zu aktivieren, das sich bereits im Runtime-Modus befindet.
DM_E_NO_OPEN_PROJECT	0x0000001E	Es ist kein Projekt geöffnet
DM_E_ALREADY_DEACTIVATED	0x0000001F	Projekt befindet sich nicht im Runtime-Modus
DM_E_NO_RIGHTS	0x00000020	falsche CreatorID übergeben
DM_E_NOT_DELETED	0x00000021	Objekt konnte nicht gelöscht werden
DM_E_LICENSE	0x00000022	Softwareschutz: keine Lizenz
DM_E_LICENSE_LIMIT	0x00000023	Softwareschutz: Grenze erreicht/ überschritten
DM_E_INVALID_OBJECTTYPE	0x00000024	Ungültiger Objekttyp
DM_E_OP_REQUIERES_PRJEDITMODE	0x00000025	Projekt befindet sich im Runtime-Modus z.B.: erfordert die Operation erfordert, dass das Projekt nicht aktiviert ist.
DM_E_INTERFACE	0x00000026	interner Fehler bei Zugriff auf Interfaces
DM_E_UNIT_NOT_FOUND	0x00000027	Unit nicht gefunden
DM_E_CONNECTION_NOT_FOUND	0x00000028	Connection nicht gefunden

Siehe auch

CMN_ERROR (Seite 1136)

2.2.1.6 Konvertierungsroutinen (Control Center) (RT Professional)

Übersicht

Nr.	Konvertierungsroutinen für den Datentyp "Vorzeichenbehafteter 8-Bit Wert"
0	CharToSignedByte
1	CharToUnsignedByte
2	CharToUnsignedWord
3	CharToUnsignedDword
4	CharToSignedWord
5	CharToSignedDword
6	CharToMSBByte
7	CharToMSBWord
8	CharToMSBDword
9	CharToBCDByte
10	CharToBCDWord
11	CharToBCDDword
12	CharToSignedBCDByte
13	CharToSignedBCDWord
14	CharToSignedBCDDword
15	CharToExtSignedBCDByte
16	CharToExtSignedBCDWord
17	CharToExtSignedBCDDword
18	CharToAikenByte
19	CharToAikenWord
20	CharToAikenDword
21	CharToSignedAikenByte
22	CharToSignedAikenWord
23	CharToSignedAikenDword
24	CharToExcessByte
25	CharToExcessWord
26	CharToExcessDword
27	CharToSignedExcessByte
28	CharToSignedExcessWord
29	CharToSignedExcessDword
> 29	(CharToSignedByte)

Nr.	Konvertierungsroutinen für den Datentyp "Vorzeichenloser 8-Bit Wert"
0	ByteToUnsignedByte

1	ByteToUnsignedWord
2	ByteToUnsignedDword
3	ByteToSignedByte
4	ByteToSignedWord
5	ByteToSignedDword
6	ByteToBCDByte
7	ByteToBCDWord
8	ByteToBCDDword
9	ByteToAikenByte
10	ByteToAikenWord
11	ByteToAikenDword
12	ByteToExcessByte
13	ByteToExcessWord
14	ByteToExcessDword
> 14	(ByteToUnsignedByte)

Nr.	Konvertierungsroutinen für den Datentyp "Vorzeichenbehafteter 16-Bit Wert"
0	ShortToSignedWord
1	ShortToUnsignedByte
2	ShortToUnsignedWord
3	ShortToUnsignedDword
4	ShortToSignedByte
5	ShortToSignedDword
6	ShortToMSBByte
7	ShortToMSBWord
8	ShortToMSBDword
9	ShortToBCDByte
10	ShortToBCDWord
11	ShortToBCDDword
12	ShortToSignedBCDByte
13	ShortToSignedBCDWord
14	ShortToSignedBCDDword
15	ShortToExtSignedBCDByte
16	ShortToExtSignedBCDWord
17	ShortToExtSignedBCDDword
18	ShortToAikenByte
19	ShortToAikenWord
20	ShortToAikenDword
21	ShortToSignedAikenByte
22	ShortToSignedAikenWord
23	ShortToSignedAikenDword
24	ShortToExcessByte

2.2 Funktionen der Datenhaltung (RT Professional)

25	ShortToExcessWord
26	ShortToExcessDword
27	ShortToSignedExcessByte
28	ShortToSignedExcessWord
29	ShortToSignedExcessDword
> 29	(ShortToSignedWord)

Nr.	Konvertierungsroutinen für den Datentyp "Vorzeichenloser 16-Bit Wert"
0	WordToUnsignedWord
1	WordToUnsignedByte
2	WordToUnsignedDword
3	WordToSignedByte
4	WordToSignedWord
5	WordToSignedDword
6	WordToBCDByte
7	WordToBCDWord
8	WordToBCDDword
9	WordToAikenByte
10	WordToAikenWord
11	WordToAikenDword
12	WordToExcessByte
13	WordToExcessWord
14	WordToExcessDword
15	WordToSimaticBCDCounter
16	WordToSimaticCounter
> 16	(WordToUnsignedWord)

Nr.	Konvertierungsroutinen für den Datentyp "Vorzeichenbehafteter 32-Bit Wert"
0	LongToSignedDword
1	LongToUnsignedByte
2	LongToUnsignedWord
3	LongToUnsignedDword
4	LongToSignedByte
5	LongToSignedWord
6	LongToMSBByte
7	LongToMSBWord
8	LongToMSBDword
9	LongToBCDByte
10	LongToBCDWord
11	LongToBCDDword

12	LongToSignedBCDByte
13	LongToSignedBCDWord
14	LongToSignedBCDDword
15	LongToExtSignedBCDByte
16	LongToExtSignedBCDWord
17	LongToExtSignedBCDDword
18	LongToAikenByte
19	LongToAikenWord
20	LongToAikenDword
21	LongToSignedAikenByte
22	LongToSignedAikenWord
23	LongToSignedAikenDword
24	LongToExcessByte
25	LongToExcessWord
26	LongToExcessDword
27	LongToSignedExcessByte
28	LongToSignedExcessWord
29	LongToSignedExcessDword
30	LongToSimaticBCDTimer
31	(LongToSignedDword)
32	(LongToSignedDword)
33	(LongToSignedDword)
34	LongToSimaticTimer
> 34	(LongToSignedDword)

Nr.	Konvertierungsroutinen für den Datentyp "Vorzeichenloser 32-Bit Wert"
0	DwordToUnsignedDword
1	DwordToUnsignedByte
2	DwordToUnsignedWord
3	DwordToSignedByte
4	DwordToSignedWord
5	DwordToSignedDword
6	DwordToBCDByte
7	DwordToBCDWord
8	DwordToBCDDword
9	DwordToAikenByte
10	DwordToAikenWord
11	DwordToAikenDword
12	DwordToExcessByte
13	DwordToExcessWord
14	DwordToExcessDword
15	DwordToSimaticBCDTimer

2.2 Funktionen der Datenhaltung (RT Professional)

16	(DwordToUnsignedDword)
17	(DwordToUnsignedDword)
18	(DwordToUnsignedDword)
19	DwordToSimaticTimer
> 19	(DwordToUnsignedDword)

Nr.	Konvertierungsroutinen für den Datentyp "Gleitkommazahl 32-Bit IEEE 754"
0	FloatToFloat
1	FloatToUnsignedByte
2	FloatToUnsignedWord
3	FloatToUnsignedDword
4	FloatToSignedByte
5	FloatToSignedWord
6	FloatToSignedDword
7	FloatToDouble
8	FloatToMSBByte
9	FloatToMSBWord
10	FloatToMSBDword
11	FloatToBCDByte
12	FloatToBCDWord
13	FloatToBCDDword
14	FloatToSignedBCDByte
15	FloatToSignedBCDWord
16	FloatToSignedBCDDword
17	FloatToExtSignedBCDByte
18	FloatToExtSignedBCDWord
19	FloatToExtSignedBCDDword
20	FloatToAikenByte
21	FloatToAikenWord
22	FloatToAikenDword
23	FloatToSignedAikenByte
24	FloatToSignedAikenWord
25	FloatToSignedAikenDword
26	FloatToExcessByte
27	FloatToExcessWord
28	FloatToExcessDword
29	FloatToSignedExcessByte
30	FloatToSignedExcessWord
31	FloatToSignedExcessDword
32	FloatToSimaticBCDTimer
33	(FloatToFloat)
34	(FloatToFloat)

35	(FloatToFloat)
36	FloatToS5Float
37	FloatToSimaticTimer
> 37	(FloatToFloat)

Nr.	Konvertierungsroutinen für den Datentyp "Gleitkommazahl 64-Bit IEEE 754"
0	DoubleToDouble
1	DoubleToUnsignedByte
2	DoubleToUnsignedWord
3	DoubleToUnsignedDword
4	DoubleToSignedByte
5	DoubleToSignedWord
6	DoubleToSignedDword
7	DoubleToFloat
8	DoubleToMSBByte
9	DoubleToMSBWord
10	DoubleToMSBDword
11	DoubleToBCDByte
12	DoubleToBCDWord
13	DoubleToBCDDword
14	DoubleToSignedBCDByte
15	DoubleToSignedBCDWord
16	DoubleToSignedBCDDword
17	DoubleToExtSignedBCDByte
18	DoubleToExtSignedBCDWord
19	DoubleToExtSignedBCDDword
20	DoubleToAikenByte
21	DoubleToAikenWord
22	DoubleToAikenDword
23	DoubleToSignedAikenByte
24	DoubleToSignedAikenWord
25	DoubleToSignedAikenDword
26	DoubleToExcessByte
27	DoubleToExcessWord
28	DoubleToExcessDword
29	DoubleToSignedExcessByte
30	DoubleToSignedExcessWord
31	DoubleToSignedExcessDword
32	DoubleToSimaticBCDTimer
33	(DoubleToDouble)
34	(DoubleToDouble)
35	(DoubleToDouble)

36	DoubleToS5Float
37	DoubleToSimaticTimer
> 37	(DoubleToDouble)

2.2.2 Strukturen (RT Professional)

2.2.2.1 DM_CONNECTION_DATA (RT Professional)

Deklaration

```
typedef struct {
    CHAR          szConnection[
                    MAX_DM_CONNECTION_NAME +3];
    CHAR          szUnitName[MAX_DM_UNIT_NAME +1];
    CHAR          szCommon[MAX_DM_CON_COMMON +1];
    CHAR          szSpecific[MAX_DM_CON_SPECIFIC +1];
    DWORD         dwVarNum;
}
DM_CONNECTION_DATA;
```

Members

szConnection

Name der logischen Verbindung

szUnitName

Namen der Kanal-Unit

szCommon

Der Parameter ist für spätere Erweiterungen reserviert.

szSpecific

szSpecific enthält die Adressparameter der Verbindung, z.B. Ethernetadresse, Steckplatznummer.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der bei den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

dwVarNum

Anzahl der zugeordneten Variablen

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_CONNECTION_PROC (Seite 446)	Verbindungsdaten auflisten (Callback)
-------------------------------------	---------------------------------------

Siehe auch

DM_ENUM_CONNECTION_PROC (Seite 446)

2.2.2.2 DM_CONNKEY (RT Professional)**Deklaration**

```
typedef struct {
    CHAR      szName[ MAX_DM_CONNECTION_NAME + 1 ];
    LPVOID    lpvUserData;
}
DM_CONNKEY;
```

Members**szName**

Name der logischen Verbindung

lpvUserData

Zeiger auf applikationsspezifische Daten

Benötigte Dateien

dmclient.h

API - Funktionen

DMEnumConnectionData (Seite 443)	Verbindungsdaten auflisten
----------------------------------	----------------------------

Siehe auch

DMEnumConnectionData (Seite 443)

2.2.2.3 DM_CYCLE_INFO (RT Professional)

Deklaration

```
typedef struct {  
    DWORD    dwCycleTime;  
    DWORD    dwCycleIndex;  
    char     szDescription[ MAX_DM_CYCLE_NAME + 1 ];  
}  
DM_CYCLE_INFO;
```

Members

dwCycleTime

Zeitbasis des Aktualisierungszyklus

dwCycleIndex

kennzeichnet die Ordnung innerhalb der Liste der Aktualisierungszyklen.

"Bei Änderung"	Index: 0
"250ms"	Index: 1
:	:
"Anwenderzyklus 5"	Index: 15

szDescription

Bezeichnung des Aktualisierungszyklus.

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_CYCLES_PROC (Seite 291)	Aktualisierungszyklen auflisten (Callback)
---------------------------------	--

Siehe auch

DM_ENUM_CYCLES_PROC (Seite 291)

2.2.2.4 DM_DATA_SERVICE (RT Professional)

Deklaration

```
typedef struct {
    DWORD    dwTeleType;
    char     szService[MAX_DM_SERVICE_NAME + 1];
    char     szSendingApp[MAX_DM_APP_NAME + 1];
    DWORD    dwSendingMachine;
    DWORD    dwDataSize;
    BYTE     byData[1];
}
DM_DATA_SERVICE;
```

Members

dwTeleType

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

szService

Name des Datentransportkanals. Dieser Name entspricht dem bei der Installation des Service (DMInstallDataService) vergebenen.

szSendingApp

logischer Name des Absenders. Dieser Name entspricht dem bei DMConnect angegebenen Applikationsnamen.

dwSendingMachine

Index des Rechners, von dem das Paket verschickt wurde (0 - .63). Für die API-Funktionen sind alle in der Rechnerliste eingetragenen Rechner über ihren Index ansprechbar. Dabei besitzt der erste Eintrag in der Rechnerliste den Index "0".

dwDataSize

Größe des Datenpakets in Byte: Data[0] ... byData[dwDataSize - 1]

byData

Zeiger auf die Daten

Benötigte Dateien

dmclient.h

API - Funktionen

DM_DATA_SERVICE_PROC (Seite 319)	Datentransportkanäle auflisten (Callback), Datentransportkanal installieren (Callback)
----------------------------------	--

Siehe auch

DM_DATA_SERVICE_PROC (Seite 319)

2.2.2.5 DM_DIRECTORY_INFO (RT Professional)

Deklaration

```
typedef struct {  
    char    szProjectDir[_MAX_PATH + 1 ];  
    char    szProjectAppDir[_MAX_PATH + 1 ];  
    char    szGlobalLibDir[_MAX_PATH + 1 ];  
    char    szProjectLibDir[_MAX_PATH + 1 ];  
    char    szLokalProjectAppDir[_MAX_PATH + 1 ];  
}  
DM_DIRECTORY_INFO;
```

Members

szProjectDir

Vollständige Pfadangabe des Projektverzeichnis, z.B. D:\WinCC\Projekt1

szProjectAppDir

Vollständige Pfadangabe des Unterverzeichnisses der Applikation im Projektverzeichnis, z.B.: D:\WinCC\Projekt1\GraCS

szGlobalLibDir

Vollständige Pfadangabe des projektübergreifenden Library-Verzeichnisses, z.B.: D:\WinCC\aplib

szProjectLibDir

Vollständige Pfadangabe des projektbezogenen Library-Verzeichnisses, z.B. D:\WinCC\Projekt1\Library

szLokalProjectAppDir

Vollständige Pfadangabe des Unterverzeichnisses der Applikation im Projektverzeichnis auf dem lokalen Rechner.

Benötigte Dateien

dmclient.h

API - Funktionen

DMGetProjectDirectory (Seite 308)	Pfad- und Dateinamen der Projektierungsdaten ermitteln
-----------------------------------	--

Siehe auch

DMGetProjectDirectory (Seite 308)

2.2.2.6 DM_DLGOPTIONS (RT Professional)**Deklaration**

```
typedef struct {
    DWORD                dwFlags;
    LPRECT               lprcPreference;
    DM_TEST_DROP_TARGET_PROC lpfnTestDropTarget;
    DM_DROP_TARGET_PROC   lpfnDropTarget;
    LPVOID               lpvUser;
}
DM_DLGOPTIONS;
```

Members**dwFlags**

Über dwFlags wird das Verhalten des Dialoges, bzw. des Dialogfensters festgelegt:

DM_DLG_NOT_MODAL	Der Variablendialog ist nicht modal.
DM_DLG_NOT_MOVEABLE	Das Dialogfenster ist nicht verschiebbar.
DM_DLG_DRAGDROP	Objekte im Dialogfenster können mit Drag & Drop bearbeitet werden.

lprcPreference

Zeiger auf eine Struktur vom Typ RECT mit den Angaben zur Größe des Dialogfensters. Ist lprcPreference == NULL, so erscheint der Dialog zentriert und mit einer vordefinierten Größe.

lpfnTestDropTarget

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

lpfnDropTarget

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

lpvUser

Zeiger auf applikationsspezifische Daten.

Benötigte Dateien

dmclient.h

API - Funktionen

DMSHOWVARDATABASE (Seite 410)	Variablen-Auswahldialog öffnen
DMSHOWVARDATABASEMULTI (Seite 415)	Variablen-Auswahldialog öffnen

Siehe auch

DMSHOWVARDATABASE (Seite 410)

DMSHOWVARDATABASEMULTI (Seite 415)

2.2.2.7 DM_FORMAT_INFO (RT Professional)

Deklaration

```
typedef struct {  
    DWORD    dwID;  
    char     szName[MAX_DM_FORMAT_NAME + 1];  
}  
DM_FORMAT_INFO;
```

Members

dwID

Nummer der zu verwendenden Konvertierungsroutine. Näheres finden Sie im Abschnitt "Konvertierungsroutinen".

szName

Name der zu verwendenden Konvertierungsroutine. Näheres finden Sie im Abschnitt "Konvertierungsroutinen".

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_FORMATS_PROC (Seite 288)	Formatkonvertierungen auflisten (Callback)
----------------------------------	--

Siehe auch

DM_ENUM_FORMATS_PROC (Seite 288)

2.2.2.8 DM_MACHINE_TABLE (RT Professional)

Deklaration

```
typedef struct {
    LONG                nNumMachines;
    LONG                nLocalMachine;
    DM_SD_TARGET_MACHINE tm[MAX_DM_OHIO_MACHINES];
}
DM_MACHINE_TABLE;
```

Members

nNumMachines

Anzahl der Rechner im Projekt (darf die max. Anzahl MAX_DM_OHIO_MACHINES nicht übersteigen).

nLocalMachine

Index des Eintrags des lokalen Rechners in der Rechnerliste.

Für die API-Funktionen sind alle in der Rechnerliste eingetragenen Rechner über ihren Index ansprechbar. Dabei besitzt der erste Eintrag in der Rechnerliste den Index "0".

tm

Vollständiges Array vom Typ DM_SD_TARGET_MACHINE (Seite 225) mit Informationen zu den am Projekt beteiligten Rechnern. Es werden hier aber nur nNumMachines dieser Strukturen übergeben (damit max. nutzbarer Index = (nMumMachines - 1)).

Benötigte Dateien

dmclient.h

API - Funktionen

DMGetMachineTable (Seite 301)	Rechnerliste abfragen
-------------------------------	-----------------------

Siehe auch

DM_SD_TARGET_MACHINE (Seite 225)

DMGetMachineTable (Seite 301)

2.2.2.9 DM_PROJECT_INFO (RT Professional)

Deklaration

```
typedef struct {  
    char    szProjectFile[ _MAX_PATH + 1 ];  
    char    szDSNName[ MAX_DM_DSN_NAME + 1 ];  
    DWORD   dwDataLocale;  
}  
DM_PROJECT_INFO;
```

Members

szProjectFile

Dateiname des Projektes, einschließlich Pfad und Extension.

szDSNName

Data-Source Name der Datenbank

dwDataLocale

Code der Sprache die bei der Projektierung verwendet wird.

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_OPENED_PROJECTS_PROC (Seite 306)	Projekte (geöffnete) auflisten (Callback)
DMEnumOpenedProjects (Seite 305)	Projektinformationen ermitteln

Siehe auch

DM_ENUM_OPENED_PROJECTS_PROC (Seite 306)

DMEnumOpenedProjects (Seite 305)

DMGetProjectInformation (Seite 309)

2.2.2.10 DM_SEND_DATA_STRUCT (RT Professional)

Deklaration

```
typedef struct {
    BOOL                fHighPriority;
    char                szService[MAX_DM_SERVICE_NAME + 1];
    DWORD               dwTargetMachineFlags;
    DWORD               dwTargetMachines;
    DM_SD_TARGET_MACHINE dmTargetMachine[
        MAX_DM_OHIO_MACHINES];
    DWORD               dwTargetApps;
    DM_SD_TARGET_APP   dmTargetApp[
        MAX_DM_OHIO_APPLICATIONS];
    DWORD               dwDataSize;
    BYTE                byData[1];
}
DM_SEND_DATA_STRUCT;
```

Members

fHighPriority

fHighPriority kennzeichnet mit welcher Priorität der Datenaustausch erfolgen soll:

1	hohe Priorität
0	normale Priorität

szService

Name des zu verwendenden Service. Damit eine Applikation die Daten empfangen kann, muss sie nicht nur den Bedingungen von dwTargetMachineFlags genügen, sondern auch den entsprechenden Service über DMInstallDataService installiert haben.

dwTargetMachineFlags

dwTargetMachineFlags kennzeichnet die Applikationen, an die mit der Funktion DMSendApplicationData Daten versandt werden sollen.

DM_SD_LOCAL	Nur an lokale Applikationen (schließt alle anderen Flags aus !) Alle anderen Flags sind zur Zeit nicht erlaubt und sind für zukünftige Erweiterungen reserviert!
DM_SD_ALL_MACHINES	An alle Rechner im Projekt. (Der Parameter ist für spätere Erweiterungen reserviert)
DM_SD_ALL_SERVERS	An alle Server im Projekt. (Der Parameter ist für spätere Erweiterungen reserviert)
DM_SD_ALL_CLIENTS	An alle Clients im Projekt. (Der Parameter ist für spätere Erweiterungen reserviert)

DM_SD_RELATED_MACHINES	An alle Clients, die zum gleichen Server gehören wie der lokale Rechner (Der Parameter ist für spätere Erweiterungen reserviert)
DM_SD_FIRST_SERVER	(Der Parameter ist für spätere Erweiterungen reserviert)
DM_SD_PRIMARY_SERVER	(Der Parameter ist für spätere Erweiterungen reserviert)
DM_SD_EXCEPT_LOCAL	An alle Rechner, außer den lokalen, die der Adressbeschreibung genügen. (Der Parameter ist für spätere Erweiterungen reserviert)

dwTargetMachines

Anzahl der ausgefüllten Strukturen in dmTargetMachine.

Reserviert für zukünftige Erweiterung, und muss mit 0 besetzt werden.

dmTargetMachine

Zeiger auf die Struktur DM_SD_TARGET_MACHINE (Seite 225) zur Spezifikation des Rechners an den die Daten verschickt werden sollen. Dabei ist nur der Parameter szMachineName von Bedeutung.

Reserviert für zukünftige Erweiterung, und sollte komplett mit 0L initialisiert werden.

dwTargetApps

Anzahl der ausgefüllten Strukturen in dmTargetApp.

dmTargetApp

Die Strukturen vom Typ DM_SD_TARGET_APP (Seite 226) enthalten die Namen der Applikationen, an die die Daten verschickt werden sollen.

dwDataSize

Größe der zu versendenden Daten in Byte

byData

Array, das die zu versendenden Daten enthält

Bemerkungen

Die Funktion DMSendApplicationData ist nur lokal implementiert.

Die in dieser Struktur vorhanden Member für Remote-Zugriffe sind für zukünftige Erweiterung reserviert.

Das dwTargetMachineFlags kann nur mit DM_SD_LOCAL besetzt werden, alle anderen Angaben führen zu Fehler, und dwTargetMachines muss 0L sein.

Benötigte Dateien

dmclient.h

API - Funktionen

DMSendApplicationData (Seite 323)	Datenaustausch mit Applikationen
-----------------------------------	----------------------------------

Siehe auch

DMSendApplicationData (Seite 323)

DM_SD_TARGET_MACHINE (Seite 225)

DM_SD_TARGET_APP (Seite 226)

2.2.2.11 DM_SD_TARGET_MACHINE (RT Professional)

Deklaration

```
typedef struct {
    BOOL    fServer;
    BOOL    fLocal;
    char    szMachineName[
        MAX_COMPUTERNAME_LENGTH + 1];
}
DM_SD_TARGET_MACHINE;
```

Members

fServer

Kennzeichnet ob es sich bei dem Rechner um einen Server oder einen Client handelt.

Der Parameter ist für die Belange des Datenaustauschs (DM_SEND_DATA_STRUCT) nicht von Interesse.

fLocal

Kennzeichnet ob die abfragende Applikation auf dem lokalen Rechner oder einem anderen projektierten Rechner im Netzwerk läuft.

Der Parameter ist für die Belange des Datenaustauschs (DM_SEND_DATA_STRUCT) nicht von Interesse.

szMachineName

Rechnername

Bemerkungen

DM_SD_TARGET_MACHINE ist Teil der Strukturen DM_MACHINE_TABLE (Seite 221) und DM_SEND_DATA_STRUCT (Seite 223).

Benötigte Dateien

dmclient.h

Siehe auch

DM_MACHINE_TABLE (Seite 221)

DM_SEND_DATA_STRUCT (Seite 223)

2.2.2.12 DM_SD_TARGET_APP (RT Professional)

Deklaration

```
typedef struct {  
    char    szAppName[MAX_DM_APP_NAME + 1];  
}  
DM_SD_TARGET_APP;
```

Members

szAppName

Verwenden Sie für szAppName den beim Aufruf von DMConnect verwendeten Namen der Applikation.

Bemerkungen

DM_SD_TARGET_APP ist Teil der Struktur DM_SEND_DATA_STRUCT (Seite 223).

Benötigte Dateien

dmclient.h

Siehe auch

DM_SEND_DATA_STRUCT (Seite 223)

2.2.2.13 DM_TYPEREF (RT Professional)

Deklaration

```
typedef struct {
    DWORD    dwType;
    DWORD    dwSize;
    char     szTypeName[MAX_DM_TYPE_NAME + 1];
}
DM_TYPEREF;
```

Members

dwType

In dwType wird der Typ der Variablen spezifiziert:

DM_VARTYPE_BIT	Binäre Variable
DM_VARTYPE_SBYTE	Vorzeichenbehafteter 8-Bit Wert
DM_VARTYPE_BYTE	Vorzeichenloser 8-Bit Wert
DM_VARTYPE_SWORD	Vorzeichenbehafteter 16-Bit Wert
DM_VARTYPE_WORD	Vorzichenloser 16-Bit Wert
DM_VARTYPE_SDWORD	Vorzeichenbehafteter 32-Bit Wert
DM_VARTYPE_DWORD	Vorzeichenloser 32-Bit Wert
DM_VARTYPE_FLOAT	Gleitkommazahl 32-Bit IEEE 754
DM_VARTYPE_DOUBLE	Gleitkommazahl 64-Bit IEEE 754
DM_VARTYPE_TEXT_8	Textvariable 8-Bit Zeichensatz
DM_VARTYPE_TEXT_16	Textvariable 16-Bit Zeichensatz
DM_VARTYPE_RAW	Rohdatentyp
DM_VARTYPE_STRUCT	Strukturvariable
DM_VARTYPE_TEXTREF	Text-Referenz aus Textbib
Werte >=1024	Anwenderdefinierte Strukturtypen Diese Typ-Id wird in manchen Aufrufen, z.B. bei Filtern benötigt, um einen ganz bestimmten Strukturtyp zu identifizieren.

dwSize

In dwSize wird die Länge des Datentyps (in Byte) auf der OS angegeben.

szTypeName

In szTypeName steht bei Strukturvariablen der Name des Strukturtyps.

Bemerkungen

DM_TYPEREF wird innerhalb der Strukturen DM_VAR_UPDATE_STRUCT (Seite 228) und DM_VARIABLE_DATA (Seite 237) verwendet.

Benötigte Files

dmclient.h

API - Funktionen

DMGetVarType (Seite 378)	Datentyp einer Variablen ermitteln
--------------------------	------------------------------------

Siehe auch

DM_VAR_UPDATE_STRUCT (Seite 228)

DMGetVarType (Seite 378)

DM_VARIABLE_DATA (Seite 237)

DM_VAR_UPDATE_STRUCTEX (Seite 230)

DM_VARIABLE_DATA4 (Seite 239)

2.2.2.14 DM_VAR_UPDATE_STRUCT (RT Professional)

Deklaration

```
typedef struct {
    DM_TYPEREF    dmTypeRef;
    DM_VARKEY     dmVarKey;
    VARIANT       dmValue;
    DWORD         dwState;
}
DM_VAR_UPDATE_STRUCT;
```

Members

dmTypeRef

Die Struktur DM_TYPEREF (Seite 227) enthält Angaben zum Variablentyp.

Ausnahme: Bei zyklischen Anforderungen wird in diese Struktur aus Performancegründen nichts eingetragen.

dmVarKey

Über die Struktur DM_VARKEY (Seite 241) wird die zu bearbeitende Variable spezifiziert.

dmValue

Wert der Variable.

Hinweis

Der VARIANT dmValue muss vor Erstbenutzung mit VariantInit(&dmValue) initialisiert und nach Benutzung mit VariantClear(&dmValue) wieder freigegeben werden.

Nur vor Erstbenutzung kann ein Array von mehreren DM_VAR_UPDATE_STRUCT mit ZeroMemory() oder memset() vorinitialisiert werden. Dies wirkt sich auf den hier enthaltenen VARIANT dmValue wie ein VariantInit(&dmValue) aus, da 0 einem VT_EMPTY entspricht.

Die Struktur DM_VAR_UPDATE_STRUCT darf nicht mit ZeroMemory() oder memset() gelöscht werden, wenn sie schon einmal benutzt wurde. Der VARIANT enthält dann Daten, die auch vom Typ VT_BSTR sein könnten.

Ist der Varianttyp vom Typ VT_BSTR kommt es zu Problemen mit der Speicherverwaltung (MemoryLeak), wenn kein VariantClear(&dmValue) gemacht wird, bevor z.B. allokierte Strukturen DM_VAR_UPDATE_STRUCT mit delete[] gelöscht werden

	(0x0000)	kein Fehler
DM_VARSTATE_NOT_ESTABLISHED	(0x0001)	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	(0x0002)	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	(0x0004)	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	(0x0008)	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	(0x0010)	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	(0x0020)	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	(0x0040)	Formatgrenze unterschritten
DM_VARSTATE_CONVERSION_ERROR	(0x0080)	Anzeige Wandlungsfehler (in Verbindung mit DM_VARSTATE_..._RANGE)
DM_VARSTATE_STARTUP_VALUE	(0x0100)	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	(0x0200)	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	(0x0400)	Adressierungsfehler im Kanal
DM_VARSTATE_INVALID_KEY	(0x0800)	Variable nicht gefunden / nicht vorhanden
DM_VARSTATE_ACCESS_FAULT	(0x1000)	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	(0x2000)	Timeout / keine Rückmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	(0x4000)	Server ist down

Benötigte Dateien

dmclient.h

API - Funktionen

DM_NOTIFY_VARIABLE_PROC (Seite 458)	Variablenwerte ermitteln (Callback)
DMGetValue (Seite 341)	Variablenwerte ermitteln

Siehe auch

DM_NOTIFY_VARIABLE_PROC (Seite 458)

DM_VARKEY (Seite 241)

DMGetValue (Seite 341)

DM_TYPEREF (Seite 227)

2.2.2.15 DM_VAR_UPDATE_STRUCTEX (RT Professional)

Deklaration

```
typedef struct {  
    DM_TYPEREF    dmTypeRef;  
    DM_VARKEY     dmVarKey;  
    VARIANT       dmValue;  
    DWORD         dwState;  
    DWORD         dwQualityCode;  
}  
DM_VAR_UPDATE_STRUCTEX;
```

Members

dmTypeRef

Die Struktur DM_TYPEREF (Seite 227) enthält Angaben zum Variablentyp.

Ausnahme: Bei zyklischen Anforderungen wird in diese Struktur aus Performancegründen nichts eingetragen.

dmVarKey

Über die Struktur DM_VARKEY (Seite 241) wird die zu bearbeitende Variable spezifiziert.

dmValue

Wert der Variable.

Hinweis

Der VARIANT dmValue muss vor Erstbenutzung mit VariantInit(&dmValue) initialisiert und nach Benutzung mit VariantClear(&dmValue) wieder freigegeben werden.

Nur vor Erstbenutzung kann ein Array von mehreren DM_VAR_UPDATE_STRUCT mit ZeroMemory() oder memset() vorinitialisiert werden. Dies wirkt sich auf den hier enthaltenen VARIANT dmValue wie ein VariantInit(&dmValue) aus, da 0 einem VT_EMPTY entspricht.

Die Struktur DM_VAR_UPDATE_STRUCTEX darf nicht mit ZeroMemory() oder memset() gelöscht werden, wenn sie schon einmal benutzt wurde. Der VARIANT enthält dann Daten, die auch vom Typ VT_BSTR sein könnten.

Ist der Varianttyp vom Typ VT_BSTR kommt es zu Problemen mit der Speicherverwaltung (MemoryLeak), wenn kein VariantClear(&dmValue) gemacht wird, bevor z.B. allokierte Strukturen DM_VAR_UPDATE_STRUCTEX mit delete[] gelöscht werden.

dwState

Kennzeichnet, ob der Wert der Variable erfolgreich geändert werden konnte oder ob Fehler aufgetreten sind:

DM_VARSTATE_NOT_ESTABLISHED	0x0001	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	0x0002	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	0x0004	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	0x0008	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	0x0010	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	0x0020	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	0x0040	Formatgrenze unterschritten
DM_VARSTATE_CONVERSION_ERROR	0x0080	Anzeige Wandlungsfehler (in Verbindung mit DM_VARSTATE_..._RANGE)
DM_VARSTATE_STARTUP_VALUE	0x0100	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	0x0200	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	0x0400	Adressierungsfehler im Kanal
DM_VARSTATE_INVALID_KEY	0x0800	Variable nicht gefunden / nicht vorhanden
DM_VARSTATE_ACCESS_FAULT	0x1000	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	0x2000	Timeout / keine Rueckmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	0x4000	Server ist down

dwQualityCode

Quality Code des Variablen-Wertes.

Die Abbildung auf den Quality Code ist wie folgt:

	WinCC-DM Variablenstatus	Quality Code nach Profibus PA/OPC
DM_VARSTATE_SERVERDOWN (0x4000)	Server ist down	Bad, out of service, 0x1C

2.2 Funktionen der Datenhaltung (RT Professional)

DM_VARSTATE_HARDWARE_ERROR (0x0004)	Netzwerkbaugruppe defekt	Bad, device failure, 0x0C
DM_VARSTATE_NOT_ESTABLISHED (0x0001)	Verbindung zum Partner nicht aufgebaut	Bad, not connected, 0x08
DM_VARSTATE_TIMEOUT (0x2000)	Timeout / keine Rueckmeldung vom Kanal	Uncertain, last usable value, 0x44
DM_VARSTATE_HANDSHAKE_ERROR (0x0002)	Protokollfehler	Bad, no communication (no usable value), 0x18
DM_VARSTATE_ADDRESS_ERROR (0x0400)	Adressierungsfehler im Kanal	Bad, configuration error, value not accepted, 0x04
DM_VARSTATE_INVALID_KEY (0x0800)	Variable nicht gefunden / nicht vorhanden	Bad, configuration error, value not accepted, 0x04
DM_VARSTATE_ACCESS_FAULT (0x1000)	Zugriff auf Variable nicht erlaubt	Bad, configuration error, value not accepted, 0x04
DM_VARSTATE_MAX_RANGE (0x0020)	Formatgrenze überschritten	Uncertain, engineering unit range violation, high limit gesetzt, 0x56
DM_VARSTATE_MIN_RANGE (0x0040)	Formatgrenze unterschritten	Uncertain, engineering unit range violation, low limit gesetzt, 0x55
DM_VARSTATE_CONVERSION_ERROR (0x0080)	Anzeige Wandlungsfehler	Uncertain, engineering unit range violation, ohne limits gesetzt, 0x54
DM_VARSTATE_MAX_LIMIT (0x0008)	projektierte Obergrenze überschritten	Abbilden auf das Limit-Bit HIGH, 0x01, zusammen mit dem sonstigen Variablenstatus verodern, z.B. good (cascade), ok, ergibt 0xC2
DM_VARSTATE_MIN_LIMIT (0x0010)	projektierte Untergrenze unterschritten	Abbilden auf das Limit-Bit LOW, 0x02, zusammen mit dem sonstigen Variablenstatus verodern, z.B. good (cascade), ok, ergibt 0xC1
DM_VARSTATE_STARTUP_VALUE (0x0100)	Initialisierungswert der Variablen	Uncertain, initial value, 0x4C
DM_VARSTATE_DEFAULT_VALUE (0x0200)	Ersatzwert der Variablen	Uncertain, substitute-set, 0x48

Benötigte Dateien

dmclient.h

API - Funktionen

DM_NOTIFY_VARIABLEEX_PROC (Seite 469)	Variablenwerte ermitteln (Callback)
DMGetValueEx (Seite 343)	Variablenwerte ermitteln

Siehe auch

DM_NOTIFY_VARIABLEEX_PROC (Seite 469)

DM_TYPEREF (Seite 227)

DM_VARKEY (Seite 241)

DMGetValueEx (Seite 343)

2.2.2.16 DM_VARFILTER (RT Professional)

Deklaration

```
typedef struct {
    DWORD        dwFlags;
    DWORD        dwNumTypes;
    LPDWORD      pdwTypes;
    LPSTR        lpszGroup;
    LPSTR        lpszName;
    LPSTR        lpszConn;
}
DM_VARFILTER;
```

Members

dwFlags

Über den Parameter dwFlags kann ein Selektionskriterium für die Variablen gesetzt werden:

DM_VARFILTER_TYPE	0x00000001	Als Selektionskriterium wird der Variablentyp (pdwTypes) verwendet.
DM_VARFILTER_GROUP	0x00000002	Als Selektionskriterium wird der Gruppenname (lpszGroup) verwendet.
DM_VARFILTER_NAME	0x00000004	Als Selektionskriterium wird der Variablenname (lpszName) verwendet.
DM_VARFILTER_CONNECTION	0x00000008	Als Selektionskriterium wird der Name der logischen Verbindung (lpszConn) verwendet.
DM_VARFILTER_FAST_CALLBACK	0x00010000	(Flag für DMEEnumVariables): Ist dieses Flag gesetzt, werden die Callback-Daten nicht mehr gepuffert. Dadurch liefert die Callback die Daten direkt von der Datenbank. Der Vorteil des Flags ist eine schnellere Reaktion, bis die erste Callback gesendet wird. Gleichzeitig wird bei großen Mengengerüsten zusätzliche Speicherbelastung vermieden. Vorsicht Sie dürfen keine andere DMClient-Funktion in dieser Callback aufrufen, da es sonst zu einer Blockade kommen kann.
DM_VARFILTER_LOCAL_ONLY	0x00020000	(Flag für DMEEnumVariables): Wenn dieses Flag gesetzt ist, werden nur lokale Variablen enumeriert. Variablen von anderen im Projekt verwendeten Servern werden nicht angefordert.

dwNumTypes

Anzahl der in pdwTypes angegebenen Variablentypen.

pdwTypes

pdwTypes kennzeichnet die Variablentypen, die als Selektionskriterium verwendet werden sollen.

DM_VARTYPE_BIT	Binäre Variable
DM_VARTYPE_SBYTE	Vorzeichenbehafteter 8-Bit Wert
DM_VARTYPE_BYTE	Vorzeichenloser 8-Bit Wert
DM_VARTYPE_SWORD	Vorzeichenbehafteter 16-Bit Wert
DM_VARTYPE_WORD	Vorzeichenloser 16-Bit Wert
DM_VARTYPE_SDWORD	Vorzeichenbehafteter 32-Bit Wert
DM_VARTYPE_DWORD	Vorzeichenloser 32-Bit Wert
DM_VARTYPE_FLOAT	Gleitkommazahl 32-Bit IEEE 754
DM_VARTYPE_DOUBLE	Gleitkommazahl 64-Bit IEEE 754
DM_VARTYPE_TEXT_8	Textvariable 8-Bit Zeichensatz
DM_VARTYPE_TEXT_16	Textvariable 16-Bit Zeichensatz

Die Typen DM_VARTYPE_RAW, DM_VARTYPE_STRUCT und DM_VARTYPE_TEXTREF werden nicht unterstützt.

Geben Sie die entsprechende TypID an, um strukturierte Variablen mit selbstdefinierten Strukturen zu selektieren. Verwenden Sie zum Ermitteln der TypID die Funktion GAPIEnumTypes.

lpszGroup

Zeiger auf den Namen einer Variablengruppe. Dieser Name soll als Selektionskriterium verwendet werden. Die Angabe von Wildcards ist nicht möglich.

lpszName

Zeiger auf den Namen einer Variablen. Dieser Name soll als Selektionskriterium verwendet werden. Die Angabe von Wildcards ist nicht möglich.

lpszConn

Zeiger auf den Namen einer logischen Verbindung. Dieser Name soll als Selektionskriterium verwendet werden. Die Angabe von Wildcards ist nicht möglich.

Benötigte Dateien

dmclient.h

API - Funktionen

DMEEnumVariables (Seite 337)	Variablennamen auflisten
DMSHowVarDatabase (Seite 410)	Variablenauswahldialog öffnen
DMSHowVarDatabaseMulti (Seite 415)	Variablenauswahldialog öffnen

Siehe auch

DMEnumVariables (Seite 337)

DMShowVarDatabase (Seite 410)

DMShowVarDatabaseMulti (Seite 415)

2.2.2.17 DM_VARGRP_DATA (RT Professional)**Deklaration**

```
typedef struct {
    CHAR        szName[ MAX_DM_VAR_NAME + 1 ];
    DWORD       dwCreatorID;
    WORD        dwVarNum;
    LPVOID      lpvUserData;
}
DM_VARGRP_DATA;
```

Members**szName**

Name der Variablengruppe

dwCreatorID

In der Erzeugerkennung können Sie Angaben treffen, von wem ein Objekt angelegt wurde. Die Werte 0 – 10100 sowie 11000 - 11100 sind intern oder für bestimmte Systeme reserviert.

dwVarNum

Anzahl der Variablen innerhalb der Variablengruppe

lpvUserData

Zeiger auf applikationsspezifische Daten.

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_VARGRP_PROC (Seite 336)	Informationen zu einer Variablengruppe auflisten (Callback)
---------------------------------	---

Siehe auch

DM_ENUM_VARGRP_PROC (Seite 336)

2.2.2.18 DM_VARGRPKEY (RT Professional)

Deklaration

```
typedef struct {  
    CHAR        szName[ MAX_DM_VAR_NAME + 1 ];  
    LPVOID      lpvUserData;  
}  
DM_VARGRPKEY;
```

Members

szName

Name der Variablengruppe

lpvUserData

Zeiger auf applikationsspezifische Daten.

Benötigte Dateien

dmclient.h

API - Funktionen

DMEnumVarGrpData (Seite 332)	Informationen zu einer Variablengruppe auflisten
------------------------------	--

Siehe auch

DMEnumVarGrpData (Seite 332)

2.2.2.19 DM_VARIABLE_DATA (RT Professional)

Deklaration

```
typedef struct {
    DM_TYPEREF          dmTypeRef;
    DM_VARLIMIT         dmVarLimit;
    VARIANT             dmStart;
    VARIANT             dmDefault;
    DWORD               dwNotify;
    DWORD               dwFlags;
    CHAR                szSpecific[MAX_DM_VAR_SPECIFIC +1];
    CHAR                szGroup[MAX_DM_GROUP_NAME +1];
    CHAR                szConnection[
        MAX_DM_CONNECTION_NAME +1];
    CHAR                szChannel[_MAX_PATH +1];
    CHAR                szUnit[MAX_DM_UNIT_NAME +1];
}
DM_VARIABLE_DATA;
```

Members

dmTypeRef

Die Struktur DM_TYPEREF (Seite 227) enthält Angaben zum Variablentyp.

dmVarLimit

Die Struktur DM_VARLIMIT (Seite 244) enthält Angaben zu den Grenzwerten einer Variablen

dmStart

Startwert der Variablen. Muss nach Benutzung mit VariantClear(&dmStart) wieder freigegeben werden.

dmDefault

Ersatzwert der Variablen. Muss nach Benutzung mit VariantClear(&dmDefault) wieder freigegeben werden.

dwNotify

In dwNotify wird festgelegt für welche Ereignisse ein Protokolleintrag generiert wird:

DM_NOTIFY_MAX_LIMIT	0x00000001	bei Erreichen der Obergrenze
DM_NOTIFY_MIN_LIMIT	0x00000002	bei Erreichen der Untergrenze
DM_NOTIFY_FORMAT_ERROR	0x00000004	bei Wandlungsfehler
DM_NOTIFY_ACCESS_FAULT	0x00000008	bei unzulässigen Schreibzugriff
DM_NOTIFY_APPLICATION_WRITE	0x00000010	bei Schreibzugriff Applikation
DM_NOTIFY_PROCESS_WRITE	0x00000020	bei Schreibzugriff Prozess

dwFlags

kennzeichnet wie der Ersatzwert verwendet werden soll:

DM_HAS_MIN_LIMIT	0x00000008	Variable hat feste Untergrenze
DM_HAS_MAX_LIMIT	0x00000010	Variable hat feste Obergrenze
DM_HAS_DEFAULT_VALUE	0x00000020	Variable hat Ersatzwert
DM_HAS_STARTUP_VALUE	0x00000040	Variable hat Initialisierungswert
DM_USE_DEFAULT_ON_STARTUP	0x00000080	Ersatzwert beim Systemstart eintragen
DM_USE_DEFAULT_ON_MAX	0x00000100	Ersatzwert beim Überschreiten der Obergrenze eintragen
DM_USE_DEFAULT_ON_MIN	0x00000200	Ersatzwert beim Unterschreiten der Untergrenze eintragen
DM_USE_DEFAULT_ON_COMM_ERROR	0x00000400	Ersatzwert bei Verbindungsfehler

szSpecific

szSpecific enthält die Adressbeziehung der Variablen, z.B. Datenbaustein und Byte im Baustein.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der in den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

szGroup

Name der Gruppe zu der die Variable gehört.
Dieser Wert wird nicht bei Servervariablen auf dem Multi-Client geliefert.

szConnection

Name der logischen Verbindung, die mit der Variablen verknüpft ist.

szChannel

Dateiname des Kanaltreibers.

szUnit

Name der Kanal-Unit, die mit der Variablen verknüpft ist.

Benötigte Dateien

dmclient.h

API - Funktion

DM_ENUM_VARIABLE_PROC (Seite 327)	Informationen zu einer Variablen auflisten (Call-back)
-----------------------------------	--

Siehe auch

DM_TYPEREF (Seite 227)
 DM_ENUM_VARIABLE_PROC (Seite 327)
 DM_VARLIMIT (Seite 244)
 DMEnumVarData (Seite 325)

2.2.2.20 DM_VARIABLE_DATA4 (RT Professional)**Deklaration**

```
typedef struct {
    DM_TYPEREF          dmTypeRef;
    DM_VARLIMIT        dmVarLimit;
    VARIANT            dmStart;
    VARIANT            dmDefault;
    DWORD              dwNotify;
    DWORD              dwFlags;
    CHAR               szSpecific[MAX_DM_VAR_SPECIFIC +1];
    CHAR               szGroup[MAX_DM_GROUP_NAME +1];
    CHAR               szConnection[
        MAX_DM_CONNECTION_NAME +1];
    CHAR               szChannel[_MAX_PATH +1];
    CHAR               szUnit[MAX_DM_UNIT_NAME +1];
    MCP_VARIABLE_SCALES Scaling;
    DWORD              dwASDataSize;
    DWORD              dwOSDataSize;
    DWORD              dwVarProperty;
    DWORD              dwFormat;
}
DM_VARIABLE_DATA4;
```

Members**dmTypeRef**

Die Struktur DM_TYPEREF (Seite 227) enthält Angaben zum Variablentyp.

dmVarLimit

Die Struktur DM_VARLIMIT (Seite 244) enthält Angaben zu den Grenzwerten einer Variablen

dmStart

Startwert der Variablen. Muss nach Benutzung mit VariantClear(&dmStart) wieder freigegeben werden.

dmDefault

Ersatzwert der Variablen. Muss nach Benutzung mit VariantClear(&dmDefault) wieder freigegeben werden.

dwNotify

In dwNotify wird festgelegt für welche Ereignisse ein Protokolleintrag generiert wird:

DM_NOTIFY_MAX_LIMIT	0x00000001	bei Erreichen der Obergrenze
DM_NOTIFY_MIN_LIMIT	0x00000002	bei Erreichen der Untergrenze
DM_NOTIFY_FORMAT_ERROR	0x00000004	bei Wandlungsfehler
DM_NOTIFY_ACCESS_FAULT	0x00000008	bei unzulässigen Schreibzugriff
DM_NOTIFY_APPLICATION_WRITE	0x00000010	bei Schreibzugriff Applikation
DM_NOTIFY_PROCESS_WRITE	0x00000020	bei Schreibzugriff Prozess

dwFlags

Kennzeichnet wie der Ersatzwert verwendet werden soll:

DM_HAS_MIN_LIMIT	0x00000008	Variable hat feste Untergrenze
DM_HAS_MAX_LIMIT	0x00000010	Variable hat feste Obergrenze
DM_HAS_DEFAULT_VALUE	0x00000020	Variable hat Ersatzwert
DM_HAS_STARTUP_VALUE	0x00000040	Variable hat Initialisierungswert
DM_USE_DEFAULT_ON_STARTUP	0x00000080	Ersatzwert beim Systemstart eintragen
DM_USE_DEFAULT_ON_MAX	0x00000100	Ersatzwert beim Überschreiten der Obergrenze eintragen
DM_USE_DEFAULT_ON_MIN	0x00000200	Ersatzwert beim Unterschreiten der Untergrenze eintragen
DM_USE_DEFAULT_ON_COMM_ERROR	0x00000400	Ersatzwert bei Verbindungsfehler

szSpecific

szSpecific enthält die Adressbeziehung der Variablen, z.B. Datenbaustein und Byte im Baustein.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der bei den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

szGroup

Name der Gruppe zu der die Variable gehört.

szConnection

Name der logischen Verbindung, die mit der Variablen verknüpft ist.

szChannel

Dateiname des Kanaltreibers.

szUnit

Name der Kanal-Unit, die mit der Variablen verknüpft ist.

Scaling

Struktur MCP_VARIABLE_SCALES (Seite 268) mit der Beschreibung der Variablenskalierung.

dwASDataSize

Länge der Variablen in der PLC (Anzahl Bits)

dwOSDataSize

Länge der Variablen in der OS (Anzahl Bits)

dwVarProperty

Kennzeichnet ob es sich um eine interne oder eine externe Variable handelt:

DM_INTERNAL_VAR	0x00000002	Interne Variable
DM_EXTERNAL_VAR	0x00000004	externe Variable

dwFormat

Nummer der verwendeten Konvertierungsroutine. Näheres finden Sie im Abschnitt "Konvertierungsroutinen".

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_VARIABLE_PROC4 (Seite 331)	Informationen zu einer Variablen auflisten (Call-back)
------------------------------------	--

Siehe auch

DMEnumVarData4 (Seite 329)

DM_ENUM_VARIABLE_PROC4 (Seite 331)

DM_TYPEREF (Seite 227)

DM_VARLIMIT (Seite 244)

MCP_VARIABLE_SCALES (Seite 268)

2.2.2.21 DM_VARKEY (RT Professional)**Deklaration**

```
typedef struct {
    DWORD    dwKeyType;
    DWORD    dwID;
    char     szName[ MAX_DM_VAR_NAME + 1 ];
    LPVOID   lpvUserData;
}
DM_VARKEY;
```

Members

dwKeyType

dwKeyType gibt an, ob die Variable über eine Schlüssel-ID oder über Ihren Namen angesprochen werden soll:

DM_VARKEY_ID	Spezifizierung über Schlüssel ID
DM_VARKEY_NAME	Spezifizierung über Variablennamen

Wird DM_VARKEY als Rückgabe-Struktur verwendet, können beide Typen besetzt sein, wenn die Schlüssel-ID und der Variablenname zurückgegeben werden.

Wird DM_VARKEY als Versorgungsparameter verwendet, sollte bevorzugt der Variablenname besetzt werden, da nur hier ein Serverpräfix mit angegeben werden kann. Wenn die Schlüssel-ID verwendet wird, so wird immer lokal gearbeitet.

dwID

Enthält, wenn dwKeyType entsprechend gesetzt ist, die Schlüssel-ID der Variablen.

szName

Enthält, wenn dwKeyType entsprechend gesetzt ist, den Namen der Variablen.

lpvUserData

Zeiger auf applikationsspezifische Daten.

Bemerkungen

Die Struktur DM_VARKEY ist Teil der Struktur DM_VAR_UPDATE_STRUCT (Seite 228).

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_TYPEMEMBERS_PROC_EX (Seite 435)	Variablen einer strukturierten Variablen auflisten (Callback)
DM_ENUM_TYPEMEMBERS_PROC (Seite 430)	Variablen einer strukturierten Variablen auflisten (Callback)
DM_ENUM_VAR_PROC (Seite 339)	Variablennamen auflisten (Callback)
DM_ENUM_VARIABLE_PROC (Seite 327)	Informationen zu einer Variablen auflisten (Callback)
DM_NOTIFY_SELECT_VAR_PROC (Seite 421)	Variablenauswahl-dialog öffnen (Callback)
DMEnumVarData (Seite 325)	Informationen zu einer Variablen auflisten
DMGetValue (Seite 341)	Variablenwerte ermitteln
DMGetValueWait (Seite 353)	Variablenwerte mit Aktualisierung ermitteln
DMGetVarLimits (Seite 372)	Grenzwerte einer Variablen ermitteln
DMGetVarInfo (Seite 360)	Variablen-ID ermitteln, Variablenname ermitteln

DMGetVarType (Seite 378)	Datentyp einer Variablen ermitteln
DMSetValue (Seite 385)	Variablenwerte ändern
DMSetValueMessage (Seite 391)	Variablenwerte mit Meldungsausgabe ändern
DMSetValueWait (Seite 394)	Variablenwerte mit Benachrichtigung ändern
DMSetValueWaitMessage (Seite 400)	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern
DMShowVarDatabase (Seite 410)	Variablen-Auswahldialog öffnen
DMStartVarUpdate (Seite 456)	Aktualisierungsvariablen festlegen

Siehe auch

DMStartVarUpdate (Seite 456)
 DM_VAR_UPDATE_STRUCT (Seite 228)
 DMGetValue (Seite 341)
 DMGetValueWait (Seite 353)
 DMGetVarInfo (Seite 360)
 DMGetVarLimits (Seite 372)
 DMGetVarType (Seite 378)
 DMSetValue (Seite 385)
 DMSetValueMessage (Seite 391)
 DMSetValueWait (Seite 394)
 DMSetValueWaitMessage (Seite 400)
 DMShowVarDatabase (Seite 410)
 DMEnumVarData (Seite 325)
 DM_ENUM_VAR_PROC (Seite 339)
 DM_ENUM_VARIABLE_PROC (Seite 327)
 DM_ENUM_TPEMEMBERS_PROC (Seite 430)
 DM_ENUM_TPEMEMBERS_PROC_EX (Seite 435)
 DM_NOTIFY_SELECT_VAR_PROC (Seite 421)
 DMStartVarUpdateEx (Seite 460)
 DM_VAR_UPDATE_STRUCTEX (Seite 230)
 DM_ENUM_VARIABLE_PROC4 (Seite 331)
 DMGetValueEx (Seite 343)
 DMGetValueWaitEx (Seite 356)
 DM_ENUM_TPEMEMBERS_PROC_EX4 (Seite 438)
 MSG_CSDATA_STRUCT_PLUS (Seite 1037)

2.2.2.22 DM_VARLIMIT (RT Professional)

Deklaration

```
typedef struct {  
    VARIANT    dmMaxRange;  
    VARIANT    dmMinRange;  
    VARIANT    dmMaxLimit;  
    VARIANT    dmMinLimit;  
}  
DM_VARLIMIT;
```

Members

- dmMaxRange**
Oberer Grenzwert der Formatkonvertierung
- dmMinRange**
Unterer Grenzwert der Formatkonvertierung
- dmMaxLimit**
Obergrenze der Variablen.
- dmMinLimit**
Untergrenze der Variablen.

Bemerkungen

DM_VARLIMIT ist Teil der Struktur DM_VARIABLE_DATA (Seite 237). Alle VARIANT sollten nach Benutzung mit VariantClear(&dmxxx) wieder freigegeben werden.

Benötigte Dateien

dmclient.h

API - Funktionen

DMGetVarLimits (Seite 372)	Grenzwerte einer Variablen ermitteln
----------------------------	--------------------------------------

Siehe auch

- DM_VARIABLE_DATA (Seite 237)
- DMGetVarLimits (Seite 372)
- DM_VARIABLE_DATA4 (Seite 239)

2.2.2.23 MCP_NEWVARIABLE_DATA (RT Professional)

Deklaration

```
typedef struct {
    DWORD                dwFlags;
    char                 szProjectFile[_MAX_PATH +1];
    char                 szConnection[
        MAX_DM_CONNECTION_NAME +3];
    char                 szVarName[MAX_DM_VAR_NAME +1];
    char                 szGroupName[MAX_DM_GROUP_NAME +1];
    MCP_VARIABLE_COMMON Common;
    MCP_VARIABLE_PROTOCOL Protocol;
    MCP_VARIABLE_LIMITS Limits;
    char                 szSpecific[MAX_DM_VAR_SPECIFIC +1];
}
MCP_NEWVARIABLE_DATA;
```

Members

dwFlags

dwFlags kennzeichnet, wie die Variable bearbeitet werden soll:

MCP_NVAR_FLAG_CREATE	1	Variable erzeugen
MCP_NVAR_FLAG_MODIFY	2	Variable verändern(darf nicht in RT verwendet werden)
MCP_NVAR_FLAG_TEST	3	die Existenz einer Variablen überprüfen

Die Flags dürfen nicht verodert werden.

szProjectFile

Name der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei kann mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt werden.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler DM_E_NOT_CONNECTED abgewiesen.

szConnection

Name der zur Variablen zugeordneten logischen Verbindung.

szVarName

Name der neu zu bearbeitenden Variablen.

szGroupName

Name der Gruppe, der die Variable angehört.

Wird ein Gruppenname angegeben, der in einer anderen Verbindung existiert, als in szConnection angegeben, so wird der Gruppenname ohne Fehlermeldung ignoriert und die Variable direkt in der Verbindung angelegt.

Wird ein Gruppenname angegeben, der noch nicht existiert, so wird die Gruppe implizit angelegt.

Beim Verändern einer Variablen mit MCP_NVAR_FLAG_MODIFY wird eine Änderung des Gruppennamens ohne Fehlermeldung ignoriert.

Common

Struktur MCP_VARIABLE_COMMON (Seite 256) mit der Beschreibung der Variablen.

Protocol

Struktur MCP_VARIABLE_PROTOCOL (Seite 266) mit der Beschreibung der Behandlung von Grenzwertverletzungen durch die Variable.

Limits

Struktur MCP_VARIABLE_LIMITS (Seite 260) mit mit den Grenzwerten der Variablen.

szSpecific

szSpecific enthält die Adressbeziehung der Variablen, z.B. Datenbaustein und Byte im Baustein.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der bei den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

Bemerkungen

Die Struktur MCP_NEWVARIABLE_DATA_EX (Seite 252) besitzt bei erweiterter Funktionalität eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

API - Funktionen

GAPICreateNewVariable (Seite 422)	Variable erzeugen
-----------------------------------	-------------------

Siehe auch

GAPICreateNewVariable (Seite 422)

MCP_VARIABLE_PROTOCOL (Seite 266)

MCP_VARIABLE_COMMON (Seite 256)

MCP_VARIABLE_LIMITS (Seite 260)

MCP_NEWVARIABLE_DATA_EX (Seite 252)

2.2.2.24 MCP_NEWVARIABLE_DATA_4 (RT Professional)

Deklaration

```

typedef struct {
    DWORD                dwFlags;
    char                 szProjectFile[_MAX_PATH +1];
    char                 szConnection[
        MAX_DM_CONNECTION_NAME +3];
    char                 szVarName[MAX_DM_VAR_NAME +1];
    char                 szGroupName[MAX_DM_GROUP_NAME +1];
    MCP_VARIABLE_COMMON Common;
    MCP_VARIABLE_PROTOCOL Protocol;
    MCP_VARIABLE_LIMITS Limits;
    char                 szSpecific[MAX_DM_VAR_SPECIFIC +1];
    MCP_VARIABLE_SCALES Scaling;
}
MCP_NEWVARIABLE_DATA_4;

```

Members

dwFlags

dwFlags kennzeichnet, wie die Variable bearbeitet werden soll:

MCP_NVAR_FLAG_CREATE	1	Variable erzeugen
MCP_NVAR_FLAG_MODIFY	2	Variable verändern (falls nicht vorhanden, dann neu erzeugen) (darf nicht in RT verwendet werden)
MCP_NVAR_FLAG_TEST	3	die Existenz einer Variablen überprüfen

Die Flags dürfen nicht verodert werden.

szProjectFile

Name der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei kann mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt werden.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler (DM_E_NOT_CONNECTED) abgewiesen.

szConnection

Name der zur Variablen zugeordneten logischen Verbindung.

szVarName

Name der neu zu bearbeitenden Variablen.

szGroupName

Name der Gruppe, der die Variable angehört.

Wird ein Gruppenname angegeben, der in einer anderen Verbindung existiert, als in szConnection angegeben, so wird der Gruppenname ohne Fehlermeldung ignoriert und die Variable direkt in der Verbindung angelegt.

Wird ein Gruppenname angegeben, der noch nicht existiert, so wird die Gruppe implizit angelegt.

Beim Verändern einer Variablen mit MCP_NVAR_FLAG_MODIFY wird eine Änderung des Gruppennamens ohne Fehlermeldung ignoriert.

Common

Struktur MCP_VARIABLE_COMMON (Seite 256) mit der Beschreibung der Variablen.

Protocol

Struktur MCP_VARIABLE_PROTOCOL (Seite 266) mit der Beschreibung der Behandlung von Grenzwertverletzungen durch die Variable.

Limits

Struktur MCP_VARIABLE_LIMITS (Seite 260) mit den Grenzwerten der Variablen.

szSpecific

szSpecific enthält die Adressbeziehung der Variablen, z.B. Datenbaustein und Byte im Baustein.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der bei den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

Scaling

Struktur MCP_VARIABLE_SCALES (Seite 268) mit der Beschreibung der Variablenskalierung.

Bemerkungen

Die Struktur MCP_NEWVARIABLE_DATA_EX4 (Seite 254) besitzt bei erweiterter Funktionalität eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

API - Funktionen

GAPICreateNewVariable4 (Seite 424)	Variable erzeugen
GAPICreateNewVariableEx4 (Seite 427)	Variable erzeugen mit Creator-ID

Siehe auch

MCP_VARIABLE_COMMON (Seite 256)
 MCP_VARIABLE_LIMITS (Seite 260)
 MCP_VARIABLE_PROTOCOL (Seite 266)
 GAPICreateNewVariable4 (Seite 424)
 MCP_VARIABLE_SCALES (Seite 268)
 MCP_NEWVARIABLE_DATA_EX4 (Seite 254)
 GAPICreateNewVariableEx4 (Seite 427)

2.2.2.25 MCP_NEWVARIABLE_DATA_5 (RT Professional)

Deklaration

```

typedef struct {
    DWORD                dwFlags;
    char                 szProjectFile[_MAX_PATH +1];
    char                 szConnection[
        MAX_DM_CONNECTION_NAME +3];
    char                 szVarName[MAX_DM_VAR_NAME +1];
    char                 szGroupName[MAX_DM_GROUP_NAME +1];
    MCP_VARIABLE_COMMON Common;
    MCP_VARIABLE_PROTOCOL Protocol;
    MCP_VARIABLE_LIMITS5 Limits;
    char                 szSpecific[MAX_DM_VAR_SPECIFIC +1];
    MCP_VARIABLE_SCALES Scaling;
}
MCP_NEWVARIABLE_DATA_5;
  
```

Members

dwFlags

dwFlags kennzeichnet, wie die Variable bearbeitet werden soll:

MCP_NVAR_FLAG_CREATE	1	Variable erzeugen
MCP_NVAR_FLAG_MODIFY	2	Variable verändern (falls nicht vorhanden, dann neu erzeugen) (darf nicht in RT verwendet werden)
MCP_NVAR_FLAG_TEST	3	die Existenz einer Variablen überprüfen

Die Flags dürfen nicht verodert werden.

szProjectFile

Name der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei kann mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt werden.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler (DM_E_NOT_CONNECTED) abgewiesen.

szConnection

Name der zur Variablen zugeordneten logischen Verbindung.

szVarName

Name der neu zu bearbeitenden Variablen.

szGroupName

Name der Gruppe, der die Variable angehört.

Wird ein Gruppenname angegeben, der in einer anderen Verbindung existiert, als in szConnection angegeben, so wird der Gruppenname ohne Fehlermeldung ignoriert und die Variable direkt in der Verbindung angelegt.

Wird ein Gruppenname angegeben, der noch nicht existiert, so wird die Gruppe implizit angelegt.

Beim Verändern einer Variablen mit MCP_NVAR_FLAG_MODIFY wird eine Änderung des Gruppennamens ohne Fehlermeldung ignoriert.

Common

Struktur MCP_VARIABLE_COMMON (Seite 256) mit der Beschreibung der Variablen.

Protocol

Struktur MCP_VARIABLE_PROTOCOL (Seite 266) mit der Beschreibung der Behandlung von Grenzwertverletzungen durch die Variable.

Limits

Struktur MCP_VARIABLE_LIMITS5 (Seite 262) mit den Grenzwerten der Variablen.

szSpecific

szSpecific enthält die Adressbeziehung der Variablen, z.B. Datenbaustein und Byte im Baustein.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der bei den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

Scaling

Struktur MCP_VARIABLE_SCALES (Seite 268) mit der Beschreibung der Variablenskalierung.

Bemerkungen

Die Struktur MCP_NEWVARIABLE_DATA_EX4 (Seite 254) besitzt bei erweiterter Funktionalität eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

API - Funktionen

GAPICreateNewVariable5 (Seite 425)	Variable erzeugen mit Creator-ID
------------------------------------	----------------------------------

Siehe auch

GAPICreateNewVariable5 (Seite 425)
MCP_VARIABLE_COMMON (Seite 256)
MCP_VARIABLE_LIMITS5 (Seite 262)
MCP_VARIABLE_PROTOCOL (Seite 266)
MCP_NEWVARIABLE_DATA_EX4 (Seite 254)
MCP_VARIABLE_SCALES (Seite 268)

2.2.2.26 MCP_NEWVARIABLE_DATA_EX (RT Professional)

Deklaration

```
typedef struct {
    DWORD dwFlags;
    char szProjectFile[_MAX_PATH +1];
    char szConnection[
        MAX_DM_CONNECTION_NAME +3];
    char szVarName[MAX_DM_VAR_NAME +1];
    char szGroupName[MAX_DM_GROUP_NAME +1];
    MCP_VARIABLE_COMMON_EX Common;
    MCP_VARIABLE_PROTOCOL_EX Protocol;
    MCP_VARIABLE_LIMITS_EX Limits;
    char szSpecific[MAX_DM_VAR_SPECIFIC +1]
}
MCP_NEWVARIABLE_DATA_EX;
```

Members

dwFlags

dwFlags kennzeichnet, wie die Variable bearbeitet werden soll:

MCP_NVAR_FLAG_CREATE	1	Variable erzeugen
MCP_NVAR_FLAG_MODIFY	2	Variable verändern(darf nicht in Runtime verwendet werden)
MCP_NVAR_FLAG_TEST	3	überprüft die Existenz einer Variablen.

Die Flags dürfen nicht verodert werden.

szProjectFile

Name der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei kann mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt werden.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler (DM_E_NOT_CONNECTED) abgewiesen.

szConnection

Name der zur Variablen zugeordneten logischen Verbindung.

szVarName

Name der neu zu bearbeitenden Variablen.

szGroupName

Name der Gruppe, der die Variable angehört.

Wird ein Gruppenname angegeben, der in einer anderen Verbindung existiert, als in szConnection angegeben, so wird der Gruppenname ohne Fehlermeldung ignoriert und die Variable direkt in der Verbindung angelegt.

Wird ein Gruppenname angegeben, der noch nicht existiert, so wird die Gruppe implizit angelegt.

Beim Verändern einer Variablen mit MCP_NVAR_FLAG_MODIFY wird eine Änderung des Gruppennamens ohne Fehlermeldung ignoriert.

Common

Struktur MCP_VARIABLE_COMMON_EX (Seite 258) mit der Beschreibung der Variablen.

Protocol

Struktur MCP_VARIABLE_PROTOCOL_EX (Seite 267) mit der Beschreibung der Behandlung von Grenzwertverletzungen durch die Variable.

Limits

Struktur MCP_VARIABLE_LIMITS_EX (Seite 264) mit den Grenzwerten der Variablen.

szSpecific

szSpecific enthält die Adressbeziehung der Variablen, z.B. Datenbaustein und Byte im Baustein.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der bei den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

Bemerkungen

Die Struktur MCP_NEWVARIABLE_DATA (Seite 245) besitzt eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

API - Funktion

DM_ENUM_TPEMEMBERS_PROC_EX (Seite 435)	Variablen einer strukturierten Variablen auflisten (Callback)
--	---

Siehe auch

MCP_NEWVARIABLE_DATA (Seite 245)

DM_ENUM_TPEMEMBERS_PROC_EX (Seite 435)

MCP_VARIABLE_COMMON_EX (Seite 258)

MCP_VARIABLE_LIMITS_EX (Seite 264)

MCP_VARIABLE_PROTOCOL_EX (Seite 267)

2.2.2.27 MCP_NEWVARIABLE_DATA_EX4 (RT Professional)

Deklaration

```
typedef struct {
    DWORD dwFlags;
    char szProjectFile[_MAX_PATH +1];
    char szConnection[
        MAX_DM_CONNECTION_NAME +3];
    char szVarName[MAX_DM_VAR_NAME +1];
    char szGroupName[MAX_DM_GROUP_NAME +1];
    MCP_VARIABLE_COMMON_EX Common;
    MCP_VARIABLE_PROTOCOL_EX Protocol;
    MCP_VARIABLE_LIMITS_EX Limits;
    char szSpecific[MAX_DM_VAR_SPECIFIC +1]
    MCP_VARIABLE_SCALES Scaling;
}
MCP_NEWVARIABLE_DATA_EX4;
```

Members

dwFlags

dwFlags kennzeichnet, wie die Variable bearbeitet werden soll:

MCP_NVAR_FLAG_CREATE	1	Variable erzeugen
MCP_NVAR_FLAG_MODIFY	2	Variable verändern(darf nicht in RT verwendet werden)
MCP_NVAR_FLAG_TEST	3	überprüft die Existenz einer Variablen.

Die Flags dürfen nicht verändert werden.

szProjectFile

Name der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei kann mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt werden.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler (DM_E_NOT_CONNECTED) abgewiesen.

szConnection

Name der zur Variablen zugeordneten logischen Verbindung.

szVarName

Name der neu zu bearbeitenden Variablen.

szGroupName

Name der Gruppe, der die Variable angehört.

Wird ein Gruppenname angegeben, der in einer anderen Verbindung existiert, als in szConnection angegeben, so wird der Gruppenname ohne Fehlermeldung ignoriert und die Variable direkt in der Verbindung angelegt.

Wird ein Gruppenname angegeben, der noch nicht existiert, so wird die Gruppe implizit angelegt.

Beim Verändern einer Variablen mit MCP_NVAR_FLAG_MODIFY wird eine Änderung des Gruppennamens ohne Fehlermeldung ignoriert.

Common

Struktur MCP_VARIABLE_COMMON_EX (Seite 258) mit der Beschreibung der Variablen.

Protocol

Struktur MCP_VARIABLE_PROTOCOL_EX (Seite 267) mit der Beschreibung der Behandlung von Grenzwertverletzungen durch die Variable.

Limits

Struktur MCP_VARIABLE_LIMITS_EX (Seite 264) mit den Grenzwerten der Variablen.

szSpecific

szSpecific enthält die Adressbeziehung der Variablen, z.B. Datenbaustein und Byte im Baustein.... PLC-spezifische Details entnehmen Sie bitte dem Kommunikationshandbuch.

Es handelt sich um den gleichen Wert, der bei den Eigenschaften einer Variablen in WinCC in der Spalte Parameter angezeigt wird.

Scaling

Struktur MCP_VARIABLE_SCALES (Seite 268) mit der Beschreibung der Variablenskalierung.

Bemerkungen

Die Struktur MCP_NEWVARIABLE_DATA_4 (Seite 247) besitzt eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

API - Funktionen

DM_ENUM_TPEMEMBERS_PROC_EX4 (Seite 438)	Variablen einer strukturierten Variablen auflisten (Callback)
---	---

Siehe auch

MCP_NEWVARIABLE_DATA_4 (Seite 247)

MCP_NEWVARIABLE_DATA_5 (Seite 249)

- DM_ENUM_TPEMEMBERS_PROC_EX4 (Seite 438)
- MCP_VARIABLE_COMMON_EX (Seite 258)
- MCP_VARIABLE_LIMITS_EX (Seite 264)
- MCP_VARIABLE_PROTOCOL_EX (Seite 267)
- MCP_VARIABLE_SCALES (Seite 268)

2.2.2.28 MCP_VARIABLE_COMMON (RT Professional)

Deklaration

```
typedef struct {
    DWORD    dwVarType;
    DWORD    dwVarLength;
    DWORD    dwVarProperty;
    DWORD    dwFormat;
}
MCP_VARIABLE_COMMON;
```

Members

dwVarType

In dwVarType wird der Typ der Variablen spezifiziert:

DM_VARTYPE_BIT	Binäre Variable
DM_VARTYPE_SBYTE	Vorzeichenbehafteter 8-Bit Wert
DM_VARTYPE_BYTE	Vorzeichenloser 8-Bit Wert
DM_VARTYPE_SWORD	Vorzeichenbehafteter 16-Bit Wert
DM_VARTYPE_WORD	Vorzeichenloser 16-Bit Wert
DM_VARTYPE_SDWORD	Vorzeichenbehafteter 32-Bit Wert
DM_VARTYPE_DWORD	Vorzeichenloser 32-Bit Wert
DM_VARTYPE_FLOAT	Gleitkommazahl 32-Bit IEEE 754
DM_VARTYPE_DOUBLE	Gleitkommazahl 64-Bit IEEE 754
DM_VARTYPE_TEXT_8	Textvariable 8-Bit Zeichensatz
DM_VARTYPE_TEXT_16	Textvariable 16-Bit Zeichensatz
DM_VARTYPE_RAW	Rohdatentyp
DM_VARTYPE_STRUCT	Strukturvariable
DM_VARTYPE_TEXTREF	Text-Referenz aus Textbib

dwVarLength

Die Angabe der Variablenlänge ist nur bei den Textvariablen DM_VARTYPE_TEXT_8 und DM_VARTYPE_TEXT_16 von Interesse.

Dabei wird die Textlänge in Zeichen (1 255) angegeben.

dwVarProperty

Kennzeichnet ob es sich um eine interne oder eine externe Variable handelt:

DM_INTERNAL_VAR	interne Variable
DM_EXTERNAL_VAR	externe Variable

dwFormat

Nummer der zu verwendenden Konvertierungsroutine. Näheres finden Sie im Abschnitt "Konvertierungsroutinen".

Bemerkungen

MCP_VARIABLE_COMMON ist Teil der Strukturen MCP_NEWVARIABLE_DATA (Seite 245) und MCP_NEWVARIABLE_DATA_4 (Seite 247).

Die Struktur MCP_VARIABLE_COMMON_EX (Seite 258) besitzt bei erweiterter Funktionalität eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

Siehe auch

MCP_NEWVARIABLE_DATA (Seite 245)

MCP_NEWVARIABLE_DATA_4 (Seite 247)

MCP_VARIABLE_COMMON_EX (Seite 258)

MCP_NEWVARIABLE_DATA_5 (Seite 249)

2.2.2.29 MCP_VARIABLE_COMMON_EX (RT Professional)**Deklaration**

```
typedef struct {
    DWORD    dwVarType;
    DWORD    dwCreatorID;
    DWORD    dwVarLength;
    DWORD    dwVarProperty;
    DWORD    dwFormat;
    DWORD    dwOSOffset;
    DWORD    dwASOffset;
    char     szStructTypeName
}
MCP_VARIABLE_COMMON_EX;
```

Members

dwVarType

In dwVarType wird der Typ der Variablen spezifiziert:

DM_VARTYPE_BIT	Binäre Variable
DM_VARTYPE_SBYTE	Vorzeichenbehafteter 8-Bit Wert
DM_VARTYPE_BYTE	Vorzeichenloser 8-Bit Wert
DM_VARTYPE_SWORD	Vorzeichenbehafteter 16-Bit Wert
DM_VARTYPE_WORD	Vorzeichenloser 16-Bit Wert
DM_VARTYPE_SDWORD	Vorzeichenbehafteter 32-Bit Wert
DM_VARTYPE_DWORD	Vorzeichenloser 32-Bit Wert
DM_VARTYPE_FLOAT	Gleitkommazahl 32-Bit IEEE 754
DM_VARTYPE_DOUBLE	Gleitkommazahl 64-Bit IEEE 754
DM_VARTYPE_TEXT_8	Textvariable 8-Bit Zeichensatz
DM_VARTYPE_TEXT_16	Textvariable 16-Bit Zeichensatz
DM_VARTYPE_RAW	Rohdatentyp
DM_VARTYPE_STRUCT	Strukturvariable
DM_VARTYPE_TEXTREF	Text-Referenz aus Textbib

dwCreatorID

In der Erzeugerkennung können Sie Angaben treffen, von wem ein Objekt angelegt wurde.

Die Werte 0 – 10100 sowie 11000 - 11100 sind intern oder für bestimmte Systeme reserviert.

dwVarLength

Die Angabe der Variablenlänge ist nur bei den Textvariablen DM_VARTYPE_TEXT_8 und DM_VARTYPE_TEXT_16 von Interesse. Dabei wird die Textlänge in Zeichen (1 255) angegeben.

dwVarProperty

Kennzeichnet ob es sich um eine interne oder eine externe Variable handelt:

DM_INTERNAL_VAR	interne Variable
DM_EXTERNAL_VAR	externe Variable

dwFormat

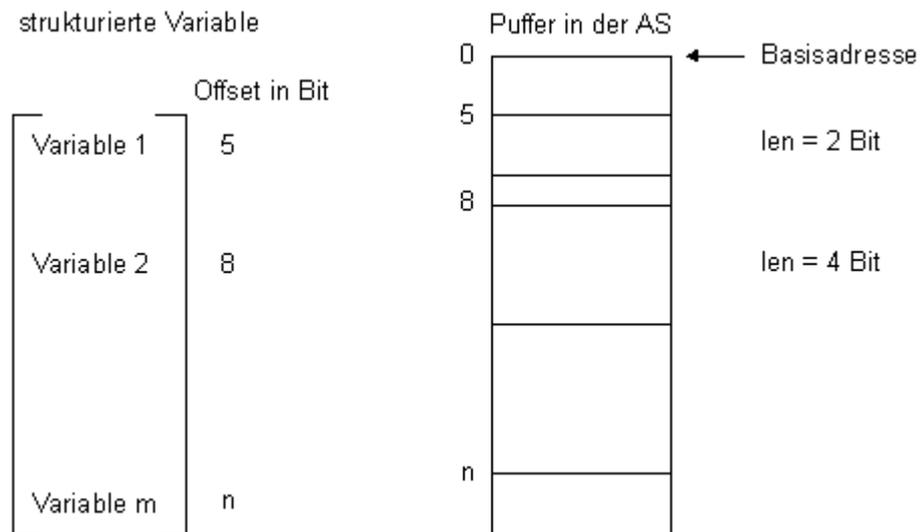
Nummer der zu verwendenden Konvertierungsroutine. Näheres finden Sie im Abschnitt "Konvertierungsroutinen".

dwOSOffset

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

dwASOffset

Offset im Puffer der PLC



Der Bit-orientierte ASOffset wird intern in BYTE gehandhabt.

szStructTypeName

Name des Strukturtyps

Bemerkungen

MCP_VARIABLE_COMMON_EX ist Teil der Struktur MCP_NEWVARIABLE_DATA_EX (Seite 252).

Die Struktur MCP_VARIABLE_COMMON (Seite 256) besitzt ähnliche Verwendung.

Benötigte Dateien

dmclient.h

Siehe auch

MCP_NEWVARIABLE_DATA (Seite 245)

MCP_VARIABLE_COMMON (Seite 256)

MCP_NEWVARIABLE_DATA_EX (Seite 252)

MCP_NEWVARIABLE_DATA_EX4 (Seite 254)

2.2.2.30 MCP_VARIABLE_LIMITS (RT Professional)

Deklaration

```
typedef struct {  
    double    dTopLimit;  
    double    dBottomLimit;  
    double    dStartValue;  
    double    dSubstituteValue;  
    BOOL      bTopLimit;  
    BOOL      bBottomLimit;  
    BOOL      bStartValue;  
    BOOL      bConnectionErr;  
    BOOL      bTopLimitValid;  
    BOOL      bBottomLimitValid;  
    BOOL      bStartValueValid;  
    BOOL      bSubstValueValid;  
}  
MCP_VARIABLE_LIMITS;
```

Members

dTopLimit

Wert der Obergrenze der Variablen

dBottomLimit

Wert der Untergrenze der Variablen

dStartValue

Startwert der Variablen

dSubstituteValue

Ersatzwert der Variablen

bTopLimit

Ist dieser Parameter gesetzt, so soll der Ersatzwert verwendet werden, wenn der Wert der Variablen den in dTopLimit angegebenen Wert übersteigt.

bBottomLimit

Ist dieser Parameter gesetzt, so soll der Ersatzwert verwendet werden, wenn der Wert der Variablen den in dBottomLimit angegebenen Wert unterschreitet.

bStartValue

Ist dieser Parameter gesetzt, so soll der Ersatzwert als Startwert verwendet werden.

bConnectionErr

Ist dieser Parameter gesetzt, so soll der Ersatzwert beim Auftreten eines Verbindungsfehlers verwendet werden.

bTopLimitValid

Ist dieser Parameter gesetzt, so ist der in dTopLimit angegebene Wert der Obergrenze gültig.

bBottomLimitValid

Ist dieser Parameter gesetzt, so ist der in dBottomLimit angegebene Wert der Untergrenze gültig.

bStartValueValid

Ist dieser Parameter gesetzt, so ist der in dStartValue angegebene Wert für den Startwert gültig.

bSubstValueValid

Ist dieser Parameter gesetzt, so ist der in dSubstitudeValue angegebene Wert für den Ersatzwert gültig.

Bemerkungen

MCP_VARIABLE_Limits ist Teil der Strukturen MCP_NEWVARIABLE_DATA (Seite 245) und MCP_NEWVARIABLE_DATA_4 (Seite 247).

Die Struktur MCP_VARIABLE_LIMITS_EX (Seite 264) besitzt bei erweiterter Funktionalität eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

Siehe auch

MCP_NEWVARIABLE_DATA (Seite 245)

MCP_NEWVARIABLE_DATA_4 (Seite 247)

MCP_VARIABLE_LIMITS_EX (Seite 264)

2.2.2.31 MCP_VARIABLE_LIMITS5 (RT Professional)

Deklaration

```
typedef struct {  
    VARIANT    varTopLimit;  
    VARIANT    varBottomLimit;  
    VARIANT    varStartValue;  
    VARIANT    varSubstituteValue;  
    BOOL       bTopLimit;  
    BOOL       bBottomLimit;  
    BOOL       bStartValue;  
    BOOL       bConnectionErr;  
    BOOL       bTopLimitValid;  
    BOOL       bBottomLimitValid;  
    BOOL       bStartValueValid;  
    BOOL       bSubstValueValid;  
}  
MCP_VARIABLE_LIMITS5;
```

Members

varTopLimit

Wert der Obergrenze der Variablen

varBottomLimit

Wert der Untergrenze der Variablen

varStartValue

Startwert der Variablen

varSubstituteValue

Ersatzwert der Variablen

bTopLimit

Ist dieser Parameter gesetzt, so soll der Ersatzwert verwendet werden, wenn der Wert der Variablen den in dTopLimit angegebenen Wert übersteigt.

bBottomLimit

Ist dieser Parameter gesetzt, so soll der Ersatzwert verwendet werden, wenn der Wert der Variablen den in dBottomLimit angegebenen Wert unterschreitet.

bStartValue

Ist dieser Parameter gesetzt, so soll der Ersatzwert als Startwert verwendet werden.

bConnectionErr

Ist dieser Parameter gesetzt, so soll der Ersatzwert beim Auftreten eines Verbindungsfehlers verwendet werden.

bTopLimitValid

Ist dieser Parameter gesetzt, so ist der in dTopLimit angegebene Wert der Obergrenze gültig.

bBottomLimitValid

Ist dieser Parameter gesetzt, so ist der in dBottomLimit angegebene Wert der Untergrenze gültig.

bStartValueValid

Ist dieser Parameter gesetzt, so ist der in dStartValue angegebene Wert für den Startwert gültig.

bSubstValueValid

Ist dieser Parameter gesetzt, so ist der in dSubstituteValue angegebene Wert für den Ersatzwert gültig.

Bemerkungen

MCP_VARIABLE_LIMITS5 ist Teil der Struktur MCP_NEWVARIABLE_DATA_5 (Seite 249) und wird benötigt um mit der Funktion GAPICreateNewVariable5 (Seite 425) Textvariablen mit Start und Ersatzwerten angeben zu können.

Benötigte Dateien

dmclient.h

Siehe auch

MCP_NEWVARIABLE_DATA_5 (Seite 249)

GAPICreateNewVariable5 (Seite 425)

2.2.2.32 MCP_VARIABLE_LIMITS_EX (RT Professional)

Deklaration

```
typedef struct {
    double    dTopLimit
    double    dBottomLimit;
    double    dStartValue;
    double    dSubstituteValue;
    DWORD    dwLimitFlags
    DWORD    dwTextBibStartText;
    char      szTextStartText[255];
    DWORD    dwTextBibSubstitute;
    char      szTextSubstitute[255]
}
MCP_VARIABLE_LIMITS_EX;
```

Members

dTopLimit

Wert der Obergrenze der Variablen

dBottomLimit

Wert der Untergrenze der Variablen

dStartValue

Startwert der Variablen

dSubstituteValue

Ersatzwert der Variablen

LimitFlags

Der Parameter LimitFlags legt die Gültigkeit der Vorgabewerte und Grenzwerte einer Variablen fest:

MCP_VARLIM_HAS_MIN_LIMIT	Variable hat feste Untergrenze
MCP_VARLIM_HAS_MAX_LIMIT	Variable hat feste Obergrenze
MCP_VARLIM_HAS_DEFAULT_VALUE	Variable hat Ersatzwert
MCP_VARLIM_HAS_STARTUP_VALUE	Variable hat Startwert
MCP_VARLIM_USE_DEFAULT_ON_STARTUP	Der Ersatzwert soll beim Systemstart verwendet werden.
MCP_VARLIM_USE_DEFAULT_ON_MAX	Der Ersatzwert soll bei einer Verletzung der Obergrenze verwendet werden.
MCP_VARLIM_USE_DEFAULT_ON_MIN	Der Ersatzwert soll bei einer Verletzung der Untergrenze verwendet werden.
MCP_VARLIM_USE_DEFAULT_ON_COMM_ERROR	Der Ersatzwert soll beim Auftreten eines Verbindungsfehlers verwendet werden.

dwTextBibStartText

Der Parameter ist nur für Textvariablen von Bedeutung. Soll der als Startwert zu verwendende Text aus den Projekttextrn eingelesen werden, so geben Sie hier die ID des entsprechenden Textes an.

szTextStartText

Der Parameter ist nur für Textvariablen von Bedeutung. Unter szTextStartText geben Sie direkt den als Startwert zu verwendenden Text an.

dwTextBibSubstitute

Der Parameter ist nur für Textvariablen von Bedeutung. Soll der als Ersatzwert zu verwendende Text aus den Projekttextrn eingelesen werden, so geben Sie hier die ID des entsprechenden Textes an.

szTextSubstitute

Der Parameter ist nur für Textvariablen von Bedeutung. Unter szTextSubstitute geben Sie direkt den als Ersatzwert zu verwendenden Text an.

Bemerkungen

MCP_VARIABLE_LIMITS_EX ist Teil der Struktur MCP_NEWVARIABLE_DATA_EX (Seite 252).

Die Struktur MCP_VARIABLE_LIMITS (Seite 260) besitzt eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

Siehe auch

MCP_VARIABLE_LIMITS (Seite 260)

MCP_NEWVARIABLE_DATA_EX (Seite 252)

MCP_NEWVARIABLE_DATA_EX4 (Seite 254)

2.2.2.33 MCP_VARIABLE_PROTOCOL (RT Professional)

Deklaration

```
typedef struct {  
    BOOL    bTopLimitErr;  
    BOOL    bBottomLimitErr;  
    BOOL    bTransformationErr;  
    BOOL    bWriteErr;  
    BOOL    bWriteErrApplication;  
    BOOL    bWriteErrProzess;  
}  
MCP_VARIABLE_PROTOCOL;
```

Members

bTopLimitErr

Ein Protokolleintrag wird generiert, wenn der Wert der Variablen den oberen Grenzwert übersteigt.

bBottomLimitErr

Ein Protokolleintrag wird generiert, wenn der Wert der Variablen den unteren Grenzwert unterschreitet.

bTransformationErr

Ein Protokolleintrag wird beim Auftreten eines Wandlungsfehler erzeugt.

bWriteErr

Ein Protokolleintrag wird bei jedem unzulässigem Schreibzugriff generiert.

bWriteErrApplication

Ein Protokolleintrag wird bei jedem unzulässigem Schreibzugriff durch die Applikation erzeugt.

bWriteErrProzess

Ein Protokolleintrag wird bei jedem unzulässigem Schreibzugriff durch den Prozess erzeugt.

Bemerkungen

MCP_VARIABLE_PROTOCOL ist Teil der Strukturen MCP_NEWVARIABLE_DATA (Seite 245) und MCP_NEWVARIABLE_DATA_4 (Seite 247).

Die Struktur MCP_VARIABLE_PROTOCOL_EX (Seite 267) besitzt bei erweiterter Funktionalität eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

Siehe auch

MCP_NEWVARIABLE_DATA (Seite 245)
 MCP_NEWVARIABLE_DATA_4 (Seite 247)
 MCP_VARIABLE_PROTOCOL_EX (Seite 267)
 MCP_NEWVARIABLE_DATA_5 (Seite 249)

2.2.2.34 MCP_VARIABLE_PROTOCOL_EX (RT Professional)**Deklaration**

```
typedef struct {
    DWORD    dwProtocolFlags
}
MCP_VARIABLE_PROTOCOL_EX;
```

Members**dwProtocolFlags**

Über den Parameter dwProtocolFlags wird festgelegt bei welchem Ereignis ein Protokolleintrag generiert werden soll:

MCP_VARPROT_TOPLIMITERR	Ein Protokolleintrag wird generiert, wenn der Wert der Variablen den oberen Grenzwert übersteigt.
MCP_VARPROT_BOTTEMLIMITERR	Ein Protokolleintrag wird generiert, wenn der Wert der Variablen den unteren Grenzwert unterschreitet.
MCP_VARPROT_TRANSFORMATIONERR	Ein Protokolleintrag wird beim Auftreten eines Wandlungsfehler erzeugt.
MCP_VARPROT_WRITEERR	Ein Protokolleintrag wird bei jedem unzulässigem Schreibzugriff generiert.
MCP_VARPROT_WRITEERRAPPLICATION	Ein Protokolleintrag wird bei jedem unzulässigem Schreibzugriff durch die Applikation erzeugt.
MCP_VARPROT_WRITEERRPROCESS	Ein Protokolleintrag wird bei jedem unzulässigem Schreibzugriff durch den Prozess erzeugt.

Eine logische Verknüpfung dieser Konstanten ist möglich.

Bemerkungen

MCP_VARIABLE_PROTOCOL_EX ist Teil der Struktur MCP_NEWVARIABLE_DATA_EX (Seite 252).

Die Struktur MCP_VARIABLE_PROTOCOL (Seite 266) besitzt eine ähnliche Verwendung.

Benötigte Dateien

dmclient.h

Siehe auch

MCP_VARIABLE_PROTOCOL (Seite 266)

MCP_NEWVARIABLE_DATA_EX (Seite 252)

MCP_NEWVARIABLE_DATA_EX4 (Seite 254)

2.2.2.35 MCP_VARIABLE_SCALES (RT Professional)

Deklaration

```
typedef struct {  
    DWORD    dwVarScaleFlags;  
    double   doMinProc;  
    double   doMaxProc;  
    double   doMinVar;  
    double   doMaxVar;  
}  
MCP_VARIABLE_SCALES;
```

Members

dwVarScaleFlags

DM_VARSCALE_NOSCALE: Keine Skalierung

DM_VARSCALE_LINEAR: Lineare Skalierung

doMinProc

Kleinster Wert der Variablen im Prozess

doMaxProc

Größter Wert der Variablen im Prozess

doMinVar

Kleinster Wert der Variablen in WinCC

doMaxVar

Größter Wert der Variablen in WinCC

Bemerkungen

MCP_VARIABLE_SCALES ist Teil der Struktur MCP_NEWVARIABLE_DATA_4 (Seite 247).

Benötigte Dateien

dmclient.h

Siehe auch

MCP_NEWVARIABLE_DATA_4 (Seite 247)

MCP_NEWVARIABLE_DATA_5 (Seite 249)

DM_VARIABLE_DATA4 (Seite 239)

MCP_NEWVARIABLE_DATA_EX4 (Seite 254)

2.2.3 Allgemeine Funktionen (RT Professional)

2.2.3.1 DMActivateRTProject (RT Professional)

Verwendung

Diese Funktion aktiviert das in WinCC geöffnete Projekt.

Deklaration

```
BOOL DMActivateRTProject (  
    LPCMN_ERROR    lpdmError);
```

Parameter

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Projekt aktiviert.

FALSE

Fehler.

Bemerkung

Diese Funktion wird intern asynchron bearbeitet. Wenn der Auftrag intern ordnungsgemäß weitergegeben wurde, liefert sie immer TRUE. Wenn z. B. der Name des Serverrechners im Projekt falsch angegeben wurde, wird folglich kein Fehler gemeldet.

Ein erfolgreicher Start wird mit DMGetRTProject(..) überprüft.

Wenn eine Prüfung auf korrekten Rechnernamen benötigt wird, benutzen Sie die Funktionen DMGetMachineTable(..) und GetComputername(..).

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Siehe auch

DMGetMachineTable (Seite 301)

2.2.3.2 DMAddNotify (RT Professional)

Deklaration

```
BOOL DMAddNotify (
    DM_NOTIFY_PROC    lpfnNotify,
    LPVOID             lpvUser,
    LPDWORD            lpdwNotifyCookie,
    LPCMN_ERROR        lpdmError);
```

Beschreibung

Anbinden eines weiteren Notify-Funktionsaufruf an "DMClient". Die Notify-Funktionalität entspricht der Funktionalität, die Sie bei "DMConnect" angeben können.

Parameter

lpfnNotify

Zeiger auf weitere Notify-Funktionen.

lpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Notify-Funktion wieder zu Verfügung gestellt.

lpdwNotifyCookie

Zeiger auf ein "DWORD", in dem ein Cookie zur Notify zurückgegeben wird.

Das Cookie wird später bei der Funktion "DMRemoveNotify" wieder benötigt, um die Notify-Bearbeitung wieder zu entfernen.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Zusätzliche Notify-Funktion angebunden.

FALSE

Fehler

Bemerkungen

Die Funktion wird für den Fall benötigt, dass bereits innerhalb des Prozesses ein "DMConnect" durchgeführt wurde, aber kein Zugriff auf die dort angegebene Notify-Funktion möglich ist. Hiermit können Sie eine zusätzliche Notify einbinden, um die betreffenden Notify's mit auszuwerten.

Wenn z. B. eine selbsterstellte DLL mit DMClient-Funktionalität in einer globalen Skript-Funktion verwendet wird, kann dort kein "DMConnect" mehr durchgeführt werden. Ein "DMConnect" ist bereits von Skript durchgeführt und eine Notifizierung ist nicht mehr möglich.

In seltenen Fällen wird bereits die Notify geliefert, bevor der Funktionsaufruf zurückgekehrt ist.

Die hier zusätzlich eingebundenen Notify-Funktionen müssen Sie vor dem "DMDisconnect" wieder mit "DMRemoveNotify" entfernen. Ebenso darf die Notify keine Memberfunktion eines Klassenobjekts sein, das vorzeitig zerstört bzw. gelöscht werden kann. Eventuell könnte auch noch ein Notify-Aufruf nach einem "DMRemoveNotify" anstehen.

Fehlermeldungen

DM_E_NOT_CONNECTED	Noch kein "DMConnect" durchgeführt. Die interne Verwaltung ist noch nicht initialisiert
DM_E_ALREADY_EXIST	Eine "NotifyProc" existiert schon mit derselben Adresse im Prozess
DM_E_MAX_LIMIT	Die maximale Anzahl zusätzlicher Notify-Funktionen ist erreicht (16)
DM_E_PARAM	Parameter Fehler

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMFireNotifyData	Sendet Daten an die zusätzliche Notify-Funktion anhand des Cookies oder an alle Funktionen
DMRemoveNotify	Entfernt die zusätzliche Notify-Funktion anhand des Cookies aus der Notify-Liste

Beispiele

Button AddNotify:

```
#include "apdefap.h"
void OnClick(char* lpszPictureName,
             char* lpszObjectName,
             char* lpszPropertyName)
{
    AddNotify();
}
```

Projekt-Funktion AddNotify:

```
extern BOOL DM_NotifyProcA(DWORD dwNotifyClass,
                          DWORD dwNotifyCode,
                          LPBYTE lpbyData,
                          DWORD dwItems,
                          LPVOID lpvUser);

void AddNotify()
{
    BOOL bRet = FALSE;
    DWORD dwNotifyCookie = 0L;
    LPVOID lpvUser = NULL;
    CMN_ERRORA err;

    memset(&err, 0, sizeof(CMN_ERRORA));

    dwNotifyCookie = GetTagDWord("DMdwNotifyCookie");
    if (dwNotifyCookie)
    {
        printf("\r\nremove first previous Notify Cookie=%08lx",
              dwNotifyCookie);

        bRet = DMRemoveNotifyA(dwNotifyCookie, &err);

        if (FALSE == bRet)
        {
            printf("\r\nERROR: DMRemoveNotifyA [%s],%ld,%ld,%ld,
%ld,%ld",
                  err.szErrorText,
                  err.dwError1,
                  err.dwError2,
                  err.dwError3,
                  err.dwError4,
                  err.dwError5);
        }
    }
}
```

```

    }
    dwNotifyCookie = 0L;
    SetTagDWord("DMdwNotifyCookie", dwNotifyCookie);
}

memset(&err, 0, sizeof(CMN_ERRORA));
lpvUser = (LPVOID)dwNotifyCookie; //set only for show in notify

bRet = DMAddNotifyA(DM_NotifyProcA,
                    lpvUser,
                    &dwNotifyCookie,
                    &err);

if (bRet)
{
    printf("\r\nNotify added, Cookie=%08lx!", dwNotifyCookie);
    SetTagDWord("DMdwNotifyCookie", dwNotifyCookie);
}
else
{
    printf("\r\nERROR: DMAddNotifyA [%s],%ld,%ld,%ld,%ld,%ld",
          err.szErrorText,
          err.dwError1,
          err.dwError2,
          err.dwError3,
          err.dwError4,
          err.dwError5);
}
}
}

```

Projekt-Funktion DM_NotifyProcA:

```

extern void RemoveNotify();

BOOL DM_NotifyProcA(DWORD dwNotifyClass,
                   DWORD dwNotifyCode,
                   LPBYTE lpbyData,
                   DWORD dwItems,
                   LPVOID lpvUser)
{
    BOOL bShowParams = FALSE;
    BOOL bSpecialActionRemoveNotify = FALSE;

    printf("\r\n\r\n#### extra added DMNotifyProc [%08lx] entry: ####",
          (DWORD)lpvUser);
    switch (dwNotifyClass)
    {
    case DM_NOTIFY_CLASS_ERROR:
        switch (dwNotifyCode)
        {
        case DM_NOTIFY_SHUTDOWN:
            printf("\r\nDM_NOTIFY_CLASS_ERROR: DM_NOTIFY_SHUTDOWN");
            bShowParams = TRUE;
            bSpecialActionRemoveNotify = TRUE;
            break;
        }
    }
}

```

```
        case DM_NOTIFY_PROCESSNET_ERROR:
            printf("\r\nDM_NOTIFY_CLASS_ERROR:
DM_NOTIFY_PROCESSNET_ERROR");
            bShowParams = TRUE;
            break;
        case DM_NOTIFY_SYSNET_ERROR:
            printf("\r\nDM_NOTIFY_CLASS_ERROR:
DM_NOTIFY_SYSNET_ERROR");
            bShowParams = TRUE;
            break;
        default:
            printf("\r\nunknown DM_NOTIFY_CLASS_ERROR:
dwNotifyCode[%08lx]", dwNotifyCode);
            bShowParams = TRUE;
    }
    break;
case DM_NOTIFY_CLASS_WARNING:
    switch (dwNotifyCode)
    {
        case DM_NOTIFY_QUEUE_50_PERCENT:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_QUEUE_50_PERCENT");
            break;
        case DM_NOTIFY_QUEUE_60_PERCENT:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_QUEUE_60_PERCENT");
            break;
        case DM_NOTIFY_QUEUE_70_PERCENT:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_QUEUE_70_PERCENT");
            break;
        case DM_NOTIFY_QUEUE_80_PERCENT:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_QUEUE_80_PERCENT");
            break;
        case DM_NOTIFY_QUEUE_90_PERCENT:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_QUEUE_90_PERCENT");
            break;
        case DM_NOTIFY_QUEUE_OVERFLOW:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_QUEUE_OVERFLOW");
            break;
        case DM_NOTIFY_CYCLES_CHANGED:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_CYCLES_CHANGED");
            bShowParams = TRUE;
            break;
        case DM_NOTIFY_MACHINES_CHANGED:
            printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_MACHINES_CHANGED");
            bShowParams = TRUE;
```

```

        break;
    case DM_NOTIFY_PROJECT_OPENED:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_PROJECT_OPENED [%s]",
            (LPSTR)lpbyData);
        bSpecialActionRemoveNotify = TRUE;
        break;
    case DM_NOTIFY_PROJECT_CLOSED:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_PROJECT_CLOSE");
        bSpecialActionRemoveNotify = TRUE;
        break;
    case DM_NOTIFY_SYSTEM_LOCALE:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_SYSTEM_LOCALE [%ld]",
            *(DWORD*)lpbyData);
        break;
    case DM_NOTIFY_DATA_LOCALE:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_DATA_LOCALE [%ld]",
            *(DWORD*)lpbyData);
        break;
    case DM_NOTIFY_PROJECT_RUNTIME:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_PROJECT_RUNTIME [%s]",
            (LPSTR)lpbyData);
        break;
    case DM_NOTIFY_PROJECT_EDIT:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_PROJECT_EDIT [%s]",
            (LPSTR)lpbyData);
        bSpecialActionRemoveNotify = TRUE;
        break;
    case DM_NOTIFY_HOTKEY_CHANGE:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_HOTKEY_CHANGE");
        bShowParams = TRUE;
        break;
    case DM_NOTIFY URSEL:
        printf("\r\nDM_NOTIFY_CLASS_WARNING: DM_NOTIFY_URSEL");
        bShowParams = TRUE;
        break;
    case DM_NOTIFY_BODO:
        printf("\r\nDM_NOTIFY_CLASS_WARNING: DM_NOTIFY_BODO");
        bShowParams = TRUE;
        break;
    case DM_NOTIFY_BEGIN_PROJECT_EDIT:
        printf("\r\nDM_NOTIFY_CLASS_WARNING:
DM_NOTIFY_BEGIN_PROJECT_EDIT [%s]",
            (LPSTR)lpbyData);
        bShowParams = TRUE;
        bSpecialActionRemoveNotify = TRUE;

```

```

        break;
    default:
        printf("\r\nunknown DM_NOTIFY_CLASS_WARNING:
dwNotifyCode[%08lx]", dwNotifyCode);
        bShowParams = TRUE;
    }
    break;
case DM_NOTIFY_CLASS_DATA:
    switch (dwNotifyCode)
    {
        case DM_NOTIFY_APPLICATION_DATA:
            printf("\r\nDM_NOTIFY_CLASS_DATA:
DM_NOTIFY_APPLICATION_DATA:");
            bShowParams = TRUE;
            break;
        case DM_NOTIFY_VARIABLE_DATA:
            printf("\r\nDM_NOTIFY_CLASS_DATA:
DM_NOTIFY_VARIABLE_DATA:");
            bShowParams = TRUE;
            break;
        case DM_NOTIFY_FIRE_DATA:
            // the data sented with DMFireNotifyData is char text
            printf("\r\nDM_NOTIFY_CLASS_DATA: DM_NOTIFY_FIRE_DATA:
(data as text: [%s])",
                (char*)lpbyData);
            bShowParams = TRUE;
            break;
        default:
            printf("\r\nunknown DM_NOTIFY_CLASS_DATA:
dwNotifyCode[%08lx]", dwNotifyCode);
            bShowParams = TRUE;
    }
    break;
default:
    printf("\r\nunknown dwNotifyClass[%08lx],
dwNotifyCode[%08lx]",
        dwNotifyClass,
        dwNotifyCode);
    bShowParams = TRUE;
}

if (bShowParams)
{
    printf("\r\ndwNotifyClass=%08lx, dwNotifyCode=%08lx,
dwItems=%ld, lpbyData=%08lx, lpvUser=%08lx",
        dwNotifyClass,
        dwNotifyCode,
        dwItems,
        lpbyData,
        (DWORD)lpvUser);
}

```

```

if (bSpecialActionRemoveNotify)
{
printf("\r\nHave to remove notify if RT exists:");
RemoveNotify();
}

printf("\r\n#### extra added Notify [%08lx] exit ####\r\n",
(DWORD)lpvUser);
return TRUE;
}

```

2.2.3.3 DMChangeDataLocale (RT Professional)

Verwendung

Veranlasst die Benachrichtigung aller mit DMConnect angeschlossenen Applikationen über die Umschaltung der Runtime-Sprache.

Deklaration

```

BOOL DMChangeDataLocale (
    LPCTSTR        lpszProjectFile,
    DWORD          dwLocaleID,
    LPCMN_ERROR    lpdmError );

```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

dwLocaleID

Zeiger auf den Code der neu eingestellten Sprache. Mögliche Werte sind die Codes aller in der Text Library projektierten Sprachen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Applikationen über Sprachumschaltung benachrichtigt.

FALSE

Fehler.

Bemerkung

Die Applikation wird über die Callback-Funktion `DM_NOTIFY_PROC` (Seite 281) mit folgenden Parametern benachrichtigt:

<code>dwNotifyClass</code>	<code>DM_NOTIFY_CLASS_WARNING</code>
<code>dwNotifyCode</code>	<code>DM_NOTIFY_DATA_LOCALE</code>
<code>lpbyData</code>	Zeiger auf den Code der neu einzustellenden Sprache.

Fehlermeldungen

<code>DM_E_NOT_CONNECTED</code>	Keine Verbindung zum Datenmanager
<code>DM_E_NOT_SUPPORTED</code>	Angeforderter Dienst nicht verfügbar

Benötigte Files

`dmclient.h`

`dmclient.lib`

`dmclient.dll`

Verwandte Funktionen

<code>AUTOHOTSPOT</code>	Verbindung zum Datenmanager aufnehmen
--------------------------	---------------------------------------

Siehe auch

`DM_NOTIFY_PROC` (Seite 281)

`DMEnumOpenedProjects` (Seite 305)

`DMGetRuntimeProject` (Seite 311)

2.2.3.4 DMConnect (RT Professional)

Verwendung

Verbindungsaufnahme einer Applikation mit dem Datenmanager. In einer Applikation (Prozess) kann nur ein `DMConnect` durchgeführt werden. Weitere Aufrufe liefern den Fehler `DM_E_ALREADY_CONNECTED`.

Deklaration

```
BOOL DMConnect (
    LPTSTR          lpszAppName,
    DM_NOTIFY_PROC lpfmNotify,
    LPVOID          lpvUser,
    LPCMN_ERROR     lpcmError );
```

Parameter

lpszAppName

Zeiger auf den Namen der aufrufenden Applikation. Der Name kann beliebig gewählt werden, da der Parameter zur internen Identifikation dient.

Die Länge des Applikationsnamens ist auf MAX_DM_APP_NAME (32 Zeichen) beschränkt. Längere Namen führen in der unterlagerten OHIOIPC.DLL zu Fehlern und zum Abbruch mit dem Fehler DM_E_NOT_CONNECTED.

lpfnNotify

Zeiger auf die Benachrichtigungsfunktion für administrative Nachrichten des Datenmanagers an die Applikation.

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Die Notify-Routine sollte auch noch nach DMDisconnect vorhanden sein, da verspätete Aufrufe eintreffen können.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Call-back-Funktion wieder zu Verfügung gestellt.

lpcmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung mit Datenmanager eingerichtet.

FALSE

Fehler.

Bemerkung

In den WinCC-Varianten, die auf den Betriebssystem-Plattformen mit Multithread-Unterstützung implementiert werden, erfolgen die Aufrufe der Benachrichtigungsfunktion asynchron zum aufrufenden Prozess.

Sie müssen eine entsprechende Synchronisation innerhalb der Applikation einrichten.

Wenn in einem Programm Datenmanager-Funktionen auf mehrere Threads verteilt werden, führen Sie in jedem dieser Threads ein DMConnect und am Ende ein DMDisconnect durch.

Prüfen Sie erst mit DMGetConnectionState, ob ein anderer Thread bereits eine Verbindung aufgebaut hat. Wenn schon eine Verbindung besteht, dürfen Sie keinen DMConnect und keinen DMDisconnect durchführen.

Um diesen Konflikt zu vermeiden, bündeln Sie in einem einzigen Thread die Funktionsaufrufe an den Datenmanager und führen Sie die Aufrufe nur von dort aus.

Wenn Datenmanager-Funktionen in eine DLL ausgelagert sind, die in eine Applikation mit Datenmanager-Funktionen eingebunden ist, prüfen Sie, ob der Verbindungsaufbau und Verbindungsabbau über die DLL laufen kann. In diesem Fall muss die Applikation ihre eigenen Datenmanager-Funktionsaufrufe jeweils mit DMGetConnectionState absichern. Bei Thread-Kapselung in der DLL bietet dies jedoch keine Sicherheit vor Verbindungsabbruch. Definieren Sie darum einen eigenen Verbindungsaufbau in einem Flag, damit DMDisconnect nur im gewünschten Fall durchgeführt wird.

Ein ungeprüfter Aufruf DMConnect mit Notify, der die Fehlermeldung DM_E_ALREADY_CONNECTED liefert, kann die Notify eines vorangegangenen Aufrufs abbrechen und zu unerwünschten Ergebnissen führen.

An die Rückruffunktion übergebene Zeiger auf Datenbereiche haben nur innerhalb der Funktion Gültigkeit, d. h. nach der Rückkehr der Callback gibt der Datenmanager eventuell allokierte Speicherbereiche frei. Benötigt die Applikation über die Zeit des Aufrufs hinaus Zugriff auf die Daten, muss sie diese entsprechend kopieren.

Darüber hinaus besitzen Zeiger auf Speicherbereiche auf den Plattformen, die dies unterstützen, ausschließlich Lesezugriff, sodass ein Schreibzugriff auf relevante Daten eine Schutzverletzung verursacht!

Der Rückgabewert der Callback wird z. Z. nicht ausgewertet und wird erst in späteren Versionen von Bedeutung sein. Die Applikation sollte hier standardmäßig TRUE liefern.

API-DLLs dürfen nicht direkt in (ISS-)Services verwendet werden, da dort die internen benötigten Ressourcen nicht verfügbar sind.

Wenn Sie Datenmanager-Funktionen z. B. über einen Wrapper in einer managed Umgebung (C#) verwenden, beachten Sie, dass mehrere ApplicationDomains in einer Applikation im selben Prozess laufen. Daher ist über alle ApplicationDomains nur ein DMConnect zulässig.

Fehlermeldungen

DM_E_ALREADY_CONNECTED	Verbindung zum Datenmanager besteht bereits.
DM_E_NOT_CONNECTED	Verbindung zum Datenmanager konnte nicht aufgebaut werden (z. B. zu langer Applikationsname, ...).
DM_E_ACCESS_FAULT	keine Rechte um Verbindung zu Datenmanager aufzubauen (ungültiger Benutzer-Kontext z. B. in ISS-Diensten, ...).

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DMDisconnect (Seite 285)	Verbindung zum Datenmanager abbauen
DMGetConnectionState (Seite 297)	Verbindung zum Datenmanager prüfen
DM_NOTIFY_PROC (Seite 281)	Benachrichtigungsfunktion

Beispiele

Connection to DM (Seite 476) "DM01.cpp"

Siehe auch

DM_NOTIFY_PROC (Seite 281)
DMDisconnect (Seite 285)
DMGetConnectionState (Seite 297)
Connection to DM (Seite 476)
APConnect (Seite 605)

2.2.3.5 DM_NOTIFY_PROC (RT Professional)

Beschreibung

Damit Ihre Applikation über z.B: Sprachumschaltungen benachrichtigt werden kann, müssen Sie eine Callback-Funktion vom Typ DM_NOTIFY_PROC bereitstellen.

Deklaration

```

BOOL ( * DM_NOTIFY_PROC) (
    DWORD      dwNotifyClass,
    DWORD      dwNotifyCode,
    LPBYTE     lpbyData,
    DWORD      dwItems,
    LPVOID     lpvUser);
    
```

Parameter

dwNotifyClass

kennzeichnet die Benachrichtigungsklasse:

DM_NOTIFY_CLASS_ERROR	(0x00000001)
DM_NOTIFY_CLASS_WARNING	(0x00000002)
DM_NOTIFY_CLASS_DATA	(0x00000003)

dwNotifyCode

Benachrichtigungscode

für DM_NOTIFY_CLASS_ERROR:		
DM_NOTIFY_SHUTDOWN	(0x00000001)	Datenmanager wird beendet
DM_NOTIFY_PROCESSNET_ERROR	(0x00000002)	Fehler auf dem Prozessbus
DM_NOTIFY_SYSNET_ERROR	(0x00000003)	Fehler auf dem Systembus
für DM_NOTIFY_CLASS_WARNING:		
DM_NOTIFY_QUEUE_50_PERCENT	(0x00000001)	Füllstand der Applikationsqueue 50%
DM_NOTIFY_QUEUE_60_PERCENT	(0x00000002)	Füllstand der Applikationsqueue 60%
DM_NOTIFY_QUEUE_70_PERCENT	(0x00000003)	Füllstand der Applikationsqueue 70%
DM_NOTIFY_QUEUE_80_PERCENT	(0x00000004)	Füllstand der Applikationsqueue 80%
DM_NOTIFY_QUEUE_90_PERCENT	(0x00000005)	Füllstand der Applikationsqueue 90%
DM_NOTIFY_QUEUE_OVERFLOW	(0x00000006)	Überlauf der Applikationsqueue
DM_NOTIFY_CYCLES_CHANGED	(0x00000010)	Aktualisierungszyklen neu einlesen
DM_NOTIFY_MACHINES_CHANGED	(0x00000011)	Rechnerliste neu einlesen
DM_NOTIFY_PROJECT_OPENED	(0x00000012)	Projekt wird geladen
DM_NOTIFY_PROJECT_CLOSE	(0x00000013)	Projekt wird geschlossen
DM_NOTIFY_SYSTEM_LOCALE	(0x00000014)	Projektierungssprache umschalten
DM_NOTIFY_DATA_LOCALE	(0x00000015)	Oberflächensprache umschalten
DM_NOTIFY_PROJECT_RUNTIME	(0x00000016)	Projekt wird aktiviert

DM_NOTIFY_PROJECT_EDIT	(0x00000017)	Projekt wird deaktiviert
DM_NOTIFY_HOTKEY_CHANGE	(0x00000018)	Ein Hotkey wurde verändert
für DM_NOTIFY_CLASS_DATA:		
DM_NOTIFY_APPLICATION_DATA	(0x00000001)	Applikationsdaten
DM_NOTIFY_VARIABLE_DATA	(0x00000002)	Variablendaten

Weitere Informationen zu den Benachrichtigungscodes finden Sie bei der Beschreibung der Konstanten.

lpbyData

Zeiger auf die innerhalb der Klasse DM_NOTIFY_CLASS_DATA bereitgestellten Daten.

dwItems

Anzahl der Einträge in lpbyData.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

Die Rückgabewerte sind abhängig von Ihrer Implementierung.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus der selben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf DMConnect zurückgekehrt ist.

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMConnect (Seite 278)	Verbindung zum Datenmanager aufnehmen
-----------------------	---------------------------------------

Beispiele

AUTOHOTSPOT "DM01.cpp"

Siehe auch

DMConnect (Seite 278)

2.2.3.6 DMDeactivateRTProject (RT Professional)

Verwendung

Das im Runtime-Modus befindliche Projekt wird deaktiviert.

Deklaration

```
BOOL DMDeactivateRTProject (  
    LPCMN_ERROR    lpdmError);
```

Parameter

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Projekt deaktiviert.

FALSE

Fehler.

Fehlermeldungen

DM_E_NO_RT_PRJ	Kein Projekt im Runtime Mode
----------------	------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Beispiele

OnTestDeactivateRuntimeProject (Seite 488) "TESTCDoc.cpp"

Siehe auch

OnTestDeactivateRuntimeProject (Seite 488)

2.2.3.7 DMDisconnect (RT Professional)

Verwendung

Mit dieser Funktion baut eine Applikation eine bestehende Verbindung zum Datenmanager ab.

Deklaration

```
BOOL DMDisConnect (  
    LPCMN_ERROR    lpdmError );
```

Parameter

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung abgebaut.

FALSE

Fehler.

Bemerkungen

Falls noch keine Verbindung zum Datenmanager mit der Funktion DMConnect eingerichtet wurde, ist der Rückgabewert FALSE. Der Fehlercode lpdmError->dwError enthält den Wert DM_E_NOT_CONNECTED, keine Verbindung zum Datenmanager.

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX, ...) benutzt werden. Aufgrund von Microsoft-spezifischen Mechanismen kann das ggfs. zum Hängenbleiben des Aufrufs und damit des Programms führen.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DMConnect (Seite 278)	Verbindung zum Datenmanager aufnehmen
-----------------------	---------------------------------------

Beispiele

Connection to DM (Seite 476) "DM01.cpp"

Siehe auch

DMConnect (Seite 278)
Connection to DM (Seite 476)

2.2.3.8 DMEnumNumberFormats (RT Professional)

Verwendung

Die Funktion listet alle innerhalb der FORMAT.DLL verfügbaren Formatkonvertierungen auf.

Deklaration

```

BOOL DMEnumNumberFormats (
    LPDWORD          lpdwItems,
    DM_ENUM_FORMATS_PROC lpfnFormat,
    LPVOID           lpvUser,
    LPCMN_ERROR      lpdmError);

```

Parameter

lpdwItems

Zeiger auf eine Doppelwortvariable der Applikation, die nach dem Aufruf die Anzahl der enumerierten Formatinformationen beinhaltet.

lpfnFormat

Zeiger auf Ihre Callback Funktion, die für jedes verfügbare Zahlenformat aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Formatkonvertierungen aufgelistet.

FALSE

Fehler.

Bemerkungen

An die Callback übergebene Zeiger haben nur innerhalb dieser Gültigkeit, da das System nach Rückkehr der Funktion alle evtl. allokierten Speicherbereich wieder frei gibt. Wenn eine Applikation außerhalb Zugriff auf die Daten benötigt, muss sie diese entsprechend kopieren.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DM_ENUM_FORMATS_PROC (Seite 288)	Formatkonvertierungen auflisten (Callback)
----------------------------------	--

Siehe auch

DM_ENUM_FORMATS_PROC (Seite 288)

2.2.3.9 DM_ENUM_FORMATS_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Formatkonvertierungen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_FORMATS_PROC bereitstellen.

Deklaration

```
BOOL ( * DM_ENUM_FORMATS_PROC ) (  
    LPDM_FORMAT_INFO    lpdmFormat,  
    DWORD                dwItem,  
    LPVOID               lpvUser );
```

Parameter

lpdmFormat

Zeiger auf eine Struktur vom Typ DM_FORMAT_INFO (Seite 220) mit den Daten einer Formatkonvertierung.

dwItem

Fortlaufender Aufrufzähler. Wird die Enumeration nicht vorzeitig abgebrochen so enthält dwItem die Anzahl der verfügbaren Formatkonvertierungen.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus der selben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMEnumNumberFormats (Seite 286)	Formatkonvertierungen auflisten
---------------------------------	---------------------------------

Siehe auch

DMEnumNumberFormats (Seite 286)

DM_FORMAT_INFO (Seite 220)

2.2.3.10 DMEnumUpdateCycles (RT Professional)

Verwendung

Die Funktion listet alle im System definierten Aktualisierungszyklen auf. Die Funktion übergibt bei jedem Aufruf der Callback-Funktion eine Informationsstruktur für den Zyklus.

Deklaration

```

BOOL DMEnumUpdateCycles (
    LPCSTR                lpszProjectFile,
    LPDWORD               lpdwItems,
    DM_ENUM_CYCLES_PROC  lpfnCycle,
    LPVOID                lpvUser,
    LPCMN_ERROR           lpdmError);

```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

lpdwItem

Zeiger auf eine Doppelwortvariable der Applikation, die nach dem Aufruf die Anzahl der enumerierten Zyklusinformationen beinhaltet.

lpfnCycle

Zeiger auf Ihre Call-back Funktion, die für jeden verfügbaren Aktualisierungszyklus aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Call-back-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktualisierungszyklen aufgelistet

FALSE

Fehler

Bemerkungen

An die Funktion Callback übergebene Zeiger haben nur innerhalb der Funktion Gültigkeit, da das System nach Rückkehr der Funktion alle eventuell allokierten Speicherbereiche wieder frei gibt.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_CYCLES_PROC (Seite 291)	Aktualisierungszyklen auflisten (Call-back)
---------------------------------	---

Beispiele

OnTestUpdateCycles (Seite 496) "TESTCDoc.cpp"

Siehe auch

DM_ENUM_CYCLES_PROC (Seite 291)

OnTestUpdateCycles (Seite 496)

2.2.3.11 DM_ENUM_CYCLES_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Aktualisierungszyklen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_CYCLES_PROC bereitstellen.

Deklaration

```

BOOL ( * DM_ENUM_CYCLES_PROC) (
    LPDM_CYCLE_INFO    lpdmCycle,
    DWORD              dwItem,
    LPVOID              lpvUser);

```

Parameter

lpdmCycle

Zeiger auf eine Struktur vom Typ DM_CYCLE_INFO (Seite 216) mit den Daten eines Aktualisierungszyklen.

dwItem

Fortlaufender Aufrufzähler. Wird die Enumeration nicht vorzeitig abgebrochen so enthält dwItem die Anzahl der verfügbaren Aktualisierungszyklen.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Files

dmclient.h

Verwandte Funktionen

DMEnumUpdateCycles (Seite 289)	Aktualisierungszyklen auflisten
--------------------------------	---------------------------------

Siehe auch

DMEnumUpdateCycles (Seite 289)

DM_CYCLE_INFO (Seite 216)

2.2.3.12 DMExitWinCC (RT Professional)

Verwendung

Mit dieser Funktion beenden Sie WinCC.

Deklaration

```
BOOL DMExitWinCC (  
    VOID);
```

Parameter

Keine

Bemerkungen

Mit der Funktion `DMExitWinCCEX` legen Sie zusätzlich fest, wie sich das Betriebssystem nach der Beendigung von WinCC verhält.

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DMExitWinCCEX (Seite 293)	WinCC mit Vorgaben beenden
---------------------------	----------------------------

Beispiele

OnTestWinCCShutdown (Seite 513) "TESTCDoc.cpp"

Siehe auch

OnTestWinCCShutdown (Seite 513)
DMExitWinCCEX (Seite 293)

2.2.3.13 DMExitWinCCEX (RT Professional)

Verwendung

Mit dieser Funktion beenden Sie WinCC. Zusätzlich können Sie festlegen, wie sich das Betriebssystem nach dem Beenden von WinCC verhalten soll.

Deklaration

```
BOOL DMExitWinCCEX (  
    DWORD    dwMode );
```

Parameter

dwMode

Über dwMode können Sie verschiedene Arten festlegen, wie sich das Betriebssystem nach dem Beenden von WinCC verhalten soll:

DM_SDMODE_WINCC	WinCC wird beendet
DM_SDMODE_LOGOFF	Nach dem Beenden von WinCC wird der Benutzer auch beim Betriebssystem abgemeldet.
DM_SDMODE_SYSTEM	Nach dem Beenden von WinCC wird das Betriebssystem heruntergefahren.
DM_SDMODE_REBOOT	Nach dem Beenden von WinCC wird das Betriebssystem neu gebootet.

Bemerkungen

Mit dem Parameter DM_SDMODE_WINCC besitzt die Funktion die gleichen Auswirkungen wie DMEExitWinCC.

Benötigte Files

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMEExitWinCC (Seite 292)	WinCC beenden
--------------------------	---------------

Siehe auch

DMEExitWinCC (Seite 292)

2.2.3.14 DMFireNotifyData (RT Professional)

Deklaration

```
BOOL DMFireNotifyData (  
    DWORD          dwNotifyCookie,  
    DWORD          dwByteCount,  
    LPBYTE         lpbyData,  
    LPCMN_ERROR    lpdmError);
```

Beschreibung

Sendet eine zusätzliche Notify an eine angebundene Notify anhand der angegebenen Cookie-Kennung.

Parameter

dwNotifyCookie

Cookie einer zusätzlichen Notify-Funktion, die mit "DMAddNotify" angebunden wurde.

Wenn "0" angegeben ist, werden die Notify-Daten an alle aktiven Notifies versendet (Broadcast).

dwByteCount

Größe des zu versendenden Datenpuffers in "BYTE".

lpbyData

Zeiger auf einen "BYTE"-Puffer, der die zu versenden Notify-Daten enthält.

Die Größe des Puffers wird in "dwByteCount" angegeben.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Notify-Daten versendet.

FALSE

Fehler

Bemerkungen

Die gesendete Notify wird in der angesprochenen Notify-Routine unter "dwNotifyClass=DM_NOTIFY_CLASS_DATA" und "dwNotifyCode=DM_NOTIFY_FIRE_DATA" bearbeitet.

Wenn ein "DMDisconnect" durchgeführt wird, ohne vorher die zusätzlichen Notify-Funktionen zu löschen, wird als Abschluß noch einmal an alle diese Notify-Funktionen ein "DM_NOTIFY_SHUTDOWN" abgesetzt und diese Funktionen dann intern mit der Verwaltung gelöscht. Sie können dann die Anbindung und Cookies wegwerfen, da nicht mehr gültig.

Fehlermeldungen

DM_E_NOT_CONNECTED	Noch kein "DMConnect" durchgeführt bzw. "DMDisconnect" wurde schon durchgeführt
DM_E_DONT_EXIST	Zum angegebenen Cookie existiert keine Notify
DM_E_PARAM	Parameter Fehler

Benötigte Files

dmclient_exstr.h
 dmclient.lib
 dmclient.dll

Verwandte Funktionen

DMAddNotify	Bindet eine zusätzliche Notify-Funktion an "DMClient" an
DMFireNotify	Entfernt die zusätzliche Notify-Funktion anhand des Cookies aus der NotifyListe

Beispiele

Button DMFireNotifyData:

```
#include "apdefap.h"
void OnClick(char* lpszPictureName,
             char* lpszObjectName,
             char* lpszPropertyName)
{
    DWORD dwNotifyCookie;
    CHAR* pszNotifyText;
    CMN_ERROR err;
    BOOL bRet;

    dwNotifyCookie = GetTagDWord("DMdwNotifyCookie");
    pszNotifyText = GetTagChar("DMNotifySendText");
    memset(&err, 0, sizeof(err));
    bRet = FALSE;

    bRet = DMFireNotifyData(dwNotifyCookie,
                           strlen(pszNotifyText)*sizeof(CHAR)+1,
                           (LPBYTE)pszNotifyText, &err);

    if (FALSE == bRet)
    {
        printf("\r\nERROR: DMFireNotifyData [%s],%ld,%ld,%ld,%ld,%ld",
              err.szErrorText,
              err.dwError1,
              err.dwError2,
              err.dwError3,
              err.dwError4,
```

```

        err.dwError5);
    }
}

```

2.2.3.15 DMGetConnectionState (RT Professional)

Verwendung

Mit dieser Funktion wird abgefragt, ob eine Verbindung zum Datenmanager besteht, z. B. um nachzuprüfen, ob der Funktionsaufruf DMConnect korrekt ausgeführt wurde.

Deklaration

```

BOOL DMGetConnectionState (
    LPCMN_ERROR    lpdmError);

```

Parameter

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung besteht.

FALSE

Fehler oder Verbindung besteht nicht.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Files

```

dmclient.h
dmclient.lib
dmclient.dll

```

Verwandte Funktionen

DMConnect (Seite 278)	Verbindung zum Datenmanager aufnehmen
-----------------------	---------------------------------------

Beispiele

AUTOHOTSPOT "DM01.cpp"

Siehe auch

- DMConnect (Seite 278)
- Connection to DM (Seite 476)

2.2.3.16 DMGetDataLocale (RT Professional)

Verwendung

Die Funktion ermittelt die ID der aktuell eingestellten Runtime-Sprache.

Deklaration

```
BOOL DMGetDataLocale (  
    LPDWORD      lpdwLocaleID,  
    LPCMN_ERROR  lpdmError);
```

Parameter

lpdwLocaleID

Zeiger auf den Code der aktuell eingestellten Sprache. Rückgabewerte sind die Codes aller in der Text Library projektierten Sprachen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Sprach-ID ermittelt.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

2.2.3.17 DMGetHotkey (RT Professional)

Verwendung

Diese Funktion ermittelt den zu einer Aktion gehörenden Hotkey.

Deklaration

```

BOOL DMGetHotKey (
    DWORD          dwHotKeyAction,
    LPDWORD        lpdwHotKey,
    LPCMN_ERROR    lpdmError);

```

Parameter

dwHotKeyAction

Aktion

lpdwHotKey

Zeiger auf die Speicherstelle in der der Hotkey abgelegt wird.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Hotkey-ID ermittelt.

FALSE

Fehler.

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

2.2.3.18 DMGetMachineInfo (RT Professional)

Verwendung

Mit dieser Funktion ermittelt eine Applikation ihre rechnerabhängigen Anlaufparameter. Dazu muss die Applikation jedoch vorher mit den Funktionen des Projektmanagements entsprechende Einträge für die Rechnereigenschaften vorgenommen haben.

Deklaration

```
BOOL DMGetMachineInfo (  
    LPCSTR      lpszLogicalName,  
    LPVOID      lpvData,  
    LPDWORD     lpdwSize,  
    LPCMN_ERROR lpdmError);
```

Parameter

lpszLogicalName

Zeiger auf den logischen Namen der Explorer-DLL, deren Anlaufparameter abgefragt werden.

lpvData

Zeiger auf einen Datenbereich, über den die Daten der Anlaufparameter an die Applikation übergeben werden.

lpdwSize

Ist lpvData = 0, so wird in lpdwSize die Größe des Datenbereichs angegeben.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Informationen über den lokalen PC ermittelt.

FALSE

Fehler.

Fehlermeldungen

DM_E_NO_RT_PRJ	Kein Projekt im Runtime Mode
DM_E_MACHINE_NOT_FOUND	PC nicht gefunden
DM_E_NO_INFO_FOUND	keine Startinformationen gefunden

Benötigte Files

dmclient.h

dmclient.lib

dmclient.dll

2.2.3.19 DMGetMachineTable (RT Professional)**Verwendung**

Mit dieser Funktion erhalten Sie neben der Anzahl der am Projekt beteiligten PCs auch Informationen zu den einzelnen PCs.

Deklaration

```

BOOL DMGetMachineTable (
    LPCSTR          lpszProjectFile,
    LPDM_MACHINE_TABLE lpdmMachineTable,
    LPCMN_ERROR     lpdmError);

```

Parameter**lpdwProjectFile**

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

lpdmMachineTable

Zeiger auf die Struktur DM_MACHINE_TABLE (Seite 221) , in der die Daten der Rechnerliste hinterlegt werden.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Daten zur Rechnerliste ermittelt.

FALSE

Fehler.

Benötigte Files

dmclient.h

dmclient.lib

dmclient.dll

Beispiele

OnTestMachines (Seite 491) "TESTCDoc.cpp"

Siehe auch

DM_MACHINE_TABLE (Seite 221)

OnTestMachines (Seite 491)

DMActivateRTProject (Seite 269)

2.2.3.20 DMRemoveNotify (RT Professional)

Deklaration

```
BOOL DMRemoveNotify (  
    DWORD          dwNotifyCookie,  
    LPCMN_ERROR    lpdmError);
```

Beschreibung

Löscht eine zusätzlich angebundene Notify-Funktion anhand ihrer Cookie-Kennung wieder aus der zusätzlichen Notify-Liste.

Diese Funktion steht erst ab WinCC Version V7.2 zur Verfügung.

Parameter

dwNotifyCookie

Cookie einer zusätzlichen Notify-Funktion, die mit "DMAddNotify" angebunden wurde.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Zusätzliche Notify-Funktion ausgetragen.

FALSE

Fehler

Bemerkungen

Wenn ein "DMDisconnect" durchgeführt wird, ohne vorher die zusätzlichen Notify-Funktionen zu löschen, wird als Abschluß noch einmal an alle diese Notify-Funktionen ein "DM_NOTIFY_SHUTDOWN" abgesetzt und diese Funktionen dann intern mit der Verwaltung gelöscht. Sie können dann die Anbindung und Cookies wegwerfen, da nicht mehr gültig.

Fehlermeldungen

DM_E_NOT_CONNECTED	Noch kein "DMConnect" durchgeführt bzw. "DMDisconnect" wurde schon durchgeführt
DM_E_DONT_EXIST	Zum angegebenen Cookie existiert keine Notify
DM_E_PARAM	Parameter Fehler

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMFireNotifyData	Sendet Daten an die zusätzliche Notify-Funktion anhand des Cookies oder an alle Funktionen
DMAddNotify	Bindet eine zusätzliche Notify-Funktion an "DMClient" an

Beispiele**Button RemoveNotify:**

```
#include "apdefap.h"
void OnClick(char* lpszPictureName,
             char* lpszObjectName,
```

```

        char* lpszPropertyName)
    {
    RemoveNotify ();
    }

Projekt-Funktion RemoveNotify:
void RemoveNotify()
{
    DWORD dwNotifyCookie = 0L;
    CMN_ERRORA err;
    BOOL bRet = FALSE;

    memset(&err, 0, sizeof(CMN_ERRORA));
    dwNotifyCookie = GetTagDWord("DMdwNotifyCookie");

    if (dwNotifyCookie)
    {
        bRet = DMRemoveNotify(dwNotifyCookie, &err);
        if (bRet)
        {
            printf("\r\nNotify [%08lx] removed", dwNotifyCookie);
            dwNotifyCookie = 0L;
            SetTagDWord("DMdwNotifyCookie", dwNotifyCookie);
        }
        else
        {
            printf("\r\nERROR: DMRemoveNotifyA [%s],%ld,%ld,%ld,
%ld,%ld, Cookie=%08lx",
                err.szErrorText,
                err.dwError1,
                err.dwError2,
                err.dwError3,
                err.dwError4,
                err.dwError5,
                dwNotifyCookie);
        }
    }
    else
    {
        printf("\r\nNo Notify present to remove!");
    }
}

```

2.2.4 Funktionen zur Projektverwaltung (RT Professional)

2.2.4.1 DMLenumOpenedProjects (RT Professional)

Verwendung

Ruft für jedes in WinCC geöffnete Projekt die übergebene Callback-Funktion auf.

Deklaration

```
BOOL DMLenumOpenedProjects (
    LPDWORD                                lpdwItems,
    DM_ENUM_OPENED_PROJECTS_PROC          lpfnEnum,
    LPVOID                                  lpvUser,
    LPCMN_ERROR                            lpdmError);
```

Parameter

lpdwItems

Zeiger auf eine Doppelwortvariable, in der nach Beendigung der Enumeration die Anzahl der geöffneten Projekte abgelegt wird.

lpfnEnum

Zeiger auf Ihre Callback-Funktion, welche die Projektdaten entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Projekte aufgelistet.

FALSE

Fehler.

Bemerkungen

Derzeit wird in WinCC nur ein Projekt geöffnet. Sie erhalten daher nur die Informationen zu diesem Projekt.

Fehlermeldungen

DM_E_FILE	Fehler bei Dateioperation
-----------	---------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_OPENED_PROJECTS_PROC (Seite 306)	Geöffnete Projekte auflisten (Callback)
--	---

Beispiele

Enum open projects (Seite 481) "DM01.cpp"

OnTestOpenProjects (Seite 495) "TESTCDoc.cpp"

Siehe auch

DM_ENUM_OPENED_PROJECTS_PROC (Seite 306)

DM_PROJECT_INFO (Seite 222)

Enum open projects (Seite 481)

OnTestOpenProjects (Seite 495)

DMChangeDataLocale (Seite 277)

2.2.4.2 DM_ENUM_OPENED_PROJECTS_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Projekte auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_OPENED_PROJECTS_PROC bereitstellen.

Deklaration

```

BOOL ( * DM_ENUM_OPENED_PROJECTS_PROC) (
    LPDM_PROJECT_INFO    lpInfo,
    LPVOID                lpvUser);

```

Parameter

lpInfo

Zeiger auf eine Struktur vom Typ DM_PROJECT_INFO (Seite 222) mit den Informationen zu einem geöffneten Projekt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Bemerkungen

Derzeit kann in WinCC nur ein Projekt geöffnet werden. Sie erhalten daher nur die Informationen zu diesem Projekt.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMEnumOpenedProjects (Seite 305)	Projekte (geöffnete) auflisten
----------------------------------	--------------------------------

Beispiele

Enum open projects "DM01.cpp"

Siehe auch

DMLenumOpenedProjects (Seite 305)

DM_PROJECT_INFO (Seite 222)

Enum open projects (Seite 481)

2.2.4.3 DMGetProjectDirectory (RT Professional)

Verwendung

Liefert den für die Applikation gültigen Pfad auf die Projektierungsdaten für das übergebene Projekt.

Deklaration

```
BOOL DMGetProjectDirectory (  
    LPCSTR          lpszAppName,  
    LPCSTR          lpszProjectFile,  
    LPDM_DIRECTORY_INFO lpdmDirInfo,  
    LPCMN_ERROR     lpdmError );
```

Parameter

lpszAppName

Zeiger auf den Namen der Applikation, für die der Pfad zu ermitteln ist.

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei kann mit DMLenumOpenedProjects ermittelt werden.

lpdmDirInfo

Zeiger auf die Struktur DM_DIRECTORY_INFO (Seite 218) , in der die Pfadangaben der Projektierungsdaten abgelegt werden.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Pfadangaben ermittelt.

FALSE

Fehler.

Fehlermeldungen

DM_E_FILE	Fehler bei Dateioperation
-----------	---------------------------

Benötigte Files

dmclient.h

dmclient.lib

dmclient.dll

Beispiele

OnTestProjectPaths (Seite 493) "TESTCDoc.cpp"

Siehe auch

DM_DIRECTORY_INFO (Seite 218)

OnTestProjectPaths (Seite 493)

2.2.4.4 DMGetProjectInformation (RT Professional)**Verwendung**

Ermittelt relevante Informationen zu dem übergebenen Projekt wie Pfadangaben, Data Source Name, Rechnerkonfiguration etc.

Deklaration

```

BOOL DMGetProjectInformation (
    LPCSTR          lpszProjectFile,
    LPDM_PROJECT_INFO lpProjectInfo,
    LPCMN_ERROR     lpdmError);

```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei kann mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt werden.

lpProjectInfo

Zeiger auf die Struktur DM_PROJECT_INFO (Seite 222) , in der die Projektinformationen abgelegt werden sollen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Projektinformationen ermittelt

FALSE

Fehler

Fehlermeldungen

DM_E_FILE	Fehler bei Dateioperation
-----------	---------------------------

Benötigte Files

dmclient.h

dmclient.lib

dmclient.dll

Beispiele

Inquire project informations (Seite 487) "DM01.cpp"

OnTestProjectInfo (Seite 491) "TESTCDoc.cpp"

Siehe auch

DM_PROJECT_INFO (Seite 222)

Inquire project informations (Seite 487)

OnTestProjectInfo (Seite 491)

2.2.4.5 DMGetRuntimeProject (RT Professional)

Verwendung

Liefert den Dateinamen des im Online Modus befindlichen Projekts. Um nach erfolgreicher Ausführung alle anderen Daten des Projekts zu ermitteln, merkt sich die aufrufende Applikation den in lpszProjectFile eingetragenen Dateinamen des Projektfiles.

Deklaration

```
BOOL DMGetRuntimeProject (  
    LPSTR          lpszProjectFile,  
    DWORD         dwBufSize,  
    LPCMN_ERROR   lpdmError);
```

Parameter

lpszProjectFile

Puffer zur Aufnahme des Namens der Projektdatei, einschließlich Pfad und Extension. Der Puffer muss mindestens _MAX_PATH Zeichen groß sein.

dwBufSize

Größe des übergebenen Puffers in Zeichen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Name des aktivierten Projekts ermittelt.

FALSE

Fehler.

Bemerkungen

Der Aufruf wird bereits dann erfolgreich ausgeführt, wenn Runtime hochläuft, aber noch nicht alle Komponenten aktiviert sind.

Fehlermeldungen

DM_E_FILE	Fehler bei Dateioperation
DM_E_NO_RT_PRJ	Kein Projekt im Runtime Mode

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Beispiele

Read tag (Seite 515) "DM02.cpp"
Write tag (Seite 520) "DM02.cpp"
OnTestRuntimeProject (Seite 495) "TESTCDoc.cpp"

Siehe auch

Read tag (Seite 515)
Write tag (Seite 520)
OnTestRuntimeProject (Seite 495)
DMChangeDataLocale (Seite 277)

2.2.4.6 DMOpenProjectDocPlus (RT Professional)

Verwendung

Öffnet das in lpszProjectFile angegebene Projekt.

Deklaration

```
BOOL DMOpenProjectDocPlus (  
    LPSTR          lpszProjectFile,  
    LPCMN_ERROR   lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der zu öffnenden Projektdatei, einschließlich Pfad und Extension.

Der von WinCC implizit erzeugte Freigabename (z. B. WinCC50_Project_Odk) kann nicht als Pfad benutzt werden, da dieser nur so lange existiert, wie dieses Projekt geöffnet ist.

Der Name der Projektdatei wird mit `DMEnumOpenedProjects` oder in RT mit `DMGetRuntimeProject` ermittelt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Projekt geöffnet.

FALSE

Fehler.

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

2.2.4.7 DMOpenProjectPlus (RT Professional)

Verwendung

Auswahl einer Projektdatei über den Standard-Auswahldialog des Systems. Wenn das Projekt in WinCC noch nicht geöffnet ist, werden die für das Projekt notwendigen Informationen geladen.

Bei erfolgreicher Ausführung sollte sich die aufrufende Applikation den in `lpszProjectFile` eingetragenen Dateinamen der Projektdatei merken, da darüber alle anderen Daten des Projekts ermittelt werden können.

Der Name der Projektdatei kann später auch mit `DMEnumOpenedProjects` ermittelt werden.

Deklaration

```
BOOL DMOpenProjectPlus (  
    HWND          hwndParent,  
    LPSTR         lpszProjectFile,  
    DWORD         dwBufSize,  
    LPCMN_ERROR   lpdmError );
```

Parameter

hwndParent

Handle auf das Fenster, das als Vaterfenster für den Dialog benutzt wird.

lpszProjectFile

Zeiger auf den Puffer zur Aufnahme des Namens der Projektdatei, einschließlich Pfad und Extension. Der Puffer sollte mindestens `_MAX_PATH` Zeichen groß sein.

`lpszProjectFile` muss beim Aufruf der Funktion ein Leer-String oder ein gültiger Pfad eines existierenden Projekts sein.

Ist `lpszProjectFile` ein Leer-String, so wird der Standard-Auswahldialog des Systems (siehe oben) aufgerufen.

Ist `lpszProjectFile` nicht leer, so interpretiert WinCC die in `lpszProjectFile` enthaltene Zeichenfolge als Projektpfad und versucht diesen ohne Auswahldialog zu öffnen.

Der von WinCC implizit erzeugte Freigabename (z. B. `WinCC50_Project_Odk`) kann nicht als Projektpfad benutzt werden, da dieser nur so lange existiert, wie dieses Projekt geöffnet ist.

dwBufSize

Größe des übergebenen Puffers in Zeichen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Dialog mit "OK" verlassen.

FALSE

Fehler oder Dialog mit "ABBRECHEN" verlassen

Bemerkungen

Der Vorgang des Öffnens eines Projekts kann mehrere Minuten dauern.

Fehlermeldungen

DM_E_CANCEL	Benutzer hat in einem Dialog "Abbrechen" gewählt
DM_E_FILE	Fehler bei Dateioperation
DM_E_UPDATE	Projekt wird gerade aktualisiert

Benötigte Files

`dmclient.h`

dmclient.lib
dmclient.dll

Beispiele

Open project by means of dialog (Seite 514) "DM01.cpp"
OnTestOpenProject (Seite 494) "TESTCDoc.cpp"

Siehe auch

Open project by means of dialog (Seite 514)
OnTestOpenProject (Seite 494)

2.2.5 Datentransportkanäle (RT Professional)

2.2.5.1 DMClearBlockQueue (RT Professional)

Verwendung

Löscht alle in der Benachrichtigungswarteschlange der Applikation vorliegenden Einträge.

Deklaration

```
BOOL DMClearBlockQueue (  
    LPCMN_ERROR    lpdmError );
```

Parameter

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benachrichtigungsschlange gelöscht.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

2.2.5.2 DMEEnumDataServices (RT Professional)

Verwendung

DMEEnumDataServices ruft die übergebene Callback-Funktion für jeden bereits installierten Service auf, solange die Callback-Funktion TRUE zurückliefert.

Deklaration

```
BOOL DMEEnumDataServices (  
    LPDWORD                lpdwItems,  
    DM_ENUM_DATA_SERVICE_PROC lpfnEnum,  
    LPVOID                 lpvUser,  
    LPCMN_ERROR             lpdmError);
```

Parameter

lpdwItems

Zeiger auf eine Doppelwortvariable der Applikation, die nach dem Aufruf die Anzahl der aufgelisteten Services beinhaltet.

lpfnEnum

Zeiger auf Ihre Callback-Funktion, die für jeden installierten Service aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Services aufgelistet.

FALSE

Fehler.

Bemerkung

An die Callback übergebene Zeiger haben nur innerhalb dieser Gültigkeit, da das System nach Rückkehr der Funktion alle evtl. allokierten Speicherbereich wieder frei gibt. Wenn eine Applikation außerhalb Zugriff auf die Daten benötigt, muss sie diese entsprechend kopieren.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_DATA_SERVICE_PROC (Seite 317)	Datentransportkanäle auflisten (Callback)
DMInstallDataService (Seite 321)	Datentransportkanal installieren

Siehe auch

DM_ENUM_DATA_SERVICE_PROC (Seite 317)

DMInstallDataService (Seite 321)

2.2.5.3 DM_ENUM_DATA_SERVICE_PROC (RT Professional)**Beschreibung**

Um die vom System aufgelisteten installierten Datentransportkanäle auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_DATA_SERVICE_PROC bereitstellen.

Deklaration

```
BOOL ( * DM_ENUM_DATA_SERVICE_PROC) (  
    LPCSTR                pszService,  
    DM_DATA_SERVICE_PROC pfnService,  
    LPVOID                lpvUser );
```

Parameter

pszService

Zeiger auf den logischen Namen des Datentransportkanals.

pfnService

Zeiger auf Ihre Callback-Funktion vom Typ DM_DATA_SERVICE_PROC (Seite 319) , über die weitere Daten eines Service zur Verfügung gestellt werden..

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus der selben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMEnumDataServices (Seite 316)	Datentransportkanäle auflisten
--------------------------------	--------------------------------

Siehe auch

DMEEnumDataServices (Seite 316)

DM_DATA_SERVICE_PROC (Seite 319)

2.2.5.4 DM_DATA_SERVICE_PROC (RT Professional)**Beschreibung**

Damit Ihre Applikation Daten empfangen kann, müssen Sie über DMInstallDataService eine Datentransportkanal installiert haben und eine Callback-Funktion vom Typ DM_DATA_SERVICE_PROC bereitstellen.

Sie benötigen eine Callback-Funktion dieses Typs auch, wenn Sie die vom System aufgelisteten Transportkanäle (Services) auswerten wollen.

Deklaration

```
BOOL ( * DM_DATA_SERVICE_PROC) (  
    LPDM_DATA_SERVICE    lpds,  
    LPVOID                lpvUser);
```

Parameter**lpds**

Zeiger auf eine Struktur vom Typ DM_DATA_SERVICE (Seite 217) mit Informationen zu einem installierten Datentransportkanal (Service).

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert**TRUE**

Je nach Verwendung der Callback-Funktion.

FALSE

Fehler

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus der selben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Benötigte Files

dmclient.h

Verwandte Funktionen

DMInstallDataService (Seite 321)	Datentransportkanal installieren
DM_ENUM_DATA_SERVICE_PROC (Seite 317)	Datentransportkanäle auflisten (Callback)

Siehe auch

- DM_ENUM_DATA_SERVICE_PROC (Seite 317)
- DM_DATA_SERVICE (Seite 217)
- DMInstallDataService (Seite 321)

2.2.5.5 DMGetNumPendingBlocks (RT Professional)

Verwendung

Mit dieser Funktion ermittelt eine Applikation die in ihrer Warteschlange vorhandenen Einträge.

Deklaration

```
BOOL DMGetNumPendingBlocks (  
    LONG          *plEntries,  
    LPCMN_ERROR   lpdmError);
```

Parameter

plEntries

Enthält bei erfolgreicher Ausführung der Funktion die Anzahl der Datenpakete in der Warteschlange der Applikation.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Anzahl der Warteschlangeneinträge ermittelt.

FALSE

Fehler.

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

2.2.5.6 DMInstallDataService (RT Professional)

Verwendung

Installiert einen Transportkanal (Service) zur Datenübertragung zwischen Applikationen. Die Funktion signalisiert die Bereitschaft der Applikation, Daten, die von beliebigen anderen Applikationen unter der angegebenen Servicebezeichnung verschickt werden, zu empfangen. Wenn unter dem übergebenen Namen bereits ein Service installiert wurde, wird die dafür installierte Rückruffunktion durch die übergebene ersetzt. Ist lpfnService == NULL wird der unter lpzService installierte Service entfernt.

Deklaration

```

BOOL DMInstallDataService (
    LPCSTR                lpzService,
    DM_DATA_SERVICE_PROC lpfnService,
    LPVOID                lpvUser,
    LPCMN_ERROR           lpdmError);

```

Parameter

lpszService

Zeiger auf den Namen des Service. Über diesen, von Ihnen frei vergebaren Namen, erfolgt die Identifikation des Transportkanals.

lpfnService

Zeiger auf eine Call-back Funktion, die aufgerufen wird, wenn auf diesem Service Daten an die Applikation geschickt werden.

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, das die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Call-back-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Datentransportkanal installiert.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_DATA_SERVICE_PROC (Seite 319)	Datentransportkanäle auflisten (Callback) Datentransportkanal installieren (Callback)
----------------------------------	---

Siehe auch

DMEnumDataServices (Seite 316)

DM_DATA_SERVICE_PROC (Seite 319)

2.2.5.7 DMSendApplicationData (RT Professional)

Verwendung

Die in `lpdmSendData` übergebenen Daten werden an alle Applikationen versendet, die den in der Struktur `DM_SEND_DATA_STRUCT` festgelegten Bedingungen entsprechen und den unter `szService` beschriebenen Service mit `DMInstallDataService` installiert haben.

Deklaration

```
BOOL DMSendApplicationData (  
    LPDM_SEND_DATA_STRUCT    lpdmSendData,  
    LPCMN_ERROR               lpdmError);
```

Parameter

lpdmSendData

Zeiger auf die Struktur `DM_SEND_DATA_STRUCT` (Seite 223).

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Daten an Applikation versendet.

FALSE

Fehler.

Bemerkung

Die Funktion ist nur lokal implementiert.

Die in der Struktur `DM_SEND_DATA_STRUCT` vorhandenen Member für Remote-Zugriffe auf "TargetMachine" sind für zukünftige Erweiterung reserviert.

Das `dwTargetMachineFlags` kann nur mit `DM_SD_LOCAL` besetzt werden, alle anderen Angaben führen zu Fehler, und `dwTargetMachines` muss 0L sein.

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Siehe auch

DM_SEND_DATA_STRUCT (Seite 223)

2.2.5.8 DMSetBlockQueueSize (RT Professional)

Verwendung

Mit der Funktion wird die maximale Anzahl der in der Benachrichtigungswarteschlange der Applikation vorkommenden Nachrichten festgelegt.

Deklaration

```
BOOL DMSetBlockQueueSize (  
    LONG          nCount,  
    LPCMN_ERROR   lpdmError);
```

Parameter

nCount

Maximale Anzahl der Nachrichten.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Größe der Warteschlange festgelegt.

FALSE

Fehler.

Bemerkung

Der Standardwert der 32 Bit Plattformen liegt bei 1000 Einträgen, was in der Regel ausreichend ist.

Jede Nachricht belegt Speicherplatz im Adressraum der Applikation. Eine Erhöhung der Anzahl Nachrichten ist bei langsamer Verarbeitung durch die Applikation sorgfältig zu überlegen.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

2.2.6 Funktionen zur Bearbeitung von Variablen (RT Professional)

2.2.6.1 DMEEnumVarData (RT Professional)

Verwendung

Die Funktion dient der Bestimmung von Daten über Objekte, die mit einer Variablen verknüpft sind, z. B. zugehörige Variablengruppe, Kanal-Unit, ...

Deklaration

```

BOOL DMEEnumVarData (
    LPCSTR                lpszProjectFile,
    LPDM_VARKEY           lpdmVarKey,
    DWORD                dwItems,
    DM_ENUM_VARIABLE_PROC lpfnEnum,
    LPVOID               lpvUser,
    LPCMN_ERROR          lpdmError);

```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.
Der Name der Projektdatei wird mit DMEEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

lpdmVarKey

Zeiger auf den Anfang eines Feldes von Strukturen vom Typ DM_VARKEY (Seite 241) , über welche die aufzulistenden Variablen spezifiziert werden. Wird ab V5.0 SP2 kein Array angegeben und mit NULL versorgt, werden alle Variablen enumeriert.

Wird bei einem DM_VARKEY ein falscher Name oder eine falsche ID angegeben, so gibt es keine Fehleranzeige, da ja bei einer Liste manche gültig und manche fehlerhaft sein können. Damit der Anwender fehlerhafte DM_VARKEY erkennt, wird ihm für diese im Callback eine Struktur vom Typ DM_VARIABLE_DATA (Seite 237) übergeben, die komplett mit 0 belegt ist.

dwItems

Anzahl der Variablen deren Daten aufgelistet werden (= Anzahl der Variablenspezifikationen in lpdmVarKey). Wenn die Anzahl auf 0 gesetzt wird, werden alle Variablen enumeriert.

lpfnEnum

Zeiger auf Ihre Callback, welche die Informationen zu einer Variablen entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Informationen zur Variablen aufgelistet.

FALSE

Fehler

Bemerkung

Es gibt eine weiter entwickelte Funktion DMEnumVarData4.

Hinweis

Bestehende Implementierungen einer mehrstufigen Enumeration von DMEnumVariables mit dem Aufruf von DMEnumVarDataX in der Callback für jedes einzelne Element sollten durch einen einzelnen Aufruf von DMEnumVarData4 ersetzt werden. Eventuell notwendige Filterungen sind hier in der Callback selber durchzuführen. Die Umstellung verbessert die Performance erheblich.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden

Benötigte Dateien

dmclient.h
 dmclient.lib
 dmclient.dll

Verwandte Funktionen

DM_ENUM_VARIABLE_PROC (Seite 327)	Informationen zu einer Variablen auflisten (Call-back)
-----------------------------------	--

Siehe auch

DM_VARKEY (Seite 241)
 DM_VARIABLE_DATA (Seite 237)
 DM_ENUM_VARIABLE_PROC (Seite 327)

2.2.6.2 DM_ENUM_VARIABLE_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Informationen zu einer Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_VARIABLE_PROC bereitstellen.

Deklaration

```

BOOL ( * DM_ENUM_VARIABLE_PROC) (
    LPDM_VARKEY          lpdmVarKey,
    LPDM_VARIABLE_DATA   lpdmVarData,
    LPVOID               lpvUser
  )

```

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Typ DM_VARKEY (Seite 241) zur Spezifikation der aufzulistenden Variablen.

lpdmVarData

Zeiger auf eine Struktur vom Typ DM_VARIABLE_DATA (Seite 237) mit den Informationen zu einer Variablen.

Ist die Struktur komplett mit 0 belegt, so war im DM_VARKEY ein falscher Name oder eine falsche ID angegeben. Abfragemöglichkeit z.B. über lpdmVarData->dmTypeRef->dwType.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMEnumVarData (Seite 325)	Informationen zu einer Variablen auflisten
---------------------------	--

Siehe auch

DM_VARKEY (Seite 241)

DM_VARIABLE_DATA (Seite 237)

DMEnumVarData (Seite 325)

2.2.6.3 DMEEnumVarData4 (RT Professional)

Verwendung

Die Funktion dient der Bestimmung von Daten über die mit einer Variablen verknüpften Objekte, z. B. zugehörige Variablengruppe oder Kanal-Unit. Sie unterscheidet sich von DMEEnumVarData durch die zusätzliche Ausgabe der Skalierungsdaten.

Deklaration

```

BOOL DMEEnumVarData4 (
    LPCSTR                lpszProjectFile,
    LPDM_VARKEY           lpdmVarKey,
    DWORD                dwItems,
    DM_ENUM_VARIABLE_PROC4 lpfnEnum,
    LPVOID               lpvUser,
    LPCMN_ERROR           lpdmError);

```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

lpdmVarKey

Zeiger auf den Anfang eines Feldes von Strukturen vom Typ DM_VARKEY (Seite 241) , über welche die aufzulistenden Variablen spezifiziert werden. Wenn der Parameter mit NULL versorgt wird, werden alle Variablen enumeriert.

Wird bei einem DM_VARKEY ein falscher Name oder eine falsche ID angegeben, so gibt es keine Fehleranzeige, da ja bei einer Liste manche gültig und manche fehlerhaft sein können. Damit der Anwender fehlerhafte DM_VARKEY erkennt, wird ihm für diese im Callback eine Struktur vom Typ DM_VARIABLE_DATA4 (Seite 239) übergeben, die komplett mit 0 belegt ist.

dwItems

Anzahl der Variablen, deren Daten aufgelistet werden sollen (=Anzahl der Variablenspezifikationen in lpdmVarKey). Wenn die Anzahl auf 0 gesetzt ist, werden alle Variablen enumeriert.

lpfnEnum

Zeiger auf Ihre Callback, welche die Informationen zu einer Variablen entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Informationen zur Variablen aufgelistet.

FALSE

Fehler.

Bemerkung

Hinweis

Bestehende Implementierungen einer mehrstufigen Enumeration von DMEnumVariables mit dem Aufruf von DMEnumVarDataX in der Callback für jedes einzelne Element sollten durch einen einzelnen Aufruf von DMEnumVarData4 ersetzt werden. Eventuell notwendige Filterungen sind hier in der Call-back selber durchzuführen. Die Umstellung verbessert die Performance erheblich.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden

Benötigte Dateien

- dmclient.h
- dmclient.lib
- dmclient.dll

Verwandte Funktionen

DM_ENUM_VARIABLE_PROC4 (Seite 331)	Informationen zu einer Variablen auflisten (Call-back)
------------------------------------	--

Beispiele

Enum Data of Tags (Seite 479) "DM01.cpp"

Siehe auch

DM_VARKEY (Seite 241)
 DM_VARIABLE_DATA4 (Seite 239)
 DM_ENUM_VARIABLE_PROC4 (Seite 331)
 Enum Data of Tags (Seite 479)

2.2.6.4 DM_ENUM_VARIABLE_PROC4 (RT Professional)**Beschreibung**

Um die vom System aufgelisteten Informationen zu einer Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_VARIABLE_PROC4 bereitstellen. Diese Callback-Funktion unterscheidet sich von DM_ENUM_VARIABLE_PROC durch die zusätzliche Ausgabe der Skalierungsdaten

Deklaration

```

BOOL ( * DM_ENUM_VARIABLE_PROC4 ) (
    LPDM_VARKEY          lpdmVarKey,
    LPDM_VARIABLE_DATA4 lpdmVarData,
    LPVOID               lpvUser
  )

```

Parameter**lpdmVarKey**

Zeiger auf eine Struktur vom Typ DM_VARKEY (Seite 241) zur Spezifikation der aufzulistenden Variablen.

lpdmVarData

Zeiger auf eine Struktur vom Typ DM_VARIABLE_DATA4 (Seite 239) mit den Informationen zu einer Variablen.

Ist die Struktur komplett mit 0 belegt, so war im DM_VARKEY ein falscher Name oder eine falsche ID angegeben. Abfragemöglichkeit z.B. über lpdmVarData->dmTypeRef->dwType.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert**TRUE**

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMLenumVarData4 (Seite 329)	Informationen zu einer Variablen auflisten
---	--

Beispiele

Enum Data of Tags (Seite 479) "DM01.cpp"

Siehe auch

- [DMLenumVarData4 \(Seite 329\)](#)
- [DM_VARKEY \(Seite 241\)](#)
- [DM_VARIABLE_DATA4 \(Seite 239\)](#)
- [Enum Data of Tags \(Seite 479\)](#)

2.2.6.5 DMLenumVarGrpData (RT Professional)

Verwendung

Die Funktion liefert die Informationen zu den Variablengruppen über die Callback-Funktion DM_ENUM_VARGRP_PROC, z. B. Name, CreatorID und Anzahl Variablen in der Gruppe.

Deklaration

```
BOOL DMEnumVarGrpData (
    LPSTR                lpszProjectFile,
    LPDM_VARGRPKEY      lpdmVarGrpKey,
    DWORD               dwItems,
    DM_ENUM_VARGRP_PROC lpfEnum,
    LPVOID              lpvUser,
    LPCMN_ERROR         lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

lpdmVarGrpKey

Zeiger auf die erste der Strukturen vom Typ DM_VARGRPKEY (Seite 236) , über welche die Variablengruppe spezifiziert wird.

dwItems

Anzahl der Variablengruppen, deren Daten aufgelistet werden.

lpfnEnum

Zeiger auf Ihre Callback-Funktion, welche die Informationen zu einer Variablengruppe entgegen nimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Informationen zu der Variablengruppe aufgelistet.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DM_ENUM_VARGRP_PROC (Seite 336)	Informationen zu einer Variablengruppe auflisten (Call-back)
---------------------------------	--

Beispiele

OnTestEnumGroupsAll (Seite 488) "TESTCDoc.cpp"

Siehe auch

DM_VARGRPKEY (Seite 236)
DM_ENUM_VARGRP_PROC (Seite 336)
OnTestEnumGroupsAll (Seite 488)

2.2.6.6 DMEEnumVarGrpDataExStr (RT Professional)

Deklaration

```
BOOL DMEEnumVarGrpDataExStr (
    LPTSTR          lpszProjectFile,
    DWORD           dwFlags,
    LPVARIANT       lpvdmVarGrpKey,
    LPDWORD         lpdwVarGrpCount,
    DM_ENUM_VARGRP_PROC_EXSTR lpfnEnum,
    LPVOID          lpvUser,
    LPCMN_ERROR     lpdmError);
```

Beschreibung

Die Funktion liefert die Informationen zu den Variablen-Gruppen über die Callback-Funktion "DM_ENUM_VARGRP_PROC_EXSTR", z. B. Name, ID und Anzahl Variablen in der Gruppe.

Parameter

IpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei kann mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermittelt werden.

dwFlags

Reserviert für eine zukünftige Nutzung. Ist mit "0L" zu besetzen.

IpvdmVarGrpKey

Zeiger auf die erste der Strukturen vom Typ "DM_VARGRPKEY" über die die Variablen-Gruppen spezifiziert wird.

Zeiger auf einen VARIANT für die Variablengruppenliste. Die Liste müssen Sie als "VT_ARRAY | VT_VARIANT" anlegen, da unterschiedliche Datentypen wie "VT_I4", "VT_BSTR" und bedingt auch "VT_LPSTR" gemischt angegeben werden. Oder Sie legen einen einfachen VARIANT an, wenn nur ein Key angegeben wird.

Wenn Sie hier "NULL" oder "VT_EMPTY" angeben, werden alle Gruppen enumeriert.

IpdwVarGrpCount

Zeiger auf "DWORD" für die Anzahl der Variablen-Gruppen.

Mit "IpfnCallback = NULL" wird so erstmals ermittelt, wieviel Speicher für die Variablen-Gruppen benötigt wird.

IpfnEnum

Zeiger auf die Callback-Funktion, die die Informationen zu einer Variablen-Gruppe entgegen nimmt.

IpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Informationen zu der Variablen-Gruppe aufgelistet

FALSE

Fehler

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden

Benötigte Files

dmclient_exstr.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DM_ENUM_VARGRP_PROC_EXSTR	Informationen zu einer Variablen-Gruppe auflisten (Callback)
---------------------------	--

2.2.6.7 DM_ENUM_VARGRP_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Informationen zu einer Variablengruppe auswerten zu können müssen Sie eine Callback-Funktion vom Typ DM_ENUM_VARGRP_PROC bereitstellen

Deklaration

```
BOOL ( * DM_ENUM_VARGRP_PROC) (  
    LPDM_VARGRP_DATA    lpdmVarGrpData,  
    LPVOID              lpvUser );
```

Parameter

lpdmVarGrpData

Zeiger auf eine Struktur vom Typ DM_VARGRP_DATA (Seite 235) mit den Informationen zur Variablengruppe.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMEnumVarGrpData (Seite 332)	Informationen zu einer Variablengruppe auflisten
------------------------------	--

Siehe auch

DMEnumVarGrpData (Seite 332)

DM_VARGRP_DATA (Seite 235)

2.2.6.8 DMEnumVariables (RT Professional)

Verwendung

Mit der Funktion werden die Namen aller Variablen aufgelistet, die einem bestimmten Selektionskriterium entsprechen. Das Selektionskriterium wird über die Struktur DM_VARFILTER festgelegt.

Deklaration

```
BOOL DMEnumVariables (  
    LPCSTR          lpszProjectFile,  
    LPDM_VARFILTER lpdmVarFilter,  
    DM_ENUM_VAR_PROC lpfnEnum,  
    LPVOID          lpvUser,  
    LPCMN_ERROR     lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.
Der Name der Projektdatei wird mit `DMEnumOpenedProjects` oder in RT mit `DMGetRuntimeProject` ermittelt.

lpdmVarFilter

Zeiger auf die Struktur `DM_VARFILTER` (Seite 233) , in welcher die Selektionskriterien festgelegt werden.

lpfnEnum

Zeiger auf die Callback-Funktion, welche die Variablennamen entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablennamen aufgelistet.

FALSE

Fehler.

Bemerkung

Hinweis

Bestehende Implementierungen einer mehrstufigen Enumeration von DMEnumVariables mit dem Aufruf von DMEnumVarDataX in der Callback für jedes einzelne Element sollten durch einen einzelnen Aufruf von DMEnumVarData4 ersetzt werden. Eventuell notwendige Filterungen sind hier in der Callback selber durchzuführen. Die Umstellung verbessert die Performance erheblich.

Benötigte Dateien

dmclient.h
 dmclient.lib
 dmclient.dll

Verwandte Funktionen

DM_ENUM_VAR_PROC (Seite 339)	Variablennamen auflisten (Callback)
------------------------------	-------------------------------------

Beispiele

OnTestEnumVariables (Seite 489) "TESTCDoc.cpp"

Siehe auch

DM_VARFILTER (Seite 233)
 DM_ENUM_VAR_PROC (Seite 339)
 OnTestEnumVariables (Seite 489)

2.2.6.9 DM_ENUM_VAR_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Namen von Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_VAR_PROC bereitstellen.

Deklaration

```

BOOL ( * DM_ENUM_VAR_PROC) (
    LPDM_VARKEY    lpdmVarKey,
    LPVOID         lpvUser
  )

```

Parameter

lpdmVarKey

Zeiger auf eine Struktur vom Typ DM_VARKEY (Seite 241) mit dem Namen einer Variablen.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Weitere Callback-Aufrufe sind erwünscht.

FALSE

Die Callback Serie soll abgebrochen werden.

Bemerkung

ACHTUNG
Sie dürfen keine andere DMClient-Funktion in dieser Callback aufrufen, da es sonst zu einer Blockade kommen kann.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMEnumVariables (Seite 337)	Variablenamen auflisten
-----------------------------	-------------------------

Siehe auch

DM_VARKEY (Seite 241)

DMEnumVariables (Seite 337)

2.2.6.10 DMGetValue (RT Professional)

Verwendung

Liest den Wert einer oder mehrerer Variablen aus dem Prozessabbild des Datenmanagers. Dabei wird der Wert ausgelesen, der zum Zeitpunkt der letzten Aktualisierung vorgelegen hat.

Die weiter entwickelte Funktion `DMGetValueEx`, liefert in der Struktur `DM_VAR_UPDATE_STRUCTEX` zusätzlich den Quality Code.

Deklaration

```
BOOL DMGetValue (
    LPDM_VARKEY          lpdmVarKey,
    DWORD               dwItems,
    LPDM_VAR_UPDATE_STRUCT lpdmvus,
    LPCMN_ERROR         lpdmError);
```

Parameter

lpdmVarKey

Zeiger auf die erste der Strukturen `DM_VARKEY` (Seite 241), welche die zu lesenden Variablenwerte kennzeichnen.

dwItems

Anzahl der übergebenen Strukturen (entspricht der Anzahl der zu lesenden Variablenwerte).

lpdmvus

Zeiger auf die erste der Strukturen `DM_VAR_UPDATE_STRUCT` (Seite 228), die nach Rückkehr der Funktion die Variablenwerte enthalten.

lpdmError

Zeiger auf die erste von `dwItems` Fehlerstrukturen vom Typ `CMN_ERROR`. Wenn beim Schreiben einer Variablen ein Fehler auftritt, so schreibt das System die Fehlerinformation in die entsprechende Struktur. Vergessen Sie daher nicht, Platz für diese Strukturen zu reservieren.

Rückgabewert

TRUE

Werte ermittelt.

FALSE

Fehler.

Bemerkung

Um z. B. einer Applikation die Möglichkeit einer eigenen Zyklusverwaltung zu geben, wird bei DMStartVarUpdate ein NULL Zeiger als Callback-Funktion angegeben. Daraufhin übernimmt der Datenmanager die Aktualisierung der Variablen in dem geforderten Zyklus, überlässt jedoch der Applikation die Verantwortung über das Timing des Auslesens aus dem Prozessabbild.

Da Aktualisierungsanforderungen für eine Variable von mehreren Applikationen in unterschiedlichen Zyklen vorliegen können, legt der Datenmanager zum Update seines Prozessabbilds immer den kleinsten geforderten Zyklus zu Grunde.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	Speichermangel

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMGetValueWait (Seite 353)	Variablenwerte mit Aktualisierung ermitteln
DMSetValue (Seite 385)	Variablenwerte ändern

Beispiele

Read tag (Seite 515) "DM02.cpp"

OnTestVariablenGetValue (Seite 499) "TESTCDoc.cpp"

Siehe auch

DM_VARKEY (Seite 241)

DM_VAR_UPDATE_STRUCT (Seite 228)

DMGetValueWait (Seite 353)

DMSetValue (Seite 385)

Read tag (Seite 515)

OnTestVariablenGetValue (Seite 499)

2.2.6.11 DMGetValueEx (RT Professional)

Verwendung

Liest den Wert einer oder mehrerer Variablen aus dem Prozessabbild des Datenmanagers. Dabei wird der Wert ausgelesen, der zum Zeitpunkt der letzten Aktualisierung vorgelegen hat.

Im Gegensatz zu DMGetValue wird in der Struktur DM_VAR_UPDATE_STRUCTEX auch der Quality Code zurückgegeben.

Deklaration

```
BOOL DMGetValueEx (
    LPDM_VARKEY          lpdmVarKey,
    DWORD                dwItems,
    LPDM_VAR_UPDATE_STRUCTEX lpdmvus,
    LPCMN_ERROR          lpdmError);
```

Parameter

lpdmVarKey

Zeiger auf die erste der Strukturen DM_VARKEY (Seite 241) , welche die zu lesenden Variablenwerte kennzeichnen.

dwItems

Anzahl der übergebenen Strukturen (entspricht der Anzahl der zu lesenden Variablenwerte).

lpdmvus

Zeiger auf die erste der Strukturen DM_VAR_UPDATE_STRUCTEX (Seite 230) , die nach Rückkehr der Funktion die Variablenwerte enthalten.

lpdmError

Zeiger auf die erste von dwItems Fehlerstrukturen vom Typ CMN_ERROR. Wenn beim Schreiben einer Variablen ein Fehler auftritt, so schreibt das System die Fehlerinformation in die entsprechende Struktur. Vergessen Sie daher nicht, Platz für diese Strukturen zu reservieren.

Rückgabewert

TRUE

Werte ermittelt.

FALSE

Fehler.

Bemerkung

Um z. B. einer Applikation die Möglichkeit einer eigenen Zyklusverwaltung zu geben, kann bei DMStartVarUpdate ein NULL Zeiger als Callback-Funktion angegeben werden. Daraufhin übernimmt der Datenmanager die Aktualisierung der Variablen in dem geforderten Zyklus, überlässt jedoch der Applikation die Verantwortung über das Timing des Auslesens aus dem Prozessabbild.

Da Aktualisierungsanforderungen für eine Variable von mehreren Applikationen in unterschiedlichen Zyklen vorliegen können, legt der Datenmanager zum Update seines Prozessabbilds immer den kleinsten geforderten Zyklus zu Grunde.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DMGetValueWait (Seite 353)	Variablenwerte mit Aktualisierung ermitteln
DMSetValue (Seite 385)	Variablenwerte ändern

Siehe auch

DM_NOTIFY_VARIABLEEX_PROC (Seite 469)
DM_VAR_UPDATE_STRUCTEX (Seite 230)
DM_VARKEY (Seite 241)
DMGetValueWait (Seite 353)
DMSetValue (Seite 385)
DMGetValueWaitEx (Seite 356)

2.2.6.12 DMGetValueExStr (RT Professional)

Deklaration

```
BOOL DMGetValueExStr (
    DWORD          dwFlags,
    LPVARIANT      lpdmVarKey,
```

```

LPDM_VAR_UPDATE_STRUCT_EXSTR    lpdmvus,
DWORD                            dwdmvusCount,
LPCMN_ERROR                      lpdmError);

```

Beschreibung

Liest den Wert einer oder mehrerer Variablen aus dem Prozessabbild des Datenmanagers. Dabei wird der Wert ausgelesen, der zum Zeitpunkt der letzten Aktualisierung vorgelegen hat.

Im Gegensatz zu "DMGetValueEx" wird nicht mehr die Struktur "DM_VARKEY" verwendet, damit keine Längenbegrenzung beim Variablennamen vorliegt.

Parameter

dmFlags

Wenn eine Rückgabe des Variablennamens im VARIANT der Struktur "DM_VAR_UPDATE_STRUCT_EXSTR" im Format "VT_LPSTR" benötigt wird, kann hier das Flag "DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR" angegeben werden.

lpvdmVarKey

Zeiger auf einen VARIANT für die Variablenliste. Die Liste müssen Sie als "VT_ARRAY | VT_VARIANT" anlegen, da unterschiedliche Datentypen wie "VT_I4", "VT_BSTR" und bedingt auch "VT_LPSTR" vorhanden sein können. Das gilt auch für einen einzelnen Key.

lpdmvus

Zeiger auf die erste der Strukturen "DM_VAR_UPDATE_STRUCT_EXSTR", die nach Rückkehr der Funktion die Werte der Variablen enthalten.

dwdmvusCount

Anzahl der übergebenen Strukturen "DM_VAR_UPDATE_STRUCT_EXSTR", entspricht der Anzahl der zu lesenden Variablenwerte. Die Anzahl muss mit der Größe des Variant-Array in "lpvdmVarKey" übereinstimmen.

lpdmError

Zeiger auf die erste von "dwdmvusCount" Fehlerstrukturen vom Typ "CMN_ERROR". Tritt beim Schreiben einer Variablen ein Fehler auf, so schreibt das System die Fehlerinformation in die entsprechende Struktur. Vergessen Sie daher nicht, Platz für diese Strukturen zu reservieren.

Rückgabewert

TRUE

Werte ermittelt

FALSE

Fehler

Bemerkung

Um z. B. einer Applikation die Möglichkeit einer eigenen Zyklusverwaltung zu geben, kann bei "DMStartVarUpdate" ein "NULL"-Zeiger als Callback-Funktion angegeben werden. Daraufhin übernimmt der Datenmanager die Aktualisierung der Variablen in dem geforderten Zyklus, überlässt jedoch der Applikation die Verantwortung über das Timing des Auslesens aus dem Prozessabbild.

Da Aktualisierungsanforderungen für eine Variable von mehreren Applikationen in unterschiedlichen Zyklen vorliegen können, legt der Datenmanager zum Update seines Prozessabbilds immer den kleinsten geforderten Zyklus zugrunde.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Diese Fehler werden nur in der ersten Fehlerstruktur zurückgeliefert.

Benötigte Files

- dmclient_exstr.h
- dmclient.lib
- dmclient.dll

Verwandte Funktionen

DMGetValueWaitExStr	Variablenwerte mit Aktualisierung ermitteln
DMSetValueExStr	Variablenwerte ändern

Beispiele

Skript-Beispiel "Button DMGetValueExStr/DMS SetValueExStr|DMGetValueWaitExStr/DM SetValueWaitExStr":

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
#pragma code ("kernel32.dll")
#define CP_ACP 0
int MultiByteToWideChar(UINT CodePage, DWORD dwFlags, LPCSTR lpMultiByteStr, int cbMultiByte, LPWSTR lpWideCharStr, int cchWideChar);
int WideCharToMultiByte(UINT CodePage, DWORD dwFlags, LPCWSTR lpWideCharStr, int cchWideChar, LPSTR lpMultiByteStr, int cbMultiByte, LPCSTR lpDefaultChar, LPBOOL lpUsedDefaultChar);
#pragma code ()
```

```

#pragma code ("OleAut32.dll")
// #include "OleAuto.h"
SAFEARRAY * SafeArrayCreateVector(VARTYPE vt, long lLbound, unsigned
int cElements );
HRESULT SafeArrayPtrOfIndex(SAFEARRAY FAR* psa, long FAR* rgIndices,
void HUGE_PTR FAR* ppvData );
HRESULT SafeArrayLock(SAFEARRAY FAR* psa);
HRESULT SafeArrayUnlock(SAFEARRAY FAR* psa);
HRESULT VariantChangeType( VARIANTARG FAR* pvargDest, VARIANTARG
FAR* pvarSrc, unsigned short wFlags, VARTYPE vt);
typedef WCHAR OLECHAR;
BSTR SysAllocString(OLECHAR FAR* sz);
HRESULT SysFreeString(BSTR bstr);
#pragma code()

extern BOOL DM_NotifyVariableProcExStr_GetValueWait(DWORD dwTAID,
LPDM_VAR_UPDATE_STRUCT_EXSTR lpdmvus, DWORD dwItems, LPVOID lpvUser);
extern BOOL DM_CompletionProc(DWORD dwTAID, LPDWORD lpdmVarState,
DWORD dwItems, LPVOID lpvUser);

VARIANT vVarKey;
VARIANT* pvElem;
VARIANT vdmValue;
SAFEARRAY* parrayKeys;
SAFEARRAY* parrayValues;
DWORD dwVal[4];
DWORD dwState[4];
DWORD dwMerk;
CMN_ERROR err;
CMN_ERROR errArray[4];
DM_VAR_UPDATE_STRUCT_EXSTR dmvus[4];
HRESULT hr;
long lInx;
BOOL bRet;
DWORD dwFlags;
DWORD* pdwVarState;
DWORD dwTAID;

memset(&err, 0, sizeof(err));
memset(dmvus, 0, sizeof(dmvus));
dwVal[0] = 1;
dwVal[1] = 2;
dwVal[2] = 3;
dwVal[3] = 4;
dwMerk = 0L;
memset(dwState, 0, sizeof(dwState));
pvElem = NULL;
parrayKeys = NULL;
parrayValues = NULL;
hr = 0L; //S_OK
lInx = 0L;
bRet = FALSE;

```



```

    }
    else
    {
        if (dmvus[i].vdmValue.vt == VT_I4)
        {
            dwVal[i] = dmvus[i].vdmValue.u.lVal;
            printf("\r\n    Var[%d]:[%s] = %ld",
                i, dmvus[i].vdmVarKey.u.pbVal,
dmvus[i].vdmValue.u.lVal);
        }
        else
        {
            printf("\r\n    wrong datatype %ld, VT_I4 expected after
VariantChange",
                dmvus[i].vdmValue.vt == VT_I4);
        }
    }
}

//clear the internal allocated Data, that no memoryleaks
//the vdmValues no need to clear here, while no PROPVARIANT here
given (double used)
//only the dmVarkey of dmvus have to free here
for (i = 0; i < 4; i++)
{
    free(dmvus[i].vdmVarKey.u.pbVal);
    dmvus[i].vdmVarKey.u.pbVal = NULL;
    dmvus[i].vdmVarKey.vt = VT_EMPTY;
}
}

// change the values before set again
dwMerk = dwVal[0];
dwVal[0] = dwVal[3];
dwVal[3] = dwMerk;
dwMerk = dwVal[1];
dwVal[1] = dwVal[2];
dwVal[2] = dwMerk;

VariantInit(&vdmValue);
parrayValues = SafeArrayCreateVector(VT_VARIANT, 0L, 4);
vdmValue.vt = VT_ARRAY | VT_VARIANT;
vdmValue.u.parray = parrayValues;

SafeArrayLock(parrayValues);
lInx = 0L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;
pvElem->u.pbVal = dwVal[0];
lInx = 1L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;

```

```

pvElem->u.pbVal = dwVal[1];
lInx = 2L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;
pvElem->u.pbVal = dwVal[2];
lInx = 3L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;
pvElem->u.pbVal = dwVal[3];
SafeArrayUnlock(parrayValues);

printf("\r\n call DMSetValueExStr:");
memset(&err, 0, sizeof(err));
bRet = DMSetValueExStr(&vVarKey, &vdmValue, pdwVarState, &err);

if (bRet == FALSE)
{
    printf("\r\n error DMSetValueExStr, err=%ld,%ld,%ld,%ld,%ld,
[%s]",
           err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}
else
{
    int i = 0;
    for (i = 0; i < 4; i++)
    {
        printf("\r\n dwVarState[%d] = %ld", i, dwState[i]);
    }
}

// clear the values before read back again with DMGetValueWait
(timeout 2sec.)
dwVal[0] = 0;
dwVal[1] = 0;
dwVal[2] = 0;
dwVal[3] = 0;

printf("\r\n call DMGetValueWaitExStr:");
memset(&err, 0, sizeof(err));
bRet = DMGetValueWaitExStr(&dwTAID, dwFlags, &vVarKey, TRUE, 2000,
DM_NotifyVariableProcExStr_GetValueWait, dwVal, &err);
if (bRet == FALSE)
{
    printf("\r\n error DMGetValueWaitExStr, err=%ld,%ld,%ld,%ld,%ld,
[%s]",
           err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}
else
{
    int i = 0;

```

```

    for (i = 0; i < 4; i++)
    {
        printf("\r\n    dwValue[%d] = %ld", i, dwVal[i]);
    }
}

//increment values for identify execute of following DMSetValueWait
dwVal[0] = dwVal[0] + 1;
if (dwVal[0] >= 100)
{
    dwVal[0] = dwVal[0] - 100;
}
dwVal[1] = dwVal[1] + 1;
if (dwVal[1] >= 100)
{
    dwVal[1] = dwVal[1] - 100;
}
dwVal[2] = dwVal[2] + 1;
if (dwVal[2] >= 100)
{
    dwVal[2] = dwVal[2] - 100;
}
dwVal[3] = dwVal[3] + 1;
if (dwVal[3] >= 100)
{
    dwVal[3] = dwVal[3] - 100;
}
VariantClear(&vdmValue);
parrayValues = SafeArrayCreateVector(VT_VARIANT, 0L, 4);
vdmValue.vt = VT_ARRAY | VT_VARIANT;
vdmValue.u.parray = parrayValues;
SafeArrayLock(parrayValues);
lInx = 0L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;
pvElem->u.pbVal = dwVal[0];
lInx = 1L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;
pvElem->u.pbVal = dwVal[1];
lInx = 2L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;
pvElem->u.pbVal = dwVal[2];
lInx = 3L;
hr = SafeArrayPtrOfIndex(parrayValues, &lInx, &pvElem);
pvElem->vt = VT_I4;
pvElem->u.pbVal = dwVal[3];
SafeArrayUnlock(parrayValues);

printf("\r\ncall DMSetValueWaitExStr:");
memset(errArray, 0, sizeof(errArray));

```

```

bRet = DMSetValueWaitExStr(&dwTAID, &vVarKey, 4, &vdmValue,
pdwVarState, 2000, DM_CompletionProc, NULL, errArray);
if (bRet == FALSE)
{
    int i = 0;
    for(i=0; i < 4; i++)
    {
        printf("\r\n    error DMSetValueWaitExStr, errArray[%d]=%ld,
%ld,%ld,%ld,%ld,[%s]",
            i, errArray[i].dwError1, errArray[i].dwError2,
errArray[i].dwError3,
            errArray[i].dwError4, errArray[i].dwError5,
errArray[i].szErrorText);
    }
}
else
{
}

printf("\r\n##### exit Test with DMGetValue(Wait)ExStr/
DMSetValue(Wait)ExStr #####\r\n");

}

```

Skript-Beispiel "Projekt-Funktion DM_NotifyVariableProcExStr_GetValueWait":

```

#pragma code ("OleAut32.dll")
// #include "OleAuto.h"
HRESULT VariantChangeType( VARIANTARG FAR* pvargDest, VARIANTARG
FAR* pvarSrc, unsigned short wFlags, VARTYPE vt);
#pragma code()

BOOL DM_NotifyVariableProcExStr_GetValueWait(DWORD dwTAID,
LPDM_VAR_UPDATE_STRUCT_EXSTR lpdmvus, DWORD dwItems, LPVOID lpvUser)
{
    BOOL bRet = FALSE;
    int i = 0;
    DWORD* pdwVal = NULL;

    pdwVal = (DWORD*)lpvUser;
    // the DMGetValueWait used PROPVARIANT and the lpvUser is ptr to
    OutDataArray (DWORDS)
    printf("\r\n*** DM_NotifyVariableProcExStr_GetValueWait entry ***");

    for (i = 0; i < dwItems; i++)
    {
        VariantChangeType( (VARIANTARG*)&(lpdmvus[i].vdmValue),
(VARIANTARG*)&(lpdmvus[i].vdmValue), 0, VT_I4);
        if (0 == strcmp((LPCSTR)lpdmvus[i].vdmVarKey.u.pbVal, "dwVal_1"))
        {
            pdwVal[0] = lpdmvus[i].vdmValue.u.lVal;
            printf("\r\n    set dwVal_1=%ld to dwVal[0]", pdwVal[0]);
            bRet = TRUE;
        }
    }
}

```

```

    }
    if (0 == strcmp((LPCSTR)lpdmvus[i].vdmVarKey.u.pbVal, "dwVal_2"))
    {
        pdwVal[1] = lpdmvus[i].vdmValue.u.lVal;
        printf("\r\n  set dwVal_2=%ld to dwVal[1]", pdwVal[1]);
        bRet = TRUE;
    }
    if (0 == strcmp((LPCSTR)lpdmvus[i].vdmVarKey.u.pbVal, "dwVal_3"))
    {
        pdwVal[2] = lpdmvus[i].vdmValue.u.lVal;
        printf("\r\n  set dwVal_3=%ld to dwVal[2]", pdwVal[2]);
        bRet = TRUE;
    }
    if (0 == strcmp((LPCSTR)lpdmvus[i].vdmVarKey.u.pbVal, "dwVal_4"))
    {
        pdwVal[3] = lpdmvus[i].vdmValue.u.lVal;
        printf("\r\n  set dwVal_4=%ld to dwVal[3]", pdwVal[3]);
        bRet = TRUE;
    }
}

printf("\r\n*** DM_NotifyVariableProcExStr_GetValueWait exit ***");
return bRet;
}

Project-Function DM_CompletionProc:

BOOL DM_CompletionProc(DWORD dwTAID, LPDWORD lpdmVarState, DWORD
dwItems, LPVOID lpvUser)
{
    BOOL bRet = FALSE;
    int i = 0;

    printf("\r\n*** DM_CompletionProc entry ***");

    for (i = 0; i < dwItems; i++)
    {
        printf("\r\n  VarState[%d] = %ld", i, lpdmVarState[i]);
    }

    printf("\r\n*** DM_CompletionProc exit ***");

    return bRet;
}

```

2.2.6.13 DMGetValueWait (RT Professional)

Verwendung

Liest den Wert einer oder mehrerer Variablen aus dem Prozessabbild des Datenmanagers. Dabei wird der Wert ausgelesen, der zum Zeitpunkt der letzten Aktualisierung vorgelegen hat.

Diese Funktion erzwingt im Unterschied zu DMGetValue eine wertemäßige Aktualisierung der Variablenwerte.

Die weiter entwickelte Funktion DMGetValueWaitEx, liefert in der Struktur DM_VAR_UPDATE_STRUCTEX in der Callback DM_NOTIFY_VARIABLESEX_PROC zusätzlich den Quality Code.

Deklaration

```
BOOL DMGetValueWait (
    LPDWORD                pdwTAID,
    LPDM_VARKEY            lpdmVarKey,
    DWORD                  dwItems,
    BOOL                    fWaitForCompletion,
    DWORD                  dwTimeout,
    DM_NOTIFY_VARIABLESEX_PROC lpfnVariable,
    LPVOID                  lpvUser,
    LPCMN_ERROR             lpdmError);
```

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion die vom Datenmanager vergebene Transaktions-ID enthält.

lpdmVarKey

Zeiger auf die erste der Strukturen DM_VARKEY (Seite 241) , welche die zu lesenden Variablenwerte kennzeichnen.

dwItems

Anzahl der übergebenen Strukturen (entspricht der Anzahl der zu lesenden Variablenwerte).

fWaitForCompletion

Wenn das Flag fWaitForCompletion gesetzt ist, wird die übergebene Callback-Funktion erst dann aufgerufen, wenn entweder alle angeforderten Variablen aktualisiert sind, oder die angegebene Timeout-Zeit überschritten wurde.

Ist fWaitForCompletion nicht gesetzt wird die übergebene Callback Funktion unmittelbar aufgerufen. Das Zeitverhalten entspricht dann dem der Funktion DMGetValue.

dwTimeout

Maximale Wartezeit der Applikation in ms. Wenn nach Ablauf der Wartezeit nicht alle Variablenwerte aktualisiert sind, wird die Callback-Funktion mit entsprechenden Fehlercodes aufgerufen.

Ist das Flag fWaitForCompletion == FALSE, so wird der übergebene Wert nicht ausgewertet.

IpfnVariable

Zeiger auf Ihre Callback-Funktion DM_NOTIFY_VARIABLE_PROC, die entweder nach der Aktualisierung aller angeforderten Variablen oder nach Ablauf der Wartezeit aufgerufen wird.

Ist fWaitForCompletion == FALSE, so wird die Callback Funktion sofort mit den momentan im Prozessabbild des Datenmanagers stehenden Werten aufgerufen.

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

IpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Werte ermittelt.

FALSE

Fehler.

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_NOTIFY_VARIABLE_PROC (Seite 458)	Variablenwerte mit Aktualisierung ermitteln (Callback)
DMGetValue (Seite 341)	Variablenwerte ermitteln
DMGetValueWaitEx (Seite 356)	Variablenwerte mit Aktualisierung ermitteln
AUTOHOTSPOT	Variablenwerte mit Benachrichtigung ändern

Beispiele

OnTestVariablenGetvaluwait (Seite 502) "TESTCDoc.cpp"

Siehe auch

- DM_NOTIFY_VARIABLE_PROC (Seite 458)
- DM_VARKEY (Seite 241)
- DMGetValue (Seite 341)
- DMGetValueEx (Seite 343)
- DMSetValueWait (Seite 394)
- OnTestVariablenGetvaluawait (Seite 502)
- DMGetValueWaitEx (Seite 356)

2.2.6.14 DMGetValueWaitEx (RT Professional)

Verwendung

Liest den Wert einer oder mehrerer Variablen aus dem Prozessabbild des Datenmanagers. Dabei wird der Wert ausgelesen, der zum Zeitpunkt der letzten Aktualisierung vorgelegen hat. Im Unterschied zu DMGetValueEx wird mit dieser Funktion eine wertemäßige Aktualisierung der Variablenwerte erzwungen.

Deklaration

```
BOOL DMGetValueWaitEx (  
    LPDWORD                pdwTAID,  
    LPDM_VARKEY            lpdmVarKey,  
    DWORD                  dwItems,  
    BOOL                    fWaitForCompletion,  
    DWORD                  dwTimeOut,  
    DM_NOTIFY_VARIABLEEX_PROC lpfnVariable,  
    LPVOID                  lpvUser,  
    LPCMN_ERROR             lpdmError);
```

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion die vom Datenmanager vergebene Transaktions-ID enthält.

lpdmVarKey

Zeiger auf die erste der Strukturen DM_VARKEY (Seite 241) , welche die zu lesenden Variablenwerte kennzeichnen.

dwItems

Anzahl der übergebenen Strukturen (entspricht der Anzahl der zu lesenden Variablenwerte).

fWaitForCompletion

Wenn das Flag `fWaitForCompletion` gesetzt ist, wird die übergebene Callback-Funktion erst dann aufgerufen, wenn entweder alle angeforderten Variablen aktualisiert sind, oder die angegebene Timeout-Zeit überschritten wurde.

Wenn `fWaitForCompletion` nicht gesetzt ist, wird die übergebene Callback Funktion unmittelbar aufgerufen. Das Zeitverhalten entspricht dann dem der Funktion `DMGetValue`.

dwTimeout

Maximale Wartezeit der Applikation in ms. Wenn nach Ablauf der Wartezeit nicht alle Variablenwerte aktualisiert sind, wird die Callback-Funktion mit entsprechenden Fehlercodes aufgerufen.

Ist das Flag `fWaitForCompletion == FALSE`, so wird der übergebene Wert nicht ausgewertet.

lpfnVariable

Zeiger auf Ihre Callback-Funktion `DM_NOTIFY_VARIABLEEX_PROC`, die entweder nach der Aktualisierung aller angeforderten Variablen oder nach Ablauf der Wartezeit aufgerufen wird.

Ist `fWaitForCompletion == FALSE`, so wird die Callback Funktion sofort mit den momentan im Prozessabbild des Datenmanagers stehenden Werten aufgerufen.

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Werte ermittelt.

FALSE

Fehler.

Verwandte Funktionen

DM_NOTIFY_VARIABLEEX_PROC (Seite 469)	Variablenwerte mit Benachrichtigung ermitteln (Callback)
DMGetValueEx (Seite 343)	Variablenwerte ermitteln
DMSetValueWait (Seite 394)	Variablenwerte mit Benachrichtigung ändern

Siehe auch

DM_NOTIFY_VARIABLEEX_PROC (Seite 469)

DMGetValueWait (Seite 353)

DM_VARKEY (Seite 241)

DMGetValueEx (Seite 343)

DMSetValueWait (Seite 394)

2.2.6.15 DMGetValueWaitExStr (RT Professional)

Deklaration

```
BOOL DMGetValueWaitExStr (
    LPDWORD                pdwTAID,
    DWORD                  dwFlags,
    LPVARIANT              lpvdmVarKey,
    LPVARIANT              lpvCookie,
    BOOL                   fWaitForCompletion,
    DWORD                  dwTimeout,
    DM_NOTIFY_VARIABLE_PROC_EXSTR lpfnVariable,
    LPVOID                 lpvUser,
    LPCMN_ERROR            lpdmError);
```

Beschreibung

Liest den Wert einer oder mehrerer Variablen aus dem Prozessabbild des Datenmanagers. Dabei wird der Wert ausgelesen, der zum Zeitpunkt der letzten Aktualisierung vorgelegen hat.

Im Unterschied zu "DMGetValueExStr" kann mit dieser Funktion eine wertemäßige Aktualisierung der Variablenwerte erzwungen werden.

Im Gegensatz zu "DMGetValueEx" wird nicht mehr die Struktur "DM_VARKEY" verwendet, damit keine Längenbegrenzung beim Variablennamen vorliegt.

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion, die vom Datenmanager vergebene Transaktions-ID enthält.

dwFlags

Wenn eine Rückgabe des Variablennamen im VARIANT der Struktur "DM_VAR_UPDATE_STRUCT_EXSTR" der "DM_NOTIFY_VARIABLE_PROC_EXSTR" im Format "VT_LPSTR" benötigt wird, kann hier das Flag "DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR" angegeben werden.

lpvdmVarKey

Zeiger auf einen VARIANT für die Variablenliste. Die Liste müssen Sie als "VT_ARRAY | VT_VARIANT" anlegen, da unterschiedliche Datentypen wie "VT_I4", "VT_BSTR" und bedingt auch "VT_LPSTR" vorhanden sein können.

lpvCookie

Zeiger auf einen VARIANT für ein zusätzliche Liste benutzerabhängiger Daten zu jeder Variable.

Das ist als Ersatz für "lpvUserData" des vorherigen "DM_VARKEY" gedacht und wird in der Struktur "DM_VAR_UPDATE_STRUCT_EXSTR" für jede Variable mit zurückgegeben.

fWaitForCompletion

Wenn "fWaitForCompletion" gesetzt ist, wird die übergebene Callback-Funktion erst aufgerufen, wenn entweder alle angeforderten Variablen aktualisiert sind, oder die angegebene Timeout-Zeit überschritten wurde.

Wenn "fWaitForCompletion" nicht gesetzt ist, wird die übergebene Callback-Funktion unmittelbar aufgerufen. Das Zeitverhalten entspricht dann der Funktion "DMGetValueExStr".

dwTimeout

Maximale Wartezeit der Applikation in "ms". Wenn nach Ablauf der Wartezeit nicht alle Variablenwerte aktualisiert sind, wird die Callback-Funktion mit entsprechenden Fehlercodes aufgerufen.

Wenn "fWaitForCompletion == FALSE", wird der übergebene Wert nicht ausgewertet.

lpfnVariable

Zeiger auf die Callback-Funktion "DM_NOTIFY_VARIABLEEX_PROC_EXSTR", die entweder nach der Aktualisierung aller angeforderten Variablen oder nach Ablauf der Wartezeit aufgerufen wird.

Wenn "fWaitForCompletion == FALSE", wird die Callback-Funktion sofort mit den momentan im Prozessabbild des Datenmanagers stehenden Werten aufgerufen.

Wenn ein Programm eine Notify-Routine anmeldet, muss das Programm regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen wird bereits ein Notify geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

lpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Werte ermittelt

FALSE

Fehler

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMGetValueExStr	Variablenwerte ermitteln
DMSetValueWaitExStr	Variablenwerte mit Benachrichtigung ändern

Beispiele

Beispiele finden Sie auf der Seite "DMGetValueExStr".

2.2.6.16 DMGetVarInfo (RT Professional)

Verwendung

DMGetVarInfo ermittelt den vollständigen Variablenschlüssel einer Variablen. Mithilfe dieses Befehls ist es möglich, zu einer Variablen-ID den Variablennamen und umgekehrt zu erhalten.

Deklaration

```
BOOL DMGetVarInfo (
    LPCSTR      lpszProjectFile,
    LPDM_VARKEY lpdmVarKey,
    DWORD       dwItems,
    LPCMN_ERROR lpdmError);
```

Parameter

IpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

IpdmVarKey

Zeiger auf die Struktur DM_VARKEY (Seite 241) mit den Variablenschlüsseln, die vervollständigt werden.

dwItems

Anzahl der übergebenen Strukturen.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablenschlüssel ermittelt.

FALSE

Fehler.

Bemerkung

Verwenden Sie den Aufruf nicht im Wechsel mit Aufrufen zum Erzeugen von Variablen. Wenn DMGetVarInfo nach dem Erzeugen einer Variablen zum ersten Mal verwendet wird, werden Suchlisten neu sortiert, was zusätzliche Laufzeit kostet.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Beispiele

OnTestVariablenGetVarInfo (Seite 506) "TESTCDoc.cpp"

Siehe auch

DM_VARKEY (Seite 241)

OnTestVariablenGetVarInfo (Seite 506)

2.2.6.17 DMGetVarInfoExStr (RT Professional)

Deklaration

```
BOOL DMGetVarInfoExStr (  
    LPCTSTR          lpszProjectFile,  
    DWORD            dwFlags,  
    LPVARIANT         lpvdmVarKeyIn,  
    LPVARIANT         lpvdmVarKeyOut,  
    LPCMN_ERROR       lpdmError);
```

Beschreibung

Ermittelt den vollständigen Variablenschlüssel einer Variablen. Damit erhalten Sie auch zu einer Variablen-ID den Variablennamen und umgekehrt.

Im Gegensatz zu "DMGetVarInfo" wird die Variablenliste als Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" übergeben. Die Rückgabe erfolgt in einem zweiten VARIANT. Damit entfällt die Begrenzung der Länge der Variablennamen.

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei können Sie mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermitteln.

dwFlags

Wenn die Rückgabe des Variablennamen im VARIANT "lpvdmVarKeyOut" im Format "VT_LPSTR" benötigt wird, kann hier "DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR" angegeben werden.

lpvdmVarKeyIn

Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" zur Übergabe der Variablenliste oder einfachen VARIANT, wenn nur eine Variable angegeben wird.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

lpvdmVarKeyOut

Pointer auf VARIANT, über den die korrespondierenden Werte zu "lpvdmVarKeyIn" zurückgeliefert werden können.

Zu einem vorgegebenen Variablennamen in "lpvdmVarKeyIn" wird dann die VariablenID geliefert oder umgekehrt.

Wenn der VARIANT mit "VT_EMPTY" initialisiert ist, wird ein entsprechendes Array mit "VT_ARRAY | VT_VARIANT" mit der gleichen Größe wie in "lpvdmVarKeyIn" angelegt und befüllt.

Wenn der VARIANT nicht leer ist, wird zuerst ein "VariantClear" aufgerufen, bevor neu angelegt wird. Wenn der VARIANT nicht initialisiert ist, kann das zu einer Exception führen.

Wenn die Variable nicht vorhanden ist oder auf die Variable nicht zugegriffen werden kann, bleibt der Typ auf "VT_EMPTY" für den betreffenden Eintrag stehen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablenschlüssel ermittelt

FALSE

Fehler

Bemerkung

Der Aufruf dürfen Sie nicht im Wechsel mit Aufrufen zum Erzeugen von Variablen verwenden. Wenn nach dem Erzeugen einer Variablen die Funktion "DMGetVarInfo" oder "DMGetVarInfoExStr" zum ersten Mal verwendet wird, müssen jedesmal Suchlisten neu sortiert werden. Das kostet zusätzliche Laufzeit.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Beispiele

Skript-Beispiel "Button DMGetVarInfoExStr":

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
```

```

{
#pragma code ("kernel32.dll")
#define CP_ACP 0
int MultiByteToWideChar(UINT CodePage,
                        DWORD dwFlags,
                        LPCSTR lpMultiByteStr,
                        int cbMultiByte,
                        LPWSTR lpWideCharStr,
                        int cchWideChar);

int WideCharToMultiByte(UINT CodePage,
                        DWORD dwFlags,
                        LPCWSTR lpWideCharStr,
                        int cchWideChar,
                        LPSTR lpMultiByteStr,
                        int cbMultiByte,
                        LPCSTR lpDefaultChar,
                        LPBOOL lpUsedDefaultChar);

#pragma code ()

#pragma code ("OleAut32.dll")
// #include "OleAuto.h"
SAFEARRAY * SafeArrayCreateVector(VARTYPE vt, long lLbound, unsigned
int cElements );
HRESULT SafeArrayPutElement(SAFEARRAY FAR* psa, long FAR* rgIndices,
void FAR* pv );
HRESULT SafeArrayGetElement(SAFEARRAY FAR* psa, long FAR* rgIndices,
void FAR* pv );
HRESULT SafeArrayPtrOfIndex(SAFEARRAY FAR* psa, long FAR* rgIndices,
void HUGE* FAR* ppvData );
HRESULT SafeArrayLock(SAFEARRAY FAR* psa);
HRESULT SafeArrayUnlock(SAFEARRAY FAR* psa);
typedef WCHAR OLECHAR;
BSTR SysAllocString(OLECHAR FAR* sz);
HRESULT SysFreeString(BSTR bstr);
#pragma code ()
VARIANT vVarkeyIn;
VARIANT vVarkeyOut;
VARIANT vElem;
VARIANT* pvElem;
HRESULT hr;
CHAR szProjectName[256];
CMN_ERROR err;
DWORD dwFlags;
BOOL bRet;
DWORD dwVarID1, dwVarID2, dwVarID3, dwVarID4;
CHAR szVarNam1[256], szVarNam2[256], szVarNam3[256], szVarNam4[265];
SAFEARRAY* parrayIN;
SAFEARRAY* parrayOUT;
long lInx;
WCHAR wszBuffer[256];
int nRet;
memset(&err, 0, sizeof(err));

```

```

bRet = FALSE;
szProjectName[0] = 0;
dwFlags = 0;
VariantInit(&vVarkeyIn);
VariantInit(&vVarkeyOut);
VariantInit(&vElem);
parrayIN = NULL;
parrayOUT = NULL;
//hr = E_FAIL;
nRet = 0;
printf("\r\n\r\n##### enter Test with DMGetVarInfoExStr
#####");

bRet = DMGetRuntimeProject(szProjectName, 256, &err);
if (!bRet)
{
    printf("\r\n error DMGetRuntimeProject =>[%s], err=%ld,%ld,%ld,
    %ld,%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}
dwVarID1 = GetTagDWord("dwVarKeyID_1");
dwVarID2 = GetTagDWord("dwVarKeyID_2");
dwVarID3 = GetTagDWord("dwVarKeyID_3");
dwVarID4 = GetTagDWord("dwVarKeyID_4");
strncpy(szVarNam1, GetTagChar("szVarKeyName_1"), 256);
strncpy(szVarNam2, GetTagChar("szVarKeyName_2"), 256);
strncpy(szVarNam3, GetTagChar("szVarKeyName_3"), 256);
strncpy(szVarNam4, GetTagChar("szVarKeyName_4"), 256);
printf("\r\n set vVarkeyIn:");
printf("\r\n dwVarID1=%ld, szVarNam1=[%s] (use name as VT_BSTR)",
dwVarID1, szVarNam1);
printf("\r\n dwVarID2=%ld, szVarNam2=[%s] (use name as VT_LPSTR)",
dwVarID2, szVarNam2);
printf("\r\n dwVarID3=%ld, szVarNam3=[%s] (use ID)", dwVarID3,
szVarNam3);
printf("\r\n dwVarID4=%ld, szVarNam4=[%s] (useID)", dwVarID4,
szVarNam4);
//preseting input Variant: name1, name2, id3, id4
// will deliver on output Variant: id1, id2, name3, name4
parrayIN = SafeArrayCreateVector(VT_VARIANT, 0L, 4);
vVarkeyIn.vt = (VT_ARRAY | VT_VARIANT);
vVarkeyIn.u.parray = parrayIN;
// name1 als VT_BSTR
vElem.vt = VT_BSTR;
nRet = MultiByteToWideChar(CP_ACP, 0, szVarNam1, -1, wszBuffer,
256);
vElem.u.bstrVal = SysAllocString(wszBuffer);
lInx = 0;
hr = SafeArrayPutElement(parrayIN, &lInx, &vElem);
VariantClear(&vElem);
// name2 als VT_LPSTR

```

```

SafeArrayLock(parrayIN);
lInx = 1;
hr = SafeArrayPtrOfIndex(parrayIN, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam2;
SafeArrayUnlock(parrayIN);
if (hr)
{
    printf("\r\nerror SafeArrayPutElement: hr = %08lx", hr);
}
VariantClear(&vElem);
vElem.vt = VT_I4;
vElem.u.lVal = dwVarID3;
lInx = 2;
hr = SafeArrayPutElement(parrayIN, &lInx, &vElem);
vElem.vt = VT_I4;
vElem.u.lVal = dwVarID4;
lInx = 3;
hr = SafeArrayPutElement(parrayIN, &lInx, &vElem);

memset(&err, 0, sizeof(err));
bRet = DMGetVarInfoExStr(szProjectName, dwFlags, &vVarkeyIn,
&vVarkeyOut, &err);
if (!bRet)
{
    printf("\r\n error DMGetVarInfoExStr[%s], err=%ld,%ld,%ld,%ld,
%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
    if (err.dwError1 == DM_E_DONT_EXIST)
    {
        //reset error, elements will be checked each for error in type
(VT_EMPTY == not exist)
        bRet = TRUE;
    }
}
else
{
    printf("\r\n DMGetVarInfoExStr return OK");
}
if (bRet)
{
    if (vVarkeyOut.vt == VT_EMPTY)
    {
        printf("\r\n vVarkeyOut is VT_EMPTY");
    }
    else
    {
        parrayOUT = vVarkeyOut.u.parray;
        lInx = 0;
        SafeArrayGetElement(parrayOUT, &lInx, &vElem);
        if (vElem.vt != VT_I4)

```

```

        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_I4", lInx,
vElem.vt);
            dwVarID1 = 0L;
        }
        else
        {
            dwVarID1 = vElem.u.lVal;
        }
        SetTagDWord("dwVarKeyID_1", dwVarID1);
        lInx = 1;
        SafeArrayGetElement(parrayOUT, &lInx, &vElem);
        if (vElem.vt != VT_I4)
        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_I4", lInx,
vElem.vt);
            dwVarID2 = 0L;
        }
        else
        {
            dwVarID2 = vElem.u.lVal;
        }
        SetTagDWord("dwVarKeyID_2", dwVarID2);
        lInx = 2;
        SafeArrayGetElement(parrayOUT, &lInx, &vElem);
        if (vElem.vt != VT_BSTR)
        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_BSTR",
lInx, vElem.vt);
            strcpy(szVarNam3, "error!!!");
        }
        else
        {
            nRet = WideCharToMultiByte(CP_ACP, 0L,
(WCHAR*)vElem.u.bstrVal, -1,
                                szVarNam3, 256, NULL, NULL);
        }
        SetTagChar("szVarKeyName_3", szVarNam3);
        lInx = 3;
        SafeArrayGetElement(parrayOUT, &lInx, &vElem);
        if (vElem.vt != VT_BSTR)
        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_BSTR",
lInx, vElem.vt);
            strcpy(szVarNam4, "error!!!");
        }
        else
        {
            nRet = WideCharToMultiByte(CP_ACP, 0L,
(WCHAR*)vElem.u.bstrVal, -1,
                                szVarNam4, 256, NULL, NULL);
        }
    }

```

```

        SetTagChar("szVarKeyName_4",szVarNam4);
        printf("\r\n  get vVarkeyOut:");
        printf("\r\n    dwVarID1=%ld, szVarNam1=[%s] (delivered ID)",
dwVarID1, szVarNam1);
        printf("\r\n    dwVarID2=%ld, szVarNam2=[%s] (delivered ID)",
dwVarID2, szVarNam2);
        printf("\r\n    dwVarID3=%ld, szVarNam3=[%s] (delivered name
VT_BSTR)", dwVarID3, szVarNam3);
        printf("\r\n    dwVarID4=%ld, szVarNam4=[%s] (delivered name
VT_BSTR)", dwVarID4, szVarNam4);
    }
}
printf("\r\n  set flag DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR and
repeat DMGetVarInfoExStr");
dwFlags = DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR;
memset(&err, 0, sizeof(err));
hr = VariantClear(&vVarkeyOut);
if (hr)
{
    printf("\r\nerror VariantClear(vVarkeyOut: hr = %08lx", hr);
}
bRet = DMGetVarInfoExStr(szProjectName, dwFlags, &vVarkeyIn,
&vVarkeyOut, &err);
if (!bRet)
{
    printf("\r\n  error DMGetVarInfoExStr[%s], err=%ld,%ld,%ld,%ld,
%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
    if (err.dwError1 == DM_E_DONT_EXIST)
    {
        //reset error, elements will be checked each for error in type
(VT_EMPTY == not exist)
        bRet = TRUE;
    }
}
else
{
    printf("\r\n  DMGetVarInfoExStr return OK");
}
if (bRet)
{
    if (vVarkeyOut.vt == VT_EMPTY)
    {
        printf("\r\n  vVarkeyOut is VT_EMPTY");
    }
    else
    {
        parrayOUT = vVarkeyOut.u.parray;
        lInx = 0;
        SafeArrayGetElement(parrayOUT, &lInx, &vElem);
        if (vElem.vt != VT_I4)

```

```

        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_I4", lInx,
vElem.vt);
            dwVarID1 = 0L;
        }
        else
        {
            dwVarID1 = vElem.u.lVal;
        }
        SetTagDWord("dwVarKeyID_1", dwVarID1);
        lInx = 1;
        SafeArrayGetElement(parrayOUT, &lInx, &vElem);
        if (vElem.vt != VT_I4)
        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_I4", lInx,
vElem.vt);
            dwVarID2 = 0L;
        }
        else
        {
            dwVarID2 = vElem.u.lVal;
        }
        SetTagDWord("dwVarKeyID_2", dwVarID2);

        SafeArrayLock(parrayOUT);
        lInx = 2;
        SafeArrayPtrOfIndex(parrayOUT, &lInx, &pvElem);
        if (pvElem->vt != VT_LPSTR)
        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_LPSTR",
lInx, pvElem->vt);
            strcpy(szVarNam3, "error!!!");
        }
        else
        {
            strncpy(szVarNam3, (LPSTR)pvElem->u.pbVal, 255);
            szVarNam3[255] = 0;
        }
        SetTagChar("szVarKeyName_3", szVarNam3);
        lInx = 3;
        SafeArrayPtrOfIndex(parrayOUT, &lInx, &pvElem);
        if (pvElem->vt != VT_LPSTR)
        {
            printf("\r\n vVarKeyOut[%ld].vt = [%ld] != VT_LPSTR",
lInx, pvElem->vt);
            strcpy(szVarNam4, "error!!!");
        }
        else
        {
            strncpy(szVarNam4, (LPSTR)pvElem->u.pbVal, 255);
            szVarNam4[255] = 0;
        }
    }

```

```

        SetTagChar("szVarKeyName_4",szVarNam4);
        SafeArrayUnlock(parrayOUT);

        printf("\r\n  get vVarkeyOut:");
        printf("\r\n    dwVarID1=%ld, szVarNam1=[%s] (delivered ID)",
dwVarID1, szVarNam1);
        printf("\r\n    dwVarID2=%ld, szVarNam2=[%s] (delivered ID)",
dwVarID2, szVarNam2);
        printf("\r\n    dwVarID3=%ld, szVarNam3=[%s] (delivered name
VT_LPSTR)", dwVarID3, szVarNam3);
        printf("\r\n    dwVarID4=%ld, szVarNam4=[%s] (delivered name
VT_LPSTR)", dwVarID4, szVarNam4);
    }
}

printf("\r\n  only 1 Input (ID = VT_I4 and dwVarID1) without
VT_ARRAY, set flag DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR and repeat
DMGetVarInfoExStr");
dwFlags = DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR;
memset(&err, 0, sizeof(err));
hr = VariantClear(&vVarkeyIn);
if (hr)
{
    printf("\r\n  error VariantClear(vVarkeyIn): hr = %08lx", hr);
    // know that error, have to free the PROPVARIANT in Array manually
    //this can be set to VT_EMPTY direct, while no alloc of string
was used
    parrayIN = vVarkeyIn.u.parray;
    SafeArrayLock(parrayIN);
    lInx = 1;
    hr = SafeArrayPtrOfIndex(parrayIN, &lInx, &pvElem);
    pvElem->vt = VT_EMPTY;
    pvElem->u.pbVal = NULL;
    SafeArrayUnlock(parrayIN);
    hr = VariantClear(&vVarkeyIn);
    if (hr)
    {
        printf("\r\n  error VariantClear(vVarkeyIn) again after
manually reset the PROPVARIANT in 2. elem: hr = %08lx", hr);
    }
    else
    {
        printf("\r\n  VariantClear(vVarkeyIn) OK after manually reset
the PROPVARIANT in 2. elem");
    }
}
hr = VariantClear(&vVarkeyOut);
if (hr)
{
    printf("\r\n  error VariantClear(vVarkeyOut): hr = %08lx", hr);
}
vVarkeyIn.vt = VT_I4;

```

```

vVarkeyIn.u.lVal = dwVarID1;
bRet = DMGetVarInfoExStr(szProjectName, dwFlags, &vVarkeyIn,
&vVarkeyOut, &err);
if (!bRet)
{
    printf("\r\n error DMGetVarInfoExStr[%s], err=%ld,%ld,%ld,%ld,
%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
    if (err.dwError1 == DM_E_DONT_EXIST)
    {
        //reset error, elements will be checked each for error in type
(VT_EMPTY == not exist)
        bRet = TRUE;
    }
}
else
{
    printf("\r\n DMGetVarInfoExStr return OK");
}
if (bRet)
{
    if (vVarkeyOut.vt == VT_EMPTY)
    {
        printf("\r\n vVarkeyOut is VT_EMPTY");
    }
    else
    {
        if (vVarkeyOut.vt != VT_LPSTR)
        {
            printf("\r\n vVarKeyOut.vt = [%ld] != VT_LPSTR",
vVarkeyOut.vt);
            strcpy(szVarNam1, "error!!!");
        }
        else
        {
            strncpy(szVarNam1, (LPSTR)vVarkeyOut.u.pbVal, 255);
            szVarNam1[255] = 0;
        }
        SetTagChar("szVarKeyName_1",szVarNam1);
    }
}

printf("\r\n##### exit Test with DMGetVarInfoExStr #####\r
\n");
}

```

2.2.6.18 DMGetVarLimits (RT Professional)

Verwendung

DMGetVarLimits ermittelt für die angegebenen Variablen die Grenzen, innerhalb derer Schreiboperationen mit den DMSetValue Funktionen möglich sind.

Deklaration

```
BOOL DMGetVarLimits (  
    LPCSTR          lpszProjectFile,  
    LPDM_VARKEY     lpdmVarKey,  
    DWORD           dwItems,  
    LPDM_VARLIMIT   lpdmVarLimit,  
    LPCMN_ERROR     lpdmError );
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

lpdmVarKey

Zeiger auf Strukturen vom Typ DM_VARKEY (Seite 241) zur Spezifikation von Variablen.

dwItems

Anzahl der übergebenen Strukturen.

lpdmVarLimit

Zeiger auf Strukturen vom Typ DM_VARLIMIT (Seite 244) , in welcher die Grenzwerte hinterlegt werden.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Grenzwerte ermittelt.

FALSE

Fehler.

Bemerkung

Zusätzlich zu den bei der Projektierung definierten Ober- und Untergrenzen einer Variablen entsteht durch eine eventuell hinterlegte Formatvorschrift eine weitere Begrenzung des erlaubten Zahlenbereichs.

Das folgende Beispiel stellt die Zusammenhänge für eine Variable vom Typ DM_VARTYPE_BYTE dar, die im Automatisierungssystem als Byte im BCD-Format hinterlegt ist.

Fehlermeldungen

DM_E_MAX_RANGE	Oberer Grenzwert der Formatkonvertierung überschritten
DM_E_MIN_RANGE	Unterer Grenzwert der Formatkonvertierung unterschritten
DM_E_MAX_LIMIT	Oberer Grenzwert der Variablen überschritten
DM_E_MIN_LIMIT	Unterer Grenzwert der Variablen unterschritten
DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden
DM_E_INVALID_TYPE	ungültiger Variablentyp

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

Beispiele

OnTestVariablenGetvarlimits (Seite 507) "TESTCDoc.cpp"

Siehe auch

DM_VARKEY (Seite 241)
DM_VARLIMIT (Seite 244)
OnTestVariablenGetvarlimits (Seite 507)

2.2.6.19 DMGetVarLimitsExStr (RT Professional)

Deklaration

```

BOOL DMGetVarLimitsExStr (
    LPCTSTR          lpszProjectFile,
    LPVARIANT        lpvdmVarKey,
    LPDM_VARLIMIT    lpdmVarLimit,

```

```
LPCMN_ERROR          lpdmError );
```

Beschreibung

Ermittelt für die angegebenen Variablen die Grenzen, innerhalb derer Schreiboperationen mit den "DMSetValueExStr"-Funktionen möglich sind.

Im Gegensatz zu "DMGetVarLimits" wird die Variablenliste als Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" übergeben. Damit entfällt die Begrenzung der Länge der Variablennamen.

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei können Sie mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermitteln.

lpvdmVarKey

Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" zur Übergabe der Variablenliste oder einfachen VARIANT, wenn nur eine Variable angegeben wird.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

lpdmVarLimit

Zeiger auf Strukturen vom Typ "DM_VARLIMIT", in der die Grenzwerte hinterlegt werden.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Grenzwerte ermittelt

FALSE

Fehler

Bemerkung

Zusätzlich zu den projektierten Ober- und Untergrenzen einer Variablen entsteht durch eine hinterlegte Formatvorschrift eine weitere Begrenzung des erlaubten Zahlenbereichs.

Das folgende Beispiel stellt die Zusammenhänge für eine Variable vom Typ "DM_VARTYPE_BYTE" dar, die in der AS als Byte im "BCD"-Format hinterlegt ist.

Fehlermeldungen

DM_E_MAX_RANGE	Oberer Grenzwert der Formatkonvertierung überschritten
DM_E_MIN_RANGE	Unterer Grenzwert der Formatkonvertierung unterschritten
DM_E_MAX_LIMIT	Oberer Grenzwert der Variablen überschritten
DM_E_MIN_LIMIT	Unterer Grenzwert der Variablen unterschritten
DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden
DM_E_INVALID_TYPE	ungültiger Variablentyp

Benötigte Files

dmclient_exstr.h
 dmclient.lib
 dmclient.dll

Beispiele

Skript-Beispiel "Button DMGetVarLimitsExStr":

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
#pragma code ("OleAut32.dll")
SAFEARRAY * SafeArrayCreateVector(VARTYPE vt, long lLbound, unsigned
int cElements );
HRESULT SafeArrayPtrOfIndex(SAFEARRAY FAR* psa, long FAR* rgIndices,
void HUGE* FAR* ppvData );
HRESULT SafeArrayLock(SAFEARRAY FAR* psa);
HRESULT SafeArrayUnlock(SAFEARRAY FAR* psa);
HRESULT VariantChangeType( VARIANTARG FAR* pvargDest, VARIANTARG
FAR* pvarSrc, unsigned short wFlags, VARTYPE vt );
#pragma code()

VARIANT vdmVarkey;
VARIANT* pvElem;
VARIANT vElem;
HRESULT hr;
CHAR szProjectName[256];
CMN_ERROR err;
BOOL bRet;
CHAR szVarNam1[256], szVarNam2[256], szVarNam3[256], szVarNam4[265];
SAFEARRAY* parray;
```

```

long lInx;
int nRet;
DM_VARLIMIT dmLimits[4];
int i;

memset(&err, 0, sizeof(err));
bRet = FALSE;
szProjectName[0] = 0;
VariantInit(&vElem);
VariantInit(&vdmVarkey);
parray = NULL;
nRet = 0;
i = 0;

printf("\r\n\r\n##### enter Test with DMGetVarLimitsExStr
#####");

bRet = DMGetRuntimeProject(szProjectName, 256, &err);
if (!bRet)
{
    printf("\r\n error DMGetRuntimeProject =>[%s], err=%ld,%ld,%ld,
%ld,%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}

strncpy(szVarNam1, GetTagChar("szVarKeyName_1"), 256);
strncpy(szVarNam2, GetTagChar("szVarKeyName_2"), 256);
strncpy(szVarNam3, GetTagChar("szVarKeyName_3"), 256);
strncpy(szVarNam4, GetTagChar("szVarKeyName_4"), 256);
printf("\r\n set the vdmVarkey with names in VT_LPSTR:");
parray = SafeArrayCreateVector(VT_VARIANT, 0L, 4);
vdmVarkey.vt = (VT_ARRAY | VT_VARIANT);
vdmVarkey.u.parray = parray;
SafeArrayLock(parray);
lInx = 0;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam1;
lInx = 1;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam2;
lInx = 2;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam3;
lInx = 3;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam4;
SafeArrayUnlock(parray);

```

```

memset(&err, 0, sizeof(err));
memset(dmLimits, 0, sizeof(DM_VARLIMIT)*4);
bRet = DMGetVarLimitsExStr(szProjectName, &vdmVarkey, dmLimits,
&err);
if (!bRet)
{
    printf("\r\n error DMGetVarLimitsExStr[%s], err=%ld,%ld,%ld,%ld,
%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
    bRet = TRUE; //clear error flag, the defect data is set to VT_EMPTY
}
else
{
    printf("\r\n DMGetVarLimitsExStr return OK");
}
if (bRet)
{
    for (i = 0; i < 4; i++)
    {
        VariantClear(&vElem);
        printf("\r\n dmLimits[%d]:{", i);
        hr = VariantChangeType((VARIANTARG*)&vElem,
(VARIANTARG*)&dmLimits[i].dmMaxRange, 0, VT_R8);
        if ((0L == hr) && (VT_R8 == vElem.vt) && (VT_EMPTY !=
dmLimits[i].dmMaxRange.vt))
        {
            printf("dmMaxRange=[%g], ", vElem.u.dblVal);
        }
        else
        {
            printf("dmMaxRange{VariantChangeType error hr = 0x%08lx,
vt=0x%04x}",
                hr, dmLimits[i].dmMaxRange.vt);
        }
        VariantClear(&vElem);
        hr = VariantChangeType((VARIANTARG*)&vElem,
(VARIANTARG*)&dmLimits[i].dmMinRange, 0, VT_R8);
        if ((0L == hr) && (VT_R8 == vElem.vt) && (VT_EMPTY !=
dmLimits[i].dmMinRange.vt))
        {
            printf("dmMinRange=[%g], ", vElem.u.dblVal);
        }
        else
        {
            printf("dmMinRange{VariantChangeType error hr = 0x%08lx,
vt=0x%04x}",
                hr, dmLimits[i].dmMinRange.vt);
        }
        VariantClear(&vElem);
    }
}

```

```

        hr = VariantChangeType((VARIANTARG*)&vElem,
(VARIANTARG*)&dmLimits[i].dmMaxLimit, 0, VT_R8);
        if ((0L == hr) && (VT_R8 == vElem.vt) && (VT_EMPTY !=
dmLimits[i].dmMaxLimit.vt))
        {
            printf("dmMaxLimit=[%g], ", vElem.u.dblVal);
        }
        else
        {
            printf("dmMaxLimitVariantChangeType error hr = 0x%08lx,
vt=0x%04x}",
                hr, dmLimits[i].dmMaxLimit.vt);
        }
        VariantClear(&vElem);
        hr = VariantChangeType((VARIANTARG*)&vElem,
(VARIANTARG*)&dmLimits[i].dmMinLimit, 0, VT_R8);
        if ((0L == hr) && (VT_R8 == vElem.vt) && (VT_EMPTY !=
dmLimits[i].dmMinLimit.vt))
        {
            printf("dmMinLimit=[%g]", vElem.u.dblVal);
        }
        else
        {
            printf("dmMinLimitVariantChangeType error hr = 0x%08lx,
vt=0x%04x}",
                hr, dmLimits[i].dmMinLimit.vt);
        }
        printf("}");
    }
}

printf("\r\n##### exit Test with DMGetVarLimitsExStr #####
\r\n");

}

```

2.2.6.20 DMGetVarType (RT Professional)

Verwendung

DMGetVarType ermittelt die Typreferenz für die übergebenen Variablen.

Deklaration

```
BOOL DMGetVarType (  
    LPCSTR      lpszProjectFile,  
    LPDM_VARKEY lpdmVarKey,  
    DWORD       dwItems,  
    LPDM_TYPEREF lpdmTypeRef,  
    LPCMN_ERROR lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

lpdmVarKey

Zeiger auf die Struktur DM_VARKEY (Seite 241) mit den Variablenschlüsseln, für welche der Typ ermittelt wird.

dwItems

Anzahl der übergebenen Strukturen.

lpdmTypeRef

Zeiger auf die Liste der Strukturen DM_TYPEREF (Seite 227) zur Rückgabe der Angaben zum Variablentyp.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Datentyp ermittelt.

FALSE

Fehler.

Bemerkung

Die Projektierungsfunktionen zum Anlegen von Variablen übergeben der aufrufenden Applikation die ausgefüllten Strukturen DM_VARREF und DM_TYPEREF mit Angaben zum Datentyp der Variablen.

Um Speicherplatz innerhalb der Applikation zu sparen, muss sich diese nicht zwangsweise die Typreferenz zu jeder Variable selbst merken, sondern kann diese mit `DMGetVarType` jederzeit vom Datenmanager abfragen.

Verwenden Sie den Aufruf nicht im Wechsel mit Aufrufen zum Erzeugen von Variablen. Wenn `DMGetVarType` nach dem Erzeugen einer Variablen zum ersten Mal verwendet wird, werden Suchlisten neu sortiert, was zusätzliche Laufzeit kostet.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Beispiele

OnTestVariablenGetvartype (Seite 508) "TESTCDoc.cpp"

Siehe auch

DM_VARKEY (Seite 241)
DM_TYPEREF (Seite 227)
OnTestVariablenGetvartype (Seite 508)

2.2.6.21 DMGetVarTypeExStr (RT Professional)

Deklaration

```
BOOL DMGetVarTypeExStr (
    LPCTSTR                lpszProjectFile,
    LPVARIANT              lpvdmVarKey,
    LPDM_TYPEREF_EXSTR    lpdmTypeRef,
    LPCMN_ERROR            lpdmError);
```

Beschreibung

Ermittelt die Typreferenz für die übergebenen Variablen.

Im Gegensatz zu "DMGetVarType" wird die Variablenliste als Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" übergeben. Damit entfällt die Begrenzung der Länge der Variablennamen. Das gilt auch für die Typnamen in den Strukturen "DM_TYPEREF_EXSTR".

Parameter

IpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei können Sie mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermitteln.

lpvdmVarKey

Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" zur Übergabe der Variablenliste.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

lpdmTypeRef

Zeiger auf die Liste der Strukturen "DM_TYPEREF_EXSTR" zur Rückgabe der Angaben zum Variablentyp.

Die Liste muss mindestens die Größe des Variant-Array in "lpvdmVarKey" haben. Die Zeiger "IpszTypeName" und "dwBufferCount" müssen initialisiert sein und auf ein eigenes String-Array verweisen.

Wenn "IpszTypeName" "NULL" ist, prüfen Sie die Rückgabe. Allokierte Zeiger müssen Sie selbst wieder freigeben.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR" . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Datentyp ermittelt

FALSE

Fehler

Bemerkung

Um Speicherplatz innerhalb der Applikation zu sparen, muss sich die Applikation nicht zwangsweise die Typreferenz zu jeder Variable merken. Die Typreferenz können Sie mit "DMGetVarType" jederzeit vom Datenmanager abfragen.

Der Aufruf dürfen Sie nicht im Wechsel mit Aufrufen zum Erzeugen von Variablen verwenden. Wenn nach dem Erzeugen einer Variablen die Funktion "DMGetVarType" oder "DMGetVarTypeExStr" zum ersten Mal verwendet wird, müssen jedesmal Suchlisten neu sortiert werden. Das kostet zusätzliche Laufzeit.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden

Benötigte Files

dmclient_exstr.h
 dmclient.lib
 dmclient.dll

Beispiele

Skript-Beispiel "Button DMGetVarTypeExStr":

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
#pragma code ("OleAut32.dll")
SAFEARRAY * SafeArrayCreateVector(VARTYPE vt, long llbound, unsigned
int cElements );
HRESULT SafeArrayPtrOfIndex(SAFEARRAY FAR* psa, long FAR* rgIndices,
void HUGE* FAR* ppvData );
HRESULT SafeArrayLock(SAFEARRAY FAR* psa);
HRESULT SafeArrayUnlock(SAFEARRAY FAR* psa);
#pragma code ()

VARIANT vdmVarkey;
VARIANT* pvElem;
HRESULT hr;
CHAR szProjectName[256];
CMN_ERROR err;
BOOL bRet;
CHAR szVarNam1[256], szVarNam2[256], szVarNam3[256], szVarNam4[256];
SAFEARRAY* parray;
long lInx;
int nRet;
DM_TYPEREF_EXSTR dmTypeRef[4];
CHAR szTypeName[4][256];
int i;

memset(&err, 0, sizeof(err));
memset(dmTypeRef, 0, sizeof(DM_TYPEREF_EXSTR)*4);
szTypeName[0][0] = 0;
szTypeName[1][0] = 0;
szTypeName[2][0] = 0;
szTypeName[3][0] = 0;
bRet = FALSE;
szProjectName[0] = 0;
VariantInit(&vdmVarkey);
```

```

parray = NULL;
nRet = 0;
i = 0;

printf("\r\n\r\n##### enter Test with DMGetVarTypeExStr
#####");

bRet = DMGetRuntimeProject(szProjectName, 256, &err);
if (!bRet)
{
    printf("\r\n error DMGetRuntimeProject =>[%s], err=%ld,%ld,%ld,%ld,
%ld,%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}

strncpy(szVarNam1, GetTagChar("szVarKeyName_1"), 256);
strncpy(szVarNam2, GetTagChar("szVarKeyName_2"), 256);
strncpy(szVarNam3, GetTagChar("szVarKeyName_3"), 256);
strncpy(szVarNam4, GetTagChar("szVarKeyName_4"), 256);
printf("\r\n set the vdmVarkey with names in VT_LPSTR:");
parray = SafeArrayCreateVector(VT_VARIANT, 0L, 4);
vdmVarkey.vt = (VT_ARRAY | VT_VARIANT);
vdmVarkey.u.parray = parray;
SafeArrayLock(parray);
lInx = 0;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam1;
lInx = 1;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam2;
lInx = 2;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam3;
lInx = 3;
hr = SafeArrayPtrOfIndex(parray, &lInx, &pvElem);
pvElem->vt = VT_LPSTR;
pvElem->u.pbVal = szVarNam4;
SafeArrayUnlock(parray);

printf("\r\n call DMGetVarTypeExStr with NULL for names");
memset(&err, 0, sizeof(err));
bRet = DMGetVarTypeExStr(szProjectName, &vdmVarkey, dmTypeRef, &err);
if (!bRet)
{
    printf("\r\n error DMGetVarTypeExStr[%s], err=%ld,%ld,%ld,%ld,
%ld,[%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}

```

```

}
else
{
    printf("\r\n  DMGetVarTypeExStr return OK");
}
if (bRet)
{
    for (i = 0; i < 4; i++)
    {
        printf("\r\n    dmTypeRef[%d]:{" , i);
        printf("dwType=[%ld]", dmTypeRef[i].dwType);
        printf(", dwSize=[%ld]", dmTypeRef[i].dwSize);
        if (dmTypeRef[i].lpszTypeName)
        {
            printf(", lpszTypeName=[%s]", dmTypeRef[i].lpszTypeName);
            printf(", dwNameCharCount=[%ld]",
dmTypeRef[i].dwNameCharCount);
            //clear the delivered strings that no memoryleak
            free(dmTypeRef[i].lpszTypeName);
            dmTypeRef[i].lpszTypeName = NULL;
            dmTypeRef[i].dwNameCharCount = 0L;
            //set name store for next read (no need to free after
call)
            dmTypeRef[i].lpszTypeName = szTypeName[i];
            dmTypeRef[i].dwNameCharCount = 255;
        }
        printf("}");
    }
}
printf("\r\n  call DMGetVarTypeExStr with pointers and size for name
storage");
memset(&err, 0, sizeof(err));
bRet = DMGetVarTypeExStr(szProjectName, &vdmVarkey, dmTypeRef, &err);
if (!bRet)
{
    printf("\r\n  error DMGetVarTypeExStr[%s], err=%ld,%ld,%ld,%ld,
%ld, [%s]", szProjectName,
        err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}
else
{
    printf("\r\n  DMGetVarTypeExStr return OK");
}
if (bRet)
{
    for (i = 0; i < 4; i++)
    {
        printf("\r\n    dmTypeRef[%d]:{" , i);
        printf("dwType=[%ld]", dmTypeRef[i].dwType);
        printf(", dwSize=[%ld]", dmTypeRef[i].dwSize);
        printf(", lpszTypeName=[%s]", dmTypeRef[i].lpszTypeName);
    }
}

```

```

        printf(", dwNameCharCount=[%ld]",
dmTypeRef[i].dwNameCharCount);
        printf("}");
    }
}
printf("\r\n##### exit Test with DMGetVarTypeExStr #####\r\n");
}

```

2.2.6.22 DMSetValue (RT Professional)

Verwendung

Die Funktion ändert die Werte der durch `lpdmVarKey` beschriebenen Variablen auf die in `lpdmValue` angegebenen Werte.

Das Absetzen der Änderungen der Variablen ist asynchron nach dem Motto "fire and forget".

Deklaration

```

BOOL DMSetValue (
    LPDM_VARKEY    lpdmVarKey,
    DWORD          dwItems,
    LPVARIANT      lpdmValue,
    LPDWORD        lpdmVarState,
    LPCMN_ERROR    lpdmError);

```

Parameter

lpdmVarKey

Zeiger auf die erste der Strukturen `DM_VARKEY` (Seite 241), welche die zu ändernden Variablen kennzeichnen.

dwItems

Anzahl der übergebenen Strukturen (entspricht der Anzahl der zu ändernden Variablen).

lpdmValue

Zeiger auf den ersten der neuen Werte der zu ändernden Variablen.

lpdmVarState

Zeiger auf die erste der Speicherstellen, an denen Informationen hinterlegt werden, ob der Variablenwert erfolgreich geändert werden konnte oder ob Fehler aufgetreten sind.

Eine 0 (OK) bedeutet, dass erfolgreich gesendet/geändert wurde, und dass mindestens eine der angegebenen Variablen vorhanden ist. Nur wenn keine der angegebenen Variablen vorhanden sind, kommt ein Fehler mit dem Status DM_VARSTATE_INVALID_KEY.

Um auch zu prüfen, ob der Wert auch übernommen wurde, muss DMSetValueWait verwendet werden.

DM_VARSTATE_NOT_ESTABLISHED	(0x0001)	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	(0x0002)	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	(0x0004)	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	(0x0008)	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	(0x0010)	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	(0x0020)	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	(0x0040)	Formatgrenze unterschritten
DM_VARSTATE_CONVERSION_ERROR	(0x0080)	Anzeige Wandlungsfehler (in Verbindung mit DM_VARSTATE_..._RANGE)
DM_VARSTATE_STARTUP_VALUE	(0x0100)	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	(0x0200)	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	(0x0400)	Adressierfehler im Kanal
DM_VARSTATE_INVALID_KEY	(0x0800)	Variable nicht gefunden / nicht vorhanden
DM_VARSTATE_ACCESS_FAULT	(0x1000)	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	(0x2000)	Timeout / keine Rückmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	(0x4000)	Server ist down

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablen geändert.

FALSE

Fehler.

Bemerkung

Die Angabe einer Zugriffsvorschrift für lpdmValue ist nicht notwendig, da der Datenmanager wegen der übergebenen Variablenschlüssel eine Zuordnung zum jeweiligen Typ treffen kann.

Bei externen Variablen erfolgt ein Schreibzugriff auf den Prozess, bei WinCC-Client PCs wird der Auftrag an den jeweiligen Server weitergeleitet.

Im Normalfall werden die Schreibaufträge der Applikationen asynchron ausgeführt. Alle Schreibaufträge werden in der Warteschlange des Datenmanagers auf dem Server eingereicht und sequenziell an die zuständige Kanal-DLL übergeben.

Wenn von der Kanal-DLL der Schreibauftrag positiv bestätigt wird, ist der neue Wert im Prozessabbild des für den Auftrag zuständigen Servers gültig. Im Anschluss wird die Aktualisierung innerhalb der über den Systembus vernetzten WinCC-Stationen vorgenommen.

Daraus ergibt sich, dass kurzfristig auf verschiedenen Stationen verschiedene Werte für eine Variable auftreten können.

Fehlermeldungen

DM_E_MAX_LIMIT	Oberer Grenzwert der Variablen überschritten
DM_E_MIN_LIMIT	Unterer Grenzwert der Variablen unterschritten
DM_E_MAX_RANGE	Oberer Grenzwert der Formatkonvertierung überschritten
DM_E_MIN_RANGE	Unterer Grenzwert der Formatkonvertierung unterschritten
DM_E_ACCESS_FAULT	unerlaubter Schreibzugriff auf eine Variable
DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DMGetValue (Seite 341)	Variablenwerte ermitteln
DMSetValueMessage (Seite 391)	Variablenwerte mit Meldungsausgabe ändern
DMSetValueWait (Seite 394)	Variablenwerte mit Benachrichtigung ändern
DMSetValueWaitMessage (Seite 400)	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern

Beispiele

Write tag (Seite 520) "DM02.cpp"

OnTestVariablenSetvalue (Seite 510) "TESTCDoc.cpp"

Siehe auch

DM_VARKEY (Seite 241)

DMGetValue (Seite 341)

DMGetValueEx (Seite 343)

DMSetValueMessage (Seite 391)

DMSetValueWait (Seite 394)

DMSetValueWaitMessage (Seite 400)

Write tag (Seite 520)

OnTestVariablenSetValue (Seite 510)

2.2.6.23 DMSetValueExStr (RT Professional)

Deklaration

```
BOOL DMSetValueExStr (  
    LPVARIANT      lpvdmVarKey,  
    LPVARIANT      lpvdmValue,  
    LPDWORD        lpdmVarState,  
    LPCMN_ERROR    lpdmError);
```

Beschreibung

Die Funktion ändert die Werte der durch "lpvdmVarKey" beschriebenen Variablen auf die in "lpvdmValue" angegebenen Werte.

Das Absetzen der Variablenänderungen geschieht asynchron nach dem Motto "fire and forget".

Im Gegensatz zu "DMSetValue" wird die Variablenliste als Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" übergeben. Damit entfällt die Begrenzung der Länge der Variablennamen.

Parameter

lpvdmVarKey

Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" zur Übergabe der Variablenliste oder einfachen VARIANT, wenn nur eine Variable angegeben wird, die die zu ändernden Variablen kennzeichnen.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

Bei Multiclient-Projekten ab WinCC V5.0 ist hier gegebenenfalls jedem Namen ein Serverpräfix mit anzugeben (siehe Projekttypen und Versionen).

IpvdmValue

Zeiger auf den ersten der neuen Werte der zu ändernden Variablen.

IpvdmVarState

Zeiger auf die erste der Speicherstellen, an denen Informationen hinterlegt werden, ob der Wert der Variablen erfolgreich geändert werden konnte oder ob Fehler aufgetreten sind.

Eine "0" (OK) bedeutet, dass erfolgreich gesendet/geändert wurde, und mindestens eine der angegebenen Variablen vorhanden ist. Nur wenn keine der angegebenen Variablen vorhanden sind, kommt ein Fehler mit dem Status "DM_VARSTATE_INVALID_KEY".

Um auch zu prüfen, ob der Wert auch vom AS übernommen wurde, muss "DMSetValueWaitExStr" verwendet werden.

DM_VARSTATE_NOT_ESTABLISHED	(0x0001)	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	(0x0002)	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	(0x0004)	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	(0x0008)	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	(0x0010)	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	(0x0020)	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	(0x0040)	Formatgrenze unterschritten
DM_VARSTATE_CONVERSION_ERROR	(0x0080)	Anzeige Wandlungsfehler (in Verbindung mit DM_VARSTATE_..._RANGE)
DM_VARSTATE_STARTUP_VALUE	(0x0100)	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	(0x0200)	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	(0x0400)	Adressierungsfehler im Kanal
DM_VARSTATE_INVALID_KEY	(0x0800)	Variable nicht gefunden / nicht vorhanden
DM_VARSTATE_ACCESS_FAULT	(0x1000)	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	(0x2000)	Timeout / keine Rückmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	(0x4000)	Server ist down

IpvdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Variablen geändert.

FALSE

Fehler

Bemerkung

Die Angabe einer Zugriffsvorschrift für "IpdmValue" ist nicht notwendig, da der Datenmanager aufgrund der übergebenen Variablenschlüssel eine Zuordnung zum jeweiligen Typ treffen kann.

Bei externen Variablen erfolgt ein Schreibzugriff auf den Prozess, wobei bei WinCC-Client Rechnern der Auftrag an den jeweiligen Server weitergeleitet wird.

Im Normalfall werden die Schreibaufträge der Applikationen asynchron ausgeführt, d.h. alle Schreibaufträge werden in der Warteschlange des Datenmanagers auf dem Server eingereicht und sequentiell an die zuständige Kanal-DLL übergeben.

Der neue Wert ist im Prozessabbild des für den Auftrag zuständigen Servers gültig, wenn von der Kanal-DLL der Schreibauftrag positiv bestätigt wird. Im Anschluss wird die Aktualisierung innerhalb der über den Systembus vernetzten WinCC-Stationen vorgenommen.

Daraus ergibt sich, dass kurzfristig auf verschiedenen Stationen verschieden Werte für eine Variable auftreten können.

Fehlermeldungen

DM_E_MAX_LIMIT	Oberer Grenzwert der Variablen überschritten
DM_E_MIN_LIMIT	Unterer Grenzwert der Variablen unterschritten
DM_E_MAX_RANGE	Oberer Grenzwert der Formatkonvertierung überschritten
DM_E_MIN_RANGE	Unterer Grenzwert der Formatkonvertierung unterschritten
DM_E_ACCESS_FAULT	unerlaubter Schreibzugriff auf eine Variable
DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Benötigte Files

- dmclient_exstr.h
- dmclient.lib
- dmclient.dll

Verwandte Funktionen

DMGetValueExStr	Variablenwerte ermitteln
DMSetValueMessageExStr	Variablenwerte mit Meldungsausgabe ändern
DMSetValueWaitExStr	Variablenwerte mit Benachrichtigung ändern
DMSetValueWaitMessageExStr	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern

Beispiele

Beispiele finden Sie auf der Seite "DMGetValueExStr".

2.2.6.24 DMSetValueMessage (RT Professional)

Verwendung

Die Funktion ändert den Wert der durch `lpdmVarKey` beschriebenen Variable auf den in `lpdmValue` angegebenen Wert.

Im Unterschied zur Funktion `DMSetValue` wird nur eine Variable geändert. Bei erfolgreicher Änderung des Variablenwerts wird ein frei definierbarer Meldetext abgesetzt.

Deklaration

```

BOOL DMSetValueMessage (
    LPDLM_VARKEY    lpdmVarKey,
    LPVARIANT       lpdmValue,
    DWORD          fFlags,
    LPSTR          lpszMessage,
    LPCMN_ERROR    lpdmError);

```

Parameter

lpdmVarKey

Zeiger auf die Struktur `DM_VARKEY` (Seite 241), welche die zu ändernde Variable kennzeichnet.

Bei Multiclient Projekten ab der WinCC Version 5.0 wird gegebenenfalls in `lpdmVarKey->szName` ein Serverpräfix mit angegeben (siehe Projekttypen und Versionen).

lpdmValue

Zeiger auf den neuen Wert der zu ändernden Variablen.

fFlags

Über `fFlags` wird festgelegt, wie der Meldetext verarbeitet wird:

DMSVM_OPERATIONREPORT	Die Meldung wird im Bedienprotokoll hinterlegt.
DMSVM_OPERATIONMESSAGE	Die Meldung wird als Bedienmeldung ausgegeben.

lpszMessage

Text der Meldung, die abgesetzt wird.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variable geändert.

FALSE

Fehler.

Bemerkung

Weitere Informationen zum Verändern von Variablenwerten finden Sie bei der Funktion DMSetValue.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMSetValue (Seite 385)	Variablenwerte ändern
DMSetValueWait (Seite 394)	Variablenwerte mit Benachrichtigung ändern
DMSetValueWaitMessage (Seite 400)	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern

Siehe auch

DM_VARKEY (Seite 241)

DMSetValue (Seite 385)

DMSetValueWait (Seite 394)

DMSetValueWaitMessage (Seite 400)

2.2.6.25 DMSsetValueMessageExStr (RT Professional)

Deklaration

```

BOOL DMSsetValueMessageExStr (
    LPVARIANT      lpvdmVarKey,
    LPVARIANT      lpvdmValue,
    DWORD          fFlags,
    LPTSTR         lpszMessage,
    LPCMN_ERROR    lpdmError);

```

Beschreibung

Die Funktion ändert den Wert der durch "lpvdmVarKey" beschriebenen Variablen auf den in "lpdmValue" angegebenen Wert.

Im Unterschied zur Funktion "DMSsetValueExStr" kann nur eine Variable geändert werden. Bei erfolgreicher Änderung des Variablenwertes wird ein frei definierbarer Meldungstext abgesetzt.

Parameter

lpvdmVarKey

Pointer auf VARIANT zur Übergabe des Variablennamens oder VariablenID oder einfachen VARIANT, wenn nur eine Variable angegeben wird, die die zu ändernden Variablen kennzeichnen.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

Wenn Sie ein Array mit "VT_ARRAY | VT_VARIANT" angeben, wird nur das erste Element ausgewertet.

Bei Multiclient Projekten ab WinCC V5.0 ist hier gegebenenfalls ein Serverpräfix im Namen mit anzugeben (siehe Projekttypen und Versionen).

lpvdmValue

Zeiger auf den neuen Wert der zu ändernden Variablen.

fFlags

Legt fest, wie der Meldungstext verarbeitet wird:

DMSVM_OPERATIONREPORT	Die Meldung wird im Bedienprotokoll hinterlegt.
DMSVM_OPERATIONMESSAGE	Die Meldung wird als Bedienmeldung ausgegeben.

lpszMessage

Text der Meldung, die abgesetzt wird.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR" . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablen geändert

FALSE

Fehler

Bemerkung

Weitere Informationen zum Ändern von Variablenwerten finden Sie bei der Funktion "DMSetValueExStr".

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMSetValueExStr	Variablenwerte ändern
DMSetValueWaitExStr	Variablenwerte mit Benachrichtigung ändern
DMSetValueWaitMessageExStr	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern

2.2.6.26 DMSetValueWait (RT Professional)

Verwendung

Die Funktion ändert die Werte der durch lpdmVarKey beschriebenen Variablen auf die in lpdmValue angegebenen Werte.

Im Unterschied zu `DMSetValue` besteht mit `DMSetValueWait` für die Applikation die Möglichkeit, sich bei erfolgtem Update auf der lokalen Maschine benachrichtigen zu lassen.

Deklaration

```

BOOL DMSetValueWait (
    LPDWORD                pdwTAID,
    LPDM_VARKEY            lpdmVarKey,
    DWORD                  dwItems,
    LPVARIANT              lpdmValue,
    DWORD                  dwTimeout,
    DM_COMPLETION_PROC     lpfnCompletion,
    LPVOID                  lpvUser,
    LPCMN_ERROR             lpdmError);

```

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion die vom Datenmanager vergebene Transaktions-ID enthält.

lpdmVarKey

Zeiger auf die erste der Strukturen `DM_VARKEY` (Seite 241), welche die zu ändernden Variablen kennzeichnen.

dwItems

Anzahl der übergebenen Strukturen (entspricht der Anzahl der zu ändernden Variablen).

lpdmValue

Zeiger auf den ersten der neuen Werte der zu ändernden Variablen.

dwTimeout

Maximale Wartezeit der Applikation in ms. Wenn nach Ablauf der Wartezeit nicht alle Variablen geschrieben wurden, so wird die Callback-Funktion mit entsprechenden Fehlercodes aufgerufen.

lpfnCompletion

Zeiger auf Ihre Callback-Funktion, die entweder nach der Aktualisierung aller angeforderten Variablen oder nach Ablauf der Wartezeit aufgerufen wird.

Wenn ein Programm eine Notify-Routine anmeldet, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die erste von dwltems Fehlerstrukturen vom Typ CMN_ERROR. Wenn beim Schreiben einer Variablen ein Fehler auftritt, so schreibt das System die Fehlerinformation in die entsprechende Struktur. Vergessen Sie daher nicht, Platz für diese Strukturen zu reservieren.

Rückgabewert

TRUE

Variablen geändert.

FALSE

Der aufgetretene Fehler ist über die Fehlerstrukturen identifizierbar.

Bemerkung

Weitere Informationen zum Verändern von Variablenwerten finden Sie bei der Funktion DMSetValue.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMSetValue (Seite 385)	Variablenwerte ändern
DMSetValueMessage (Seite 391)	Variablenwerte mit Meldungsausgabe ändern
DMSetValueWaitMessage (Seite 400)	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern
DM_COMPLETITION_PROC (Seite 405)	Variablenwerte ändern (Callback)

Beispiele

OnTestVariablenSetvaluwait (Seite 511) "TESTCDoc.cpp"

Siehe auch

DM_VARKEY (Seite 241)
 DMGetValueWait (Seite 353)
 DMGetValueWaitEx (Seite 356)
 DMSetValue (Seite 385)
 DMSetValueMessage (Seite 391)
 DMSetValueWaitMessage (Seite 400)
 DM_COMPLETITION_PROC (Seite 405)
 OnTestVariablenSetvaluwait (Seite 511)

2.2.6.27 DMSetValueWaitExStr (RT Professional)

Deklaration (

```

BOOL DMSetValueWaitExStr (
    LPDWORD                pdwTAID,
    LPVARIANT              lpvdmVarKey,
    DWORD                  dwItems,
    LPVARIANT              lpvdmValue,
    LPDWORD                lpdwState,
    DWORD                  dwTimeOut,
    DM_COMPLETITION_PROC  lpfnCompletition,
    LPVOID                 lpvUser,
    LPCMN_ERROR            lpdmError);

```

Beschreibung

Die Funktion ändert die Werte der durch "lpvdmVarKey" beschriebenen Variablen auf die in "lpvdmValue" angegebenen Werte.

Im Unterschied zu "DMSetValueExStr" besteht mit "DMSetValueWaitExStr" für die Applikation die Möglichkeit, sich nach einem Update auf dem lokalen Rechner benachrichtigen zu lassen.

Im Gegensatz zu "DMSetValueWait" wird die Variablenliste als Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" übergeben. Damit entfällt die Begrenzung der Länge der Variablennamen.

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion die vom Datenmanager vergebene Transaktions-ID enthält.

lpvdmVarKey

Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" zur Übergabe der Variablenliste oder einfachen VARIANT, wenn nur eine Variable angegeben wird, die die zu ändernden Variablen kennzeichnen.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

Bei Multiclient Projekten ab der WinCC Version 5.0 ist hier gegebenenfalls jedem Namen ein Serverpräfix mit anzugeben (siehe Projekttypen und Versionen).

dwlItems

Anzahl der übergebenen Werte in "lpvdmValue" und Fehlerstrukturen in "lpdmError". Die Anzahl muss mindestens so groß sein wie das Eingangs-Array in "lpvdmVarKey".

Wenn die Anzahl größer ist als das Eingangs-Array, werden die überzähligen Werte ignoriert und die betreffenden Fehlerstrukturen mit "DM_E_PARAM" besetzt, da keine Zuordnung möglich ist.

Wenn die Anzahl kleiner ist als das Eingangs-Array, wird nur in der ersten Fehlerstruktur der Fehler "DM_E_PARAM" zurückgegeben und keine Werte geschrieben.

lpvdmValue

Zeiger auf den ersten der neuen Werte der zu ändernden Variablen.

lpdwState

Zeiger auf die erste der Speicherstellen, an denen Informationen hinterlegt werden, ob der Wert der Variablen erfolgreich geändert werden konnte oder ob Fehler aufgetreten sind.

Eine 0 (OK) bedeutet, dass erfolgreich gesendet/geändert wurde, und dass mindestens eine der angegebenen Variablen vorhanden ist. Nur wenn keine der angegebenen Variablen vorhanden sind, kommt ein Fehler mit dem Status DM_VARSTATE_INVALID_KEY.

DM_VARSTATE_NOT_ESTABLISHED	(0x0001)	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	(0x0002)	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	(0x0004)	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	(0x0008)	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	(0x0010)	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	(0x0020)	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	(0x0040)	Formatgrenze unterschritten
DM_VARSTATE_CONVERSION_ERROR	(0x0080)	Anzeige Wandlungsfehler (in Verbindung mit DM_VARSTATE_..._RANGE)
DM_VARSTATE_STARTUP_VALUE	(0x0100)	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	(0x0200)	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	(0x0400)	Adressierungsfehler im Kanal
DM_VARSTATE_INVALID_KEY	(0x0800)	Variable nicht gefunden / nicht vorhanden

DM_VARSTATE_ACCESS_FAULT	(0x1000)	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	(0x2000)	Timeout / keine Rückmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	(0x4000)	Server ist down

dwTimeout

Maximale Wartezeit der Applikation in "ms". Wenn nach Ablauf der Wartezeit nicht alle Variablen geschrieben sind, wird die Callback-Funktion mit entsprechenden Fehlercodes aufgerufen.

lpfnCompletion

Zeiger auf die Callback-Funktion, die entweder nach der Aktualisierung aller angeforderten Variablen oder nach Ablauf der Wartezeit aufgerufen wird.

Wenn ein Programm eine Notify-Routine anmeldet, muss das Programm regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen werden bereits die Notify geliefert, bevor der Funktionsaufruf zurückgekehrt ist.

lpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die erste von dwltems Fehlerstrukturen vom Typ "CMN_ERROR". Tritt beim Schreiben einer Variablen ein Fehler auf, so schreibt das System die Fehlerinformation in die entsprechende Struktur. Vergessen Sie daher nicht, Platz für diese Strukturen zu reservieren und zu initialisieren.

Rückgabewert**TRUE**

Variablen geändert

FALSE

Der aufgetretene Fehler ist über die Fehlerstrukturen identifizierbar.

Bemerkung

Weitere Informationen zum Verändern von Variablenwerten finden Sie bei der Funktion "DMSetValueExStr".

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter

Benötigte Files

dmclient_exstr.h
dmclient.lib
dmclient.dll

Verwandte Funktionen

DMSetValueExStr	Variablenwerte ändern
DMSetValueMessageExStr	Variablenwerte mit Meldungsausgabe ändern
DMSetValueWaitMessageExStr	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern
DM_COMPLETITION_PROC	Variablenwerte ändern (Callback)

Beispiele

Beispiele finden Sie auf der Seite "DMGetValueExStr".

2.2.6.28 DMSetValueWaitMessage (RT Professional)

Verwendung

Die Funktion vereinigt die Funktionalitäten von DMSetValueMessage und DMSetValueWait. Im Unterschied zur Funktion DMSetValueWait ist nur eine Variable änderbar.

Der Wert, der durch lpdmVarKey beschriebenen Variablen wird, auf den in lpdmValue angegebenen Wert geändert. Bei erfolgreicher Änderung des Variablenwerts wird zum einen ein frei definierbarer Meldetext abgesetzt, zum anderen besteht für die Applikation die Möglichkeit, sich bei erfolgtem Update auf der lokalen Maschine benachrichtigen zu lassen.

Deklaration

```
BOOL DMSetValueWaitMessage (
    LPDWORD          pdwTAID,
    LPDM_VARKEY      lpdmVarKey,
    LPVARIANT         lpdmValue,
    DWORD            dwTimeOut,
    DM_COMPLETITION_PROC lpfnCompletion,
    DWORD            fFlags,
    LPSTR             lpszMessage,
    LPVOID            lpvUser,
    LPCMN_ERROR       lpdmError);
```

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion die vom Datenmanager vergebene Transaktions-ID enthält.

lpdmVarKey

Zeiger auf die Struktur DM_VARKEY (Seite 241) , welche die zu ändernde Variable kennzeichnen.

lpdmValue

Zeiger auf den neuen Wert der zu ändernden Variablen.

dwTimeOut

Maximale Wartezeit der Applikation in ms. Wenn nach Ablauf der Wartezeit nicht alle Variablen geschrieben wurden, so wird die Callback-Funktion mit entsprechenden Fehlercodes aufgerufen.

lpfnCompletion

Zeiger auf Ihre Callback-Funktion, die entweder nach der Aktualisierung aller angeforderten Variablen oder nach Ablauf der Wartezeit aufgerufen wird.

Wenn ein Programm eine Notify-Routine anmeldet, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

fFlags

Über fFlags wird festgelegt, wie der Meldetext verarbeitet wird:

DMSVM_OPERATIONREPORT	Die Meldung wird im Bedienprotokoll hinterlegt.
DMSVM_OPERATIONMESSAGE	Die Meldung wird als Bedienmeldung ausgegeben.

lpzMMessage

Meldetext, der abgesetzt wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die erste von dwltems Fehlerstrukturen vom Typ CMN_ERROR. Wenn beim Schreiben einer Variablen ein Fehler auftritt, so schreibt das System die Fehlerinformation in die entsprechende Struktur. Vergessen Sie daher nicht, Platz für diese Strukturen zu reservieren.

Rückgabewert

TRUE

Variablen geändert.

FALSE

Fehler.

Bemerkung

Weitere Informationen zum Verändern von Variablenwerten finden Sie bei der Funktion `DMSetValue`.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_COMPLETITION_PROC (Seite 405)	Variablenwerte ändern (Call-back)
DMSetValue (Seite 385)	Variablenwerte ändern
DMSetValueMessage (Seite 391)	Variablenwerte mit Meldungsausgabe ändern
DMSetValueWait (Seite 394)	Variablenwerte mit Benachrichtigung ändern

Siehe auch

DM_VARKEY (Seite 241)

DMSetValue (Seite 385)

DMSetValueMessage (Seite 391)

DMSetValueWait (Seite 394)

OnTestVariablenSetvaluwait (Seite 511)

DM_COMPLETITION_PROC (Seite 405)

2.2.6.29 DMSetValueWaitMessageExStr (RT Professional)

Deklaration

```

BOOL DMSetValueWaitMessageExStr (
    LPDWORD                pdwTAID,
    LPVARIANT              lpvdmVarKey,
    LPVARIANT              lpvdmValue,
    DWORD                  dwTimeOut,
    DM_COMPLETION_PROC     lpfnCompletion,
    DWORD                  fFlags,
    LPTSTR                 lpszMessage,
    LPVOID                 lpvUser,
    LPCMN_ERROR            lpdmError);

```

Beschreibung

Die Funktion vereinigt die Funktionalitäten von "DMSetValueMessageExStr" und "DMSetValueWaitExStr".

Im Unterschied zur Funktion "DMSetValueWaitExStr" kann nur eine Variable geändert werden.

Der Wert der durch "lpvdmVarKey" beschriebenen Variablen wird auf den in "lpvdmValue" angegebenen Wert geändert. Bei erfolgreicher Änderung des Variablenwertes wird zum einen ein frei definierbarer Meldungstext abgesetzt. Zum anderen besteht für die Applikation die Möglichkeit, sich bei erfolgtem Update auf dem lokalen Rechner benachrichtigen zu lassen.

Im Gegensatz zu "DMSetValueWaitMessage" wird die Variable als Pointer auf VARIANT übergeben. Damit entfällt die Begrenzung der Länge des Variablennamen.

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion die vom Datenmanager vergebene Transaktions-ID enthält.

lpvdmVarKey

Pointer auf VARIANT zur Übergabe des Variablennamens, VariablenID oder einfachen VARIANT, wenn nur eine Variable angegeben wird, die die zu ändernden Variablen kennzeichnen.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

Bei Multiclient Projekten ab der WinCC Version 5.0 ist hier gegebenenfalls beim Namen ein Serverpräfix mit anzugeben (siehe Projekttypen und Versionen).

lpvdmValue

Zeiger auf den neuen Wert der zu ändernden Variablen.

dwTimeOut

Maximale Wartezeit der Applikation in "ms". Wenn nach Ablauf der Wartezeit nicht alle Variablen geschrieben sind, wird die Callback-Funktion mit entsprechenden Fehlercodes aufgerufen.

lpfnCompletion

Zeiger auf die Callback-Funktion, die entweder nach der Aktualisierung aller angeforderten Variablen oder nach Ablauf der Wartezeit aufgerufen wird.

Wenn ein Programm eine Notify-Routine anmeldet, muss das Programm regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen werden bereits die Notify geliefert, bevor der Funktionsaufruf zurückgekehrt ist.

fFlags

Über fFlags wird festgelegt wie der Meldungstext verarbeitet wird:

DMSVM_OPERATIONREPORT	Die Meldung wird im Bedienprotokoll hinterlegt.
DMSVM_OPERATIONMESSAGE	Die Meldung wird als Bedienmeldung ausgegeben.

lpzMessage

Text der Meldung, die abgesetzt wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die erste von dwItems Fehlerstrukturen vom Typ "CMN_ERROR". Tritt beim Schreiben einer Variablen ein Fehler auf, so schreibt das System die Fehlerinformation in die entsprechende Struktur. Vergessen Sie daher nicht, Platz für diese Strukturen zu reservieren.

Rückgabewert

TRUE

Variablen geändert

FALSE

Fehler

Bemerkung

Weitere Informationen zum Verändern von Variablenwerten finden Sie bei der Funktion "DMSetValueExStr".

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_PARAM	Ungültiger Parameter
DM_E_OOM	out of memory

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_COMPLETITION_PROC	Variablenwerte ändern (Callback)
DMSetValueExStr	Variablenwerte ändern
DMSetValueMessageExStr	Variablenwerte mit Meldungsausgabe ändern
DMSetValueWaitExStr	Variablenwerte mit Benachrichtigung ändern

2.2.6.30 DM_COMPLETITION_PROC (RT Professional)

Beschreibung

Damit ihre Applikation über die erfolgte Änderung von Variablenwerten benachrichtigt werden kann müssen Sie eine Callback-Funktion vom Typ DM_COMPLETITION_PROC bereitstellen.

Deklaration

```

BOOL ( * DM_COMPLETITION_PROC) (
    DWORD          dwTAID,
    LPDWORD        lpdmVarState,
    DWORD          dwItems,
    LPVOID         lpvUser);

```

Parameter

dwTAID

Transaktions-ID, die vom Datenmanager an die Funktion zur Änderung der Variablenwerte vergeben wurde.

lpdmVarState

Zeiger auf die erste der Speicherstellen, an denen Informationen hinterlegt werden, ob der Wert der Variablen erfolgreich geändert werden konnte oder ob Fehler aufgetreten sind:

Eine 0 (OK) bedeutet, dass erfolgreich gesendet und geändert wurde, und dass mindestens eine der angegebenen Variablen vorhanden ist. Nur wenn keine der angegebenen Variablen vorhanden sind, kommt ein Fehler mit dem Status DM_VARSTATE_INVALID_KEY.

DM_VARSTATE_NOT_ESTABLISHED	(0x0001)	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	(0x0002)	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	(0x0004)	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	(0x0008)	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	(0x0010)	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	(0x0020)	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	(0x0040)	Formatgrenze unterschritten
DM_VARSTATE_CONVERSION_ERROR	(0x0080)	Anzeige Wandlungsfehler (in Verbindung mit DM_VARSTATE_..._RANGE)
DM_VARSTATE_STARTUP_VALUE	(0x0100)	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	(0x0200)	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	(0x0400)	Adressierungsfehler im Kanal
DM_VARSTATE_INVALID_KEY	(0x0800)	Variable nicht gefunden / nicht vorhanden
DM_VARSTATE_ACCESS_FAULT	(0x1000)	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	(0x2000)	Timeout / keine Rueckmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	(0x4000)	Server ist down

dwItems

Anzahl der Variablenänderungen, deren Status in lpdmVarState übergeben wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

Die Rückgabewerte sind abhängig von Ihrer Implementierung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMSetValueWait (Seite 394)	Variablenwerte mit Benachrichtigung ändern
DMSetValueWaitMessage (Seite 400)	Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern

Siehe auch

DMSetValueWait (Seite 394)

DMSetValueWaitMessage (Seite 400)

2.2.6.31 DMShowVarPropertiesExStr (RT Professional)

Deklaration

```

BOOL DMShowVarPropertiesExStr (
    LPCTSTR          lpszProjectFile,
    HWND             hwndParent,
    LPCTSTR          lpszVariableName,
    DWORD            dwVariableID,
    LPCMN_ERROR      lpdmError);

```

Beschreibung

Die Funktion öffnet den Dialog zur Bearbeitung der Eigenschaften von Variablen.

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei kann mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermittelt werden.

hwndParent

Handle auf das Fenster, das als Vaterfenster für den Dialog benutzt wird.

lpszVariableName

Zeiger auf den Namen der Variable, deren Eigenschaften angezeigt werden.

Wenn "lpszVariableName" "NULL" ist, müssen Sie eine gültige "dwVariableID" angeben. Sonst kommt der Fehler "DM_E_PARAM".

dwVariableID

ID der Variable. Wird nicht verwendet, wenn in "lpszVariableName" ein Name angegeben ist.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Dialog mit "OK" verlassen.

FALSE

Fehler oder Dialog mit "Abbrechen" verlassen

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_KEY	Variable nicht gefunden
DM_E_CANCEL	Benutzer hat in einem Dialog "Abbrechen" gewählt
DM_E_PARAM	Mindestens ein Parameter ist fehlerhaft übergeben

Benötigte Files

dmclient_exstr.h
dmclient.lib
dmclient.dll

Beispiele

Skript-Beispiel "Button DMSHOWVarPropertiesExStr":

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
    BOOL bRet = FALSE;
    CHAR szProjectName[256];
    CHAR szVariableName[256];
    DWORD dwVariableID;
    CMN_ERROR err;
    HWND hwndParent;

    szProjectName[0] = 0;
    szVariableName[0] = 0;
    dwVariableID = 0L;
    memset(&err, 0, sizeof(err));
    hwndParent = NULL;

    printf("\r\n\r\n##### enter Test with DMSHOWVarProperitiesExStr
#####");

    bRet = DMGetRuntimeProject(szProjectName, 256, &err);
    if (!bRet)
    {
        printf("\r\n error DMGetRuntimeProject =>[%s], err=%ld,%ld,%ld,
%ld,%ld, [%s]", szProjectName,
            err.dwError1, err.dwError2, err.dwError3, err.dwError4,
            err.dwError5, err.szErrorText);
    }

    memset(&err, 0, sizeof(err));

    dwVariableID = GetTagDWord("dwVarKeyID_1");
    printf("\r\n call DMSHOWVarPropertiesExStr with ID");
    bRet = DMSHOWVarPropertiesExStr(szProjectName, hwndParent, NULL,
dwVariableID, &err);
    if (!bRet)
    {
        printf("\r\n error DMSHOWVarPropertiesExStr: err=%ld,%ld,%ld,
%ld,%ld, [%s]",
            err.dwError1, err.dwError2, err.dwError3, err.dwError4,
            err.dwError5, err.szErrorText);
    }
    else
    {
        printf("\r\n DMSHOWVarPropertiesExStr OK!");
    }

    strncpy(szVariableName, GetTagChar("szVarKeyName_1"), 255);
    szVariableName[255] = 0;
```

```
printf("\r\n call DMSHOWVARPROPERTIESEXSTR WITH VARNAME");
bRet = DMSHOWVARPROPERTIESEXSTR(szProjectName, hwndParent,
szVariableName, 0L, &err);
if (!bRet)
{
    printf("\r\n error DMSHOWVARPROPERTIESEXSTR: err=%ld,%ld,%ld,
%ld,%ld, [%s]",
err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}
else
{
    printf("\r\n DMSHOWVARPROPERTIESEXSTR OK!");
}

printf("\r\n##### exit Test with DMSHOWVARPROPERTIESEXSTR
#####\r\n");
}
```

2.2.6.32 DMSHOWVARDATABASE (RT Professional)

Verwendung

Die Funktion öffnet für das spezifizierte Projekt den Variablen-Auswahldialog. Im Gegensatz zu ShowVarDatabaseMulti wird nur eine Variable selektiert.

Deklaration

```
BOOL DMSHOWVARDATABASE (
    LPCSTR          lpszProjectFile,
    HWND            hwndParent,
    LPDM_DLGOPTIONS lpdmOptions,
    LPDM_VARFILTER  lpdmFilter,
    LPDM_VARKEY     lpdmVarKey,
    LPCMN_ERROR     lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

hwndParent

Handle auf das Fenster, das als Vaterfenster für den Dialog benutzt wird.

IpdmOptions

Zeiger auf die Struktur DM_DLGOPTIONS (Seite 219) mit Vorgaben, wie sich der Dialog verhalten soll. Wenn der Zeiger NULL ist, wird der Standarddialog ausgewählt.

IpdmFilter

Zeiger auf die Filterstruktur DM_VARFILTER (Seite 233). Wenn der Zeiger NULL ist, werden alle Variablen angezeigt.

Nur das Filtern nach Variablenname und Variablentyp wird unterstützt. Das Filtern von Variablen nach Gruppen und Verbindungen ist nicht implementiert.

IpdmVarKey

Zeiger auf die Struktur DM_VARKEY (Seite 241). Wenn die Eigenschaften einer Variablen angezeigt werden sollen, enthält nach Beenden des Dialogs IpdmVarKey den Schlüssel zu dieser Variablen.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Dialog mit "OK" verlassen.

FALSE

Fehler oder Dialog mit "ABBRECHEN" verlassen.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INV_PRJ	angegebenes Projekt nicht gefunden / geladen

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMSHOWVARDATABASEMULTI (Seite 415)	Variablen-Auswahldialog öffnen
------------------------------------	--------------------------------

Siehe auch

- DM_VARKEY (Seite 241)
- DM_VARFILTER (Seite 233)
- DM_DLGOPTIONS (Seite 219)
- DMSHOWVARDATABASEMULTI (Seite 415)

2.2.6.33 DMSHOWVARDATABASEEXSTR (RT Professional)

Deklaration

```
BOOL DMSHOWVARDATABASEEXSTR (  
    LPCTSTR          lpszProjectFile,  
    HWND            hwndParent,  
    LPDM_DLGOPTIONS lpdmOptions,  
    LPDM_VARFILTER  lpdmFilter,  
    LPTSTR*         lppsVariableName,  
    LPDWORD         lpdwVarNameCharCount,  
    LPDWORD         lpdwVariableID,  
    LPCMN_ERROR     lpdmError);
```

Beschreibung

Die Funktion öffnet für das spezifizierte Projekt den Variablen-Auswahldialog. Im Gegensatz zu "SHOWVARDATABASEMULTIEXSTR" können Sie nur eine Variable selektieren.

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei können Sie mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermitteln.

hwndParent

Handle auf das Fenster, das als Vaterfenster für den Dialog benutzt wird.

lpdmOptions

Zeiger auf die Struktur "DM_DLGOPTIONS" mit Vorgaben, wie sich der Dialog verhält, wenn "NULL" ist gleich Standarddialog.

lpdmFilter

Zeiger auf die Filterstruktur "DM_VARFILTER". Wenn "NULL", werden alle Variablen angezeigt.

Unterstützt werden nur das Filtern nach Variablenname und Variablentyp. Das Filtern von Variablen nach Gruppen und Verbindungen ist nicht implementiert.

lpdzVariableName

Zeiger auf Zeiger des Rückgabepuffers für den Variablennamen.

Wenn "NULL" angegeben ist, muss ein gültiger Zeiger auf "lpdwVariableID" vorliegen. Dann wird nur die ID und kein Name geliefert. "dwVarNameCharCount" wird nicht berücksichtigt. Wenn beide Zeiger "NULL" sind, wird "DM_E_PARAM" zurückgegeben.

Wenn der innere Zeiger "NULL" ist, wird der Puffer allokiert und zurückgegeben. Die Größe wird in "lpdwVarNameCharCount" zurückgegeben.

lpdwVarNameCharCount

Zeiger auf ein "DWORD", in dem die Größe des Rückgabepuffers abgelegt wird. Die Größe müssen Sie groß genug wählen, damit der Variablenname mit Nullterminierung abgelegt werden kann.

Wenn der Puffer zu klein ist, wird der abgeschnittene Name abgelegt und der Fehler "DM_E_OOM" zurückgegeben.

lpdwVariableID

Zeiger auf ein "DWORD", in dem die VariablenID abgelegt wird.

Wenn "NULL" angegeben ist, muss ein gültiger Zeiger auf "lpdzVariableName" und eine gültige Größe in "dwVarNameCharCount" vorliegen. Dann wird nur der Variablenname zurückgegeben.

Die Rückgabe der VariablenID kann nicht immer gewährleistet werden, z. B. PackageVariable, bestimmte S7-Variablen usw.. In diesen Fällen wird "0" geliefert.

Bedingt kann dann bei Bedarf die Information über einen Folgeaufruf von "DMGetVarInfoExStr" ergänzt werden.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Dialog mit "OK" verlassen.

FALSE

Fehler oder Dialog mit "ABBRECHEN" verlassen.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INV_PRJ	Angegebenes Projekt nicht gefunden / geladen
DM_E_PARAM	Parameter nicht konsistent
DM_E_OOM	Speicherfehler: der Puffer "lpdzVariableName" ist zu klein in "dwVarNameCharCount" angegeben und der Name wurde beschnitten

Benötigte Files

dmclient_exstr.h
 dmclient.lib
 dmclient.dll

Verwandte Funktionen

DMSHOWVARDATABASEMULTIEXSTR	Variablen-Auswahldialog öffnen
-----------------------------	--------------------------------

Beispiele

Skript-Beispiel "Button DMSHOWVARDATABASEEXSTR":

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
    BOOL bRet = FALSE;
    CMN_ERROR err;
    CHAR szProjectName[256];
    HWND hwndParent;
    CHAR szVariableName[256];
    LPSTR pszVariableName;
    DWORD dwVarNamCharCount = 256;
    DWORD dwVarID = 0L;

    szProjectName[0] = 0;
    szVariableName[0] = 0;
    hwndParent = NULL;
    pszVariableName= szVariableName;

    memset(&err, 0, sizeof(err));

    printf("\r\n\r\n##### enter Test with DMSHOWVARDATABASEEXSTR
#####");

    bRet = DMGetRuntimeProject(szProjectName, 256, &err);
    if (!bRet)
    {
        printf("\r\n error DMGetRuntimeProject =>[%s], err=%ld,%ld,%ld,
%ld,%ld,[%s]", szProjectName,
            err.dwError1, err.dwError2, err.dwError3, err.dwError4,
            err.dwError5, err.szErrorText);
    }

    bRet = DMSHOWVARDATABASEEXSTR(szProjectName, hwndParent, NULL, NULL,
        &pszVariableName, &dwVarNamCharCount,
        &dwVarID, &err);
    if (!bRet)
    {
```

```

        printf("\r\n  error DMSHOWVARDATABASEEXSTR: err=%ld,%ld,%ld,%ld,
%ld,[%s]",
            err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
    }
else
{
    printf("\r\n  DMSHOWVARDATABASEESSTR OK! (set to first line)");
    printf("\r\n  szVarName=[%s], dwVarID=[%ld]", szVariableName,
dwVarID);
    SetTagChar("szVarKeyName_1",szVariableName);
    SetTagDWord("dwVarKeyID_1",dwVarID);
}
}

```

2.2.6.34 DMSHOWVARDATABASEMULTI (RT Professional)

Verwendung

Die Funktion öffnet für das spezifizierte Projekt den Variablen-Auswahldialog. Im Gegensatz zu ShowVarDatabase können mehrere Variablen selektiert werden.

Deklaration

```

BOOL DMSHOWVARDATABASEMULTI (
    LPCSTR                lpszProjectFile,
    HWND                  hwndParent,
    LPDM_DLGOPTIONS      lpdmOptions,
    LPDM_VARFILTER       lpdmFilter,
    LPDWORD               lpdwItems,
    DM_NOTIFY_SELECT_VAR_PROC lpfnVariables,
    LPVOID                lpvUser,
    LPCMN_ERROR           lpdmError);

```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in RT mit DMGetRuntimeProject ermittelt.

hwndParent

Handle auf das Fenster, das als Vaterfenster für den Dialog benutzt wird.

lpdmOptions

Zeiger auf die Struktur DM_DLGOPTIONS (Seite 219) mit Vorgaben, wie sich der Dialog verhalten soll. Wenn NULL, wird der Standarddialog verwendet.

lpdmFilter

Zeiger auf die Filterstruktur DM_VARFILTER (Seite 233). Wenn NULL, werden alle Variablen angezeigt.

Nur das Filtern nach Variablenname und Variablentyp wird unterstützt. Das Filtern von Variablen nach Gruppen und Verbindungen ist nicht implementiert.

lpdwItems

Zeiger auf einen DWORD-Puffer, der die Anzahl der selektierten Variablen aufnimmt.

lpfnVariables

Zeiger auf Ihre Callback-Funktion, die für jede selektierte Variable aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Dialog mit "OK" verlassen.

FALSE

Fehler oder Dialog mit "ABBRECHEN" verlassen.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INV_PRJ	angegebenes Projekt nicht gefunden / geladen

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMShowVarDatabase (Seite 410)	Variablen-Auswahldialog öffnen
DM_NOTIFY_SELECT_VAR_PROC (Seite 421)	Variablenauswahldialog öffnen (Callback)

Siehe auch

DM_VARFILTER (Seite 233)

DMShowVarDatabase (Seite 410)

DM_DLGOPTIONS (Seite 219)

DM_NOTIFY_SELECT_VAR_PROC (Seite 421)

2.2.6.35 DMShowVarDatabaseMultiExStr (RT Professional)

Deklaration

```

BOOL DMShowVarDatabaseMultiExStr (
    LPCTSTR                lpszProjectFile,
    HWND                   hwndParent,
    LPDM_DLGOPTIONS        lpdmOptions,
    LPDM_VARFILTER         lpdmFilter,
    LPDWORD                 lpdwItems,
    DM_NOTIFY_SELECT_VAR_PROC_EXSTR lpfnVariables,
    LPVOID                  lpvUser,
    LPCMN_ERROR             lpdmError);

```

Beschreibung

Die Funktion öffnet für das spezifizierte Projekt den Variablen-Auswahldialog. Im Gegensatz zu "ShowVarDatabaseExStr" können mehrere Variablen selektiert werden.

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei kann mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermittelt werden.

hwndParent

Handle auf das Fenster, das als Vaterfenster für den Dialog benutzt wird.

lpdmOptions

Zeiger auf die Struktur "DM_DLGOPTIONS" mit Vorgaben, wie sich der Dialog verhält, wenn "NULL" ist gleich Standarddialog.

lpdmFilter

Zeiger auf die Filterstruktur "DM_VARFILTER". Bei "NULL" werden alle Variablen angezeigt. Unterstützt wird nur das Filtern nach Variablenname und Variablentyp. Das Filtern von Variablen nach Gruppen und Verbindungen ist nicht implementiert.

lpdwItems

Zeiger auf einen DWORD-Puffer, der die Anzahl der selektierten Variablen in Summe aufnimmt.

lpfnVariables

Zeiger auf die Callback-Funktion, die für jede selektierte Variable aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Dialog mit "OK" verlassen.

FALSE

Fehler oder Dialog mit "ABBRECHEN" verlassen.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INV_PRJ	angegebenes Projekt nicht gefunden / geladen

Benötigte Files

- dmclient_exstr.h
- dmclient.lib
- dmclient.dll

Verwandte Funktionen

DMSHOWVARDATABASEEXSTR	Variablen-Auswahldialog öffnen
DM_NOTIFY_SELECT_VAR_PROC_EXSTR	Variablenauswahldialog öffnen (Callback)

Beispiele

Skript-Beispiel "Button DMSHOWVARDATABASEMULTIEXSTR":

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
extern BOOL  DM_NotifySelectVarProcA(LPCSTR lpszVariableName, DWORD
dwVariableID, LPVOID lpvUser);

BOOL bRet = FALSE;
CMN_ERROR err;
CHAR szProjectName[256];
HWND hwndParent;
DWORD dwItems;
DWORD dwInxDecr;

szProjectName[0] = 0;
hwndParent = NULL;
memset(&err, 0, sizeof(err));
dwItems = 0;
dwInxDecr = 4; /*for decrement index to save in DM tags from callback
set to lpvUser*/

printf("\r\n\r\n##### enter Test with
DMSHOWVARDATABASEMULTIEXSTR #####");

bRet = DMGetRuntimeProject(szProjectName, 256, &err);
if (!bRet)
{
printf("\r\n  error DMGetRuntimeProject =>[%s], err=%ld,%ld,%ld,
%ld,%ld, [%s]", szProjectName,
err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}

bRet = DMSHOWVARDATABASEMULTIEXSTR(szProjectName, hwndParent, NULL,
NULL,
                                &dwItems, DM_NotifySelectVarProcA,
&dwInxDecr, &err);
if (!bRet)
{
printf("\r\n  error DMSHOWVARDATABASEMULTIEXSTR: err=%ld,%ld,%ld,
%ld,%ld, [%s]",
err.dwError1, err.dwError2, err.dwError3, err.dwError4,
err.dwError5, err.szErrorText);
}
else
{
printf("\r\n  DMSHOWVARDATABASEEXSTRMULTI OK!");
printf("\r\n  [%ld] Var's selected", dwItems);
}
}
```

```
printf("\r\n##### exit Test with DMShowVarDatabaseMultiExStr
#####\r\n");
}
```

Skript-Beispiel "Project-Function DM_NotifySelectVarProcA":

```
BOOL DM_NotifySelectVarProcA(LPCSTR lpszVariableName, DWORD
dwVariableID, LPVOID lpvUser)
{
    BOOL bRet = TRUE;
    DWORD* pdwInxDecr = NULL;

    pdwInxDecr = (DWORD*)lpvUser;

    printf("\r\n\r\n### DM_NotifySelectVarProc entry: ###");
    printf("\r\n    VarName=[%s], VarID=[%ld], lpvUser=[0x%08lx]",
    lpszVariableName, dwVariableID, lpvUser);
    if (pdwInxDecr)
    {
        // handle max 4 outputs, ignore all others
        if(4 == *pdwInxDecr )
        {
            SetTagChar("szVarKeyName_4",lpszVariableName);
            SetTagDWord("dwVarKeyID_4",dwVariableID);
            printf("\r\n        save in szVarKeyName_4 and dwVarKeyID_4");
        }
        if(3 == *pdwInxDecr )
        {
            SetTagChar("szVarKeyName_3",lpszVariableName);
            SetTagDWord("dwVarKeyID_3",dwVariableID);
            printf("\r\n        save in szVarKeyName_3 and dwVarKeyID_3");
        }
        if(2 == *pdwInxDecr )
        {
            SetTagChar("szVarKeyName_2",lpszVariableName);
            SetTagDWord("dwVarKeyID_2",dwVariableID);
            printf("\r\n        save in szVarKeyName_2 and dwVarKeyID_2");
        }
        if(1 == *pdwInxDecr )
        {
            SetTagChar("szVarKeyName_1",lpszVariableName);
            SetTagDWord("dwVarKeyID_1",dwVariableID);
            printf("\r\n        save in szVarKeyName_1 and dwVarKeyID_1");
        }
    }
    if (pdwInxDecr && (0L < *pdwInxDecr))
    {
        *pdwInxDecr = *pdwInxDecr - 1L;
        printf("\r\n        Inx=%ld", *pdwInxDecr);
    }

    printf("\r\n### DM_NotifySelectVarProc exit ###\r\n");
}
```

```
return bRet;  
}
```

2.2.6.36 DM_NOTIFY_SELECT_VAR_PROC (RT Professional)

Beschreibung

Um mit der Funktion `DMSHOWVARDATABASEMULTI` im Auswahldialog angewählten Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ `DM_NOTIFY_SELECT_VAR_PROC` bereitstellen.

Deklaration

```
BOOL ( * DM_NOTIFY_SELECT_VAR_PROC ) (  
    LPDM_VARKEY    lpdmVarKey,  
    DWORD          dwItem,  
    LPVOID         lpvUser );
```

Parameter

lpdmVarKey

Zeiger auf die erste von Strukturen vom Typ `DM_VARKEY` (Seite 241) mit dem Namen und der Kennung einer Variablen.

dwItem

Anzahl der in `lpdmVarKey` übergebenen Strukturen.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMSHOWVARDATABASEMULTI (Seite 415)	Variablen-Auswahldialog öffnen
------------------------------------	--------------------------------

Siehe auch

DM_VARKEY (Seite 241)

DMSHOWVARDATABASEMULTI (Seite 415)

2.2.6.37 GAPICreateNewVariable (RT Professional)

Verwendung

Erzeugt eine neue Variable oder prüft, ob eine Variable existiert. Diese Funktion kann nur auf temporäre Variablen, \$-Variablen, angewendet werden.

Deklaration

```
BOOL GAPICreateNewVariable (  
    LPMCP_NEWVARIABLE_DATA    pData,  
    LPCMN_ERROR                lpdmError);
```

Parameter

pData

Zeiger auf die Struktur MCP_NEWVARIABLE_DATA (Seite 245) mit den Daten der Variablen.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Variable erzeugt.

Beim Prüfen: Variable noch nicht vorhanden.

FALSE

Fehler.

Beim Prüfen mit Errorcode1 = DM_E_ALREADY_EXIST: Variable vorhanden

Bemerkung

Es gibt weiter entwickelte Funktionen.

GAPICreateNewVariable4
GAPICreateNewVariableEx4
GAPICreateNewVariable5

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_ALREADY_EXIST	Anzulegendes Objekt existiert bereits

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

GAPICreateNewVariable4 (Seite 424)	Variable erzeugen
GAPICreateNewVariableEx4 (Seite 427)	Variable erzeugen mit Creator-ID
GAPICreateNewVariable5 (Seite 425)	Variable erzeugen mit Creator-ID

Siehe auch

MCP_NEWVARIABLE_DATA (Seite 245)

GAPICreateNewVariable4 (Seite 424)

GAPICreateNewVariable5 (Seite 425)

GAPICreateNewVariableEx4 (Seite 427)

2.2.6.38 GAPICreateNewVariable4 (RT Professional)

Verwendung

Erzeugt eine neue Variable oder prüft, ob eine Variable existiert. Diese Funktion kann nur auf temporäre Variablen, \$-Variablen, angewendet werden.

Diese Funktion unterscheidet sich von GAPICreateNewVariable durch die mögliche Angabe von Skalierungsdaten.

Deklaration

```
BOOL GAPICreateNewVariable4 (  
    LPMCP_NEWVARIABLE_DATA_4    pData,  
    LPCMN_ERROR                 lpdmError);
```

Parameter

pData

Zeiger auf die Struktur MCP_NEWVARIABLE_DATA_4 (Seite 247) mit den Daten der Variablen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variable erzeugt.

Beim Prüfen: Variable noch nicht vorhanden.

FALSE

Fehler.

Beim Prüfen mit ErrorCode1 = DM_E_ALREADY_EXIST: Variable vorhanden.

Bemerkung

Es gibt weiter entwickelte Funktionen.

GAPICreateNewVariableEx4	Variable erzeugen mit Creator-ID
GAPICreateNewVariable5	Variable erzeugen mit Creator-ID

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_ALREADY_EXIST	anzulegendes Objekt existiert bereits

Benötigte Dateien

dmclient.h
 dmclient.lib
 dmclient.dll

Verwandte Funktionen

GAPICreateNewVariable (Seite 422)	Variable erzeugen
GAPICreateNewVariableEx4 (Seite 427)	Variable erzeugen mit Creator-ID
GAPICreateNewVariable5 (Seite 425)	Variable erzeugen mit Creator-ID

Siehe auch

GAPICreateNewVariable (Seite 422)
 MCP_NEWVARIABLE_DATA_4 (Seite 247)
 GAPICreateNewVariable5 (Seite 425)
 GAPICreateNewVariableEx4 (Seite 427)

2.2.6.39 GAPICreateNewVariable5 (RT Professional)**Verwendung**

Erzeugt eine neue Variable oder prüft, ob eine Variable existiert. Diese Funktion kann nur auf temporäre Variablen, \$-Variablen, angewendet werden.

Diese Funktion unterscheidet sich von GAPICreateNewVariable4 durch die Angabe der Erzeugerkennung und dass die Start- und Ersatzwerte auch für Textvariablen angegeben werden können.

Deklaration

```
BOOL WINAPI CreateNewVariable5 (
    DWORD dwCreatorID,
    LPMCP_NEWVARIABLE_DATA_5 pData,
    LPCMN_ERROR lpdmError);
```

Parameter

dwCreatorID

In der Erzeugerkennung können Sie Angaben treffen, von wem ein Objekt angelegt wurde. Die Werte 0 – 10100 sowie 11000 - 11100 sind intern oder für bestimmte Systeme reserviert.

pData

Zeiger auf die Struktur MCP_NEWVARIABLE_DATA_5 (Seite 249) mit den Daten der Variablen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variable erzeugt.
Beim Prüfen: Variable noch nicht vorhanden.

FALSE

Fehler.
Beim Prüfen mit ErrorCode1 = DM_E_ALREADY_EXIST: Variable vorhanden.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_ALREADY_EXIST	Anzulegendes Objekt existiert bereits

Benötigte Dateien

- dmclient.h
- dmclient.lib
- dmclient.dll

Verwandte Funktionen

GAPICreateNewVariable (Seite 422)	Variable erzeugen
GAPICreateNewVariable4 (Seite 424)	Variable erzeugen
GAPICreateNewVariableEx4 (Seite 427)	Variable erzeugen mit Creator-ID

Siehe auch

[GAPICreateNewVariable \(Seite 422\)](#)
[GAPICreateNewVariable4 \(Seite 424\)](#)
[MCP_NEWVARIABLE_DATA_5 \(Seite 249\)](#)
[GAPICreateNewVariableEx4 \(Seite 427\)](#)
[MCP_VARIABLE_LIMITS5 \(Seite 262\)](#)

2.2.6.40 GAPICreateNewVariableEx4 (RT Professional)

Verwendung

Erzeugt eine neue Variable oder prüft, ob eine Variable existiert. Diese Funktion kann nur auf temporäre Variablen, \$-Variablen, angewendet werden.

Diese Funktion unterscheidet sich von GAPICreateNewVariable4 durch die Angabe der Erzeugerkennung.

Deklaration

```

BOOL GAPICreateNewVariableEx4 (
    DWORD                dwCreatorID,
    LPMCP_NEWVARIABLE_DATA_4  pData,
    LPCMN_ERROR          lpdmError);

```

Parameter

dwCreatorID

In der Erzeugerkennung können Sie Angaben treffen, von wem ein Objekt angelegt wurde. Die Werte 0 – 10100 sowie 11000 - 11100 sind intern oder für bestimmte Systeme reserviert.

pData

Zeiger auf die Struktur MCP_NEWVARIABLE_DATA_4 (Seite 247) mit den Daten der Variablen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variable erzeugt.

Beim Prüfen: Variable noch nicht vorhanden.

FALSE

Fehler.

Beim Prüfen mit ErrorCode1 = DM_E_ALREADY_EXIST: Variable vorhanden.

Bemerkung

Es gibt eine weiter entwickelte Funktion.

GAPICreateNewVariable5

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_ALREADY_EXIST	anzulegendes Objekt existiert bereits

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

GAPICreateNewVariable (Seite 422)	Variable erzeugen
GAPICreateNewVariable4 (Seite 424)	Variable erzeugen
GAPICreateNewVariable5 (Seite 425)	Variable erzeugen mit Creator-ID

Siehe auch

GAPICreateNewVariable (Seite 422)

GAPICreateNewVariable4 (Seite 424)

MCP_NEWVARIABLE_DATA_4 (Seite 247)

GAPICreateNewVariable5 (Seite 425)

2.2.7 Funktionen zur Bearbeitung von strukturierten Variablen (RT Professional)

2.2.7.1 GAPIEnumTypeMembers (RT Professional)

Verwendung

Die Funktion liefert die Namen der zu einer strukturierten Variablen gehörigen Variablen.

Deklaration

```
BOOL GAPIEnumTypeMembers (
    LPCSTR                lpszProjectFile,
    LPCSTR                lpszTypeName,
    DM_ENUM_TYPEMEMBERS_PROC lpfnCallback,
    LPVOID                lpvUser,
    LPCMN_ERROR           lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler (DM_E_NOT_CONNECTED) abgewiesen.

lpszTypeName

Name des Typs einer strukturierten Variablen, deren Variablen aufgelistet werden sollen.

lpfnCallback

Zeiger auf Ihre Callback Funktion, die für jede Variable aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablen einer strukturierten Variablen auflisten.

FALSE

Fehler.

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_TPEMEMBERS_PROC (Seite 430)	Variablen einer strukturierten Variablen auflisten (Callback)
GAPIEnumTypeMembersEx (Seite 432)	Variablen einer strukturierten Variablen auflisten
GAPIEnumTypeMembersEx4 (Seite 436)	Variablen einer strukturierten Variablen auflisten

Siehe auch

DM_ENUM_TPEMEMBERS_PROC (Seite 430)

GAPIEnumTypeMembersEx (Seite 432)

GAPIEnumTypeMembersEx4 (Seite 436)

2.2.7.2 DM_ENUM_TPEMEMBERS_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Namen von Variablen auswerten zu können, müssen Sie eine Callback Funktion vom Typ DM_ENUM_TPEMEMBERS_PROC bereitstellen.

Deklaration

```
BOOL ( * DM_ENUM_TPEMEMBERS_PROC ) (
    LPCSTR    lpszMemberName,
    LPVOID    lpvUser );
```

Parameter

IpszStructTypeName

Zeiger auf den Namen der zu einer strukturierten Variablen gehörenden Variablen.

IpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - ODK-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

GAPIEnumTypeMembers (Seite 429)	Variablen einer strukturierten Variablen auflisten
---------------------------------	--

Siehe auch

GAPIEnumTypeMembers (Seite 429)

2.2.7.3 GAPIEnumTypeMembersEx (RT Professional)

Verwendung

Die Funktion liefert eine Beschreibung, inklusive aller Vorbesetzungen der zu einer strukturierten Variable gehörenden Variablen, jedoch ohne Skalierungsdaten.

Wenn Sie auch die Skalierungsdaten auslesen wollen, benötigen Sie die Funktion GAPIEnumTypeMembersEx4.

Deklaration

```
BOOL GAPIEnumTypeMembersEx (
    LPCSTR                lpszProjectFile,
    LPCSTR                lpszTypeName,
    DM_ENUM_TYPEMEMBERS_PROC_EX lpfnCallback,
    LPVOID                lpvUser,
    LPCMN_ERROR           lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler (DM_E_NOT_CONNECTED) abgewiesen.

lpszTypeName

Name des Typs einer strukturierten Variablen, deren Variablen aufgelistet werden sollen.

Auf Clients können nur lokale Variablentypen enumeriert werden.

lpfnCallback

Zeiger auf Ihre Callback Funktion, die für jede Variable aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Variablen einer strukturierten Variablen aufgelistet.

FALSE

Fehler.

Bemerkung

Es gibt eine weiter entwickelte Funktion `GAPIEnumTypeMembersEx4`.

Benötigte Dateien

`dmclient.h`

`dmclient.lib`

`dmclient.dll`

Verwandte Funktionen

<code>DM_ENUM_TYPEMEMBERS_PROC_EX</code> (Seite 435)	Variablen einer strukturierten Variablen auflisten (Call-back)
<code>GAPIEnumTypeMembers</code> (Seite 429)	Variablen einer strukturierten Variablen auflisten
<code>GAPIEnumTypeMembersEx4</code> (Seite 436)	Variablen einer strukturierten Variablen auflisten

Siehe auch

`GAPIEnumTypeMembers` (Seite 429)

`DM_ENUM_TYPEMEMBERS_PROC_EX` (Seite 435)

`GAPIEnumTypeMembersEx4` (Seite 436)

2.2.7.4 GAPIEnumTypeMembersExStr (RT Professional)**Deklaration**

```

BOOL GAPIEnumTypeMembersExStr (
    LPCTSTR                                lpszProjectFile,
    LPCTSTR                                lpszTypeName,
    DM_ENUM_TYPEMEMBERS_PROC_EXSTR        lpfnCallback,
    LPVOID                                  lpvUser,
    LPCMN_ERROR                            lpdmError);

```

Beschreibung

Die Funktion liefert eine Beschreibung inklusive aller Vorbesetzungen der zu einer strukturierten Variable gehörenden Variablen, jedoch ohne Skalierungsdaten.

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei kann mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermittelt werden.

Ab der WinCC Version V5.0 SP2 wird bei Angabe eines Leerstrings ein internes "DMEnumOpenedProjects" auf das aktuell geöffnete Projekt durchgeführt.

In Runtime dürfen Sie nur das aktuell geöffnete Projekt angeben. Andere Angaben werden ab WinCC V5.0 SP2 mit Fehler "DM_E_NOT_CONNECTED" abgewiesen.

lpszTypeName

Name des Typs einer strukturierten Variablen, deren Variablen aufgelistet werden.

Auf Multiclients (V5) bzw. Clients (V6) können nur lokale Variablentypen enumeriert werden.

lpfnCallback

Zeiger auf die Callback-Funktion, die für jede Variable aufgerufen wird.

lpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablen einer strukturierten Variablen aufgelistet

FALSE

Fehler

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_TYPEMEMBERS_PROC_EXSTR	Variablen einer strukturierten Variablen auflisten (Callback)
GAPIEnumTypeMembersEx4	Variablen einer strukturierten Variablen auflisten

2.2.7.5 DM_ENUM_TYPEMEMBERS_PROC_EX (RT Professional)

Beschreibung

Um die vom System aufgelisteten Beschreibungen von Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_TYPEMEMBERS_PROC_EX bereitstellen.

Deklaration

```

BOOL ( * DM_ENUM_TYPEMEMBERS_PROC_EX) (
    LPDM_VARKEY                lpdmVarKey,
    LPMCP_NEWVARIABLE_DATA_EX  lpdmVarDataEx,
    LPVOID                      lpvUser );

```

Parameter

lpdmVarKey

Zeiger auf die erste der Strukturen vom Typ AUTOHOTSPOT mit den Schlüsseln (ID und Name) einer Variablen

lpdmVarDataEx

Zeiger auf eine Struktur vom Typ MCP_NEWVARIABLE_DATA_EX (Seite 252) mit der Beschreibung einer Variablen.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Benötigte Dateien

dmclient.h

Verwandte Funktionen

GAPIEnumTypeMembersEx (Seite 432)	Variablen einer strukturierten Variablen auflisten
-----------------------------------	--

Siehe auch

GAPIEnumTypeMembersEx (Seite 432)

MCP_NEWVARIABLE_DATA_EX (Seite 252)

2.2.7.6 GAPIEnumTypeMembersEx4 (RT Professional)

Verwendung

Die Funktion liefert eine komplette Beschreibung, inklusive aller Vorbesetzungen der zu einer strukturierten Variable gehörenden Variablen.

Deklaration

```
BOOL GAPIEnumTypeMembersEx4 (
    LPCSTR                lpszProjectFile,
    LPCSTR                lpszTypeName,
    DM_ENUM_TPEMEMBERS_PROC_EX4 lpfnCallback,
    LPVOID               lpvUser,
    LPCMN_ERROR          lpdmError);
```

Parameter

IpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden (DM_E_NOT_CONNECTED) abgewiesen.

IpszTypeName

Name des Typs einer strukturierten Variablen, deren Variablen aufgelistet werden sollen.

Auf Clients können nur lokale Variablentypen enumeriert werden.

Ipfncallback

Zeiger auf Ihre Callback Funktion, die für jede Variable aufgerufen wird.

IpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablen einer strukturierten Variablen aufgelistet.

FALSE

Fehler.

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_TYPEMEMBERS_PROC_EX4 (Seite 438)	Variablen einer strukturierten Variablen auflisten (Callback)
GAPIEnumTypeMembers (Seite 429)	Variablen einer strukturierten Variablen auflisten

Siehe auch

- GAPIEnumTypeMembers (Seite 429)
- GAPIEnumTypeMembersEx (Seite 432)
- DM_ENUM_TYPEMEMBERS_PROC_EX4 (Seite 438)

2.2.7.7 DM_ENUM_TYPEMEMBERS_PROC_EX4 (RT Professional)

Beschreibung

Um die vom System aufgelisteten Beschreibungen von Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_TYPEMEMBERS_PROC_EX4 bereitstellen. Sie unterscheidet sich von DM_ENUM_TYPEMEMBERS_PROC_EX durch die zusätzliche Ausgabe der Skalierungsdaten.

Deklaration

```
BOOL ( * DM_ENUM_TYPEMEMBERS_PROC_EX4 ) (
    LPDM_VARKEY                lpdmVarKey,
    LPMCP_NEWVARIABLE_DATA_EX4 lpdmVarDataEx,
    LPVOID                      lpvUser );
```

Parameter

lpdmVarKey

Zeiger auf die erste der Strukturen vom Typ DM_VARKEY (Seite 241) mit den Schlüsseln (ID und Name) einer Variablen

lpdmVarDataEx

Zeiger auf eine Struktur vom Typ MCP_NEWVARIABLE_DATA_EX4 (Seite 254) mit der Beschreibung einer Variablen.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

GAPIEnumTypeMembersEx4 (Seite 436)	Variablen einer strukturierten Variablen auflisten
------------------------------------	--

Siehe auch

GAPIEnumTypeMembersEx4 (Seite 436)

DM_VARKEY (Seite 241)

MCP_NEWVARIABLE_DATA_EX4 (Seite 254)

2.2.7.8 GAPIEnumTypes (RT Professional)

Verwendung

Mit der Funktion werden die Namen und ID-Nummern der projizierten strukturierten Variablentypen aufgelistet.

Deklaration

```
BOOL GAPIEnumTypes (
    LPCSTR          lpszProjectFile,
    DM_ENUM_TYPES_PROC lpfnCallback,
    LPVOID          lpvUser,
    LPCMN_ERROR     lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit DMEnumOpenedProjects oder in Runtime mit DMGetRuntimeProject ermittelt.

Bei Angabe eines Leerstrings wird ein internes DMEnumOpenedProjects auf das aktuell geöffnete Projekt durchgeführt.

In Runtime darf nur das aktuell geöffnete Projekt angegeben werden. Andere Angaben werden mit Fehler (DM_E_NOT_CONNECTED) abgewiesen.

lpfnCallback

Zeiger auf Ihre Callback-Funktion, welche die Daten der Variablentypen entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Typen von strukturierten Variablen aufgelistet.

FALSE

Fehler.

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_TYPES_PROC (Seite 441)	Typen von strukturierten Variablen auflisten (Call-back)
--------------------------------	--

Beispiele

Enumerate all structured types (Seite 483) "DM01.cpp"

Siehe auch

DM_ENUM_TYPES_PROC (Seite 441)

Enumerate all structured types (Seite 483)

2.2.7.9 DM_ENUM_TYPES_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Variablentypen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_TYPES_PROC bereitstellen.

Deklaration

```

BOOL ( * DM_ENUM_TYPES_PROC) (
    LPCSTR    lpszTypeName,
    DWORD     dwTypeID,
    DWORD     dwCreatorID,
    LPVOID    lpvUser );

```

Parameter

lpszTypeName

Zeiger auf den Namen des Variablentyps.

dwTypeID

dwTypeID entspricht der Kennung wie sie von GAPICreateType für den Variablentyp vergeben wurde.

dwCreatorID

In der Erzeugerkennung können Sie Angaben treffen, von wem ein Objekt angelegt wurde.

Die Werte 0 – 10100 sowie 11000 - 11100 sind intern oder für bestimmte Systeme reserviert.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

GAPIEnumTypes (Seite 439)	Typen von strukturierten Variablen auflisten
---	--

Beispiele

Enumerate all structured types (Seite 483) "DM01.cpp"

Siehe auch

[GAPIEnumTypes \(Seite 439\)](#)

[Enumerate all structured types \(Seite 483\)](#)

2.2.8 Funktionen zur Bearbeitung von Verbindungen (RT Professional)

2.2.8.1 DMEnumConnectionData (RT Professional)

Verwendung

Die Funktion ermittelt die Daten einer projektierten logischen Verbindung.

Deklaration

```
BOOL DMEnumConnectionData (
    LPCSTR                lpszProjectFile,
    LPDM_CONNKEY          lpdmConnKey,
    DWORD                 dwItems,
    DM_ENUM_CONNECTION_PROC lpfnCallback,
    LPVOID                 lpvUser,
    LPCMN_ERROR            lpdmError);
```

Parameter

lpszProjectFile

Zeiger auf den Namen der Projektdatei, einschließlich Pfad und Extension.

Der Name der Projektdatei wird mit `DMEnumOpenedProjects` oder in Runtime mit `DMGetRuntimeProject` ermittelt.

lpdmConnKey

Zeiger auf die erste der Strukturen vom Typ `DM_CONNKEY` (Seite 215). Über diese Strukturen werden die logischen Verbindungen spezifiziert, deren Daten enumeriert werden.

dwItems

Anzahl der logischen Verbindungen deren Daten aufgelistet werden.

Mit dem Wert 0 wird eine Auflistung aller Verbindungen angestoßen.

lpfnCallback

Zeiger auf Ihre Callback-Funktion, welche die Daten der logischen Verbindung entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Daten aufgelistet.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_CONNECTION_PROC (Seite 446)	Verbindungsdaten auflisten (Call-back)
-------------------------------------	--

Beispiele

Enumerate all connections (Seite 485) "DM01.cpp"

OnTestEnumConnectionDataAll (Seite 490) "TESTCDoc.cpp"

Siehe auch

DM_CONNKEY (Seite 215)

DM_ENUM_CONNECTION_PROC (Seite 446)

Enumerate all connections (Seite 485)

OnTestEnumConnectionDataAll (Seite 490)

2.2.8.2 DMLenumConnectionDataExStr (RT Professional)

Deklaration

```
BOOL DMLenumConnectionDataExStr (
    LPCTSTR                lpszProjectFile,
    DM_ENUM_CONNECTION_PROC_EXSTR lpfnCallback,
    LPVOID                 lpvUser,
    LPDWORD                lpdwConnectionCount,
    LPCMN_ERROR            lpdmError);
```

Beschreibung

Mit der Funktion werden alle projektierten logischen Verbindung ermittelt.

Parameter**IpszProjectFile**

Zeiger auf den Namen der Projektdatei mit Pfad und Extension.

Der Name der Projektdatei kann mit "DMEnumOpenedProjects" oder in Runtime mit "DMGetRuntimeProject" ermittelt werden.

lpfnCallback

Zeiger auf die Callback-Funktion, die die Daten der logischen Verbindung entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdwConnectionCount

Zeiger auf ein "DWORD", in dem die Anzahl der vorhandenen Verbindungen zurückgegeben wird.

Mit "lpfnCallback = NULL" wird so erstmal ermittelt, wieviel Speicher für die Verbindungen benötigt wird.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Daten aufgelistet

FALSE

Fehler

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DM_ENUM_CONNECTION_PROC_EXSTR	Verbindungsdaten auflisten (Callback)
-------------------------------	---------------------------------------

2.2.8.3 DM_ENUM_CONNECTION_PROC (RT Professional)

Beschreibung

Um die vom System aufgelisteten Daten einer logischen Verbindung auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_ENUM_CONNECTION_PROC bereitstellen.

Deklaration

```
BOOL ( * DM_ENUM_CONNECTION_PROC) (  
    LPDM_CONNECTION_DATA    lpdmConData,  
    LPVOID                   lpvUser);
```

Parameter

lpdmConData

Zeiger auf eine Struktur vom Typ DM_CONNECTION_DATA (Seite 214) in der die Daten einer logischen Verbindung hinterlegt sind.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMEnumConnectionData (Seite 443)	Verbindungsdaten auflisten
----------------------------------	----------------------------

Beispiele

Enumerate all connections (Seite 485) "DM01.cpp"

Siehe auch

DM_CONNECTION_DATA (Seite 214)

DMEnumConnectionData (Seite 443)

Enumerate all connections (Seite 485)

2.2.9 Funktionen zur Arbeitsplattform (RT Professional)**2.2.9.1 DMGetOSVersion (RT Professional)****Verwendung**

Ermittelt das auf dem PC eingesetzte Betriebssystem.

Deklaration

```
DWORD DMGetOSVersion (
    VOID);
```

Parameter

Keine.

Rückgabewert

Der Rückgabewert kennzeichnet das verwendete Betriebssystem:

DM_OS_UNKNOWN	0	unbekanntes Betriebssystem
DM_OS_NT	1	Windows NT
DM_OS_32S	2	Windows 3.x mit Win 32s

DM_OS_CHICAGO	3	Windows 95
DM_OS_2000	4	Windows 2000
DM_OS_XP	5	Windows XP
DM_OS_2003	6	Windows 2003

Benötigte Files

dmclient.h
dmclient.lib
dmclient.dll

2.2.9.2 DMGetSystemLocale (RT Professional)

Verwendung

Liefert den Code der derzeit verwendeten Projektierungssprache zurück.

Deklaration

```
BOOL DMGetSystemLocale (  
    LPDWORD      lpdwLocaleID,  
    LPCMN_ERROR  lpdmError);
```

Parameter

lpdwLocaleID

Zeiger auf den Code der derzeit eingestellten Sprache.

Mögliche Rückgabewerte sind die Sprachcodes von:

Deutsch	0x0407
Englisch	0x0409
Französisch	0x040C

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktuell eingestellte Projektierungssprache ermittelt.

FALSE

Fehler

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Beispiele

OnTestSystemLocale (Seite 496) "TESTCDoc.cpp"

Siehe auch

OnTestSystemLocale (Seite 496)

2.2.9.3 DMSetLanguage (RT Professional)**Verwendung**

Veranlasst die Umschaltung der Projektierungssprache.

Deklaration

```

BOOL DMSetLanguage (
    DWORD          dwLocaleID,
    LPCMN_ERROR    lpdmError );

```

Parameter**dwLocaleID**

Zeiger auf den Code der einzustellenden Sprache. Mögliche Werte sind die Sprachcodes von:

Deutsch	0x0407
Englisch	0x0409
Französisch	0x040C

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Sprache umgeschaltet.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_SUPPORTED	Angeforderter Dienst nicht verfügbar
--------------------	--------------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

2.2.9.4 DMShowLanguageDialog (RT Professional)

Verwendung

Die Funktion öffnet eine Dialogbox zur Sprachauswahl aus.

Deklaration

```
BOOL DMShowLanguageDialog (  
    HWND          hwndParent,  
    DWORD         dwFlags,  
    DWORD         dwSetLocaleIDs[],  
    UINT          uSetIDArraySize,  
    LPDWORD       lpdwGetLocaleID,  
    LPCMN_ERROR   lpdmError);
```

Parameter

hwndParent

Handle auf das Fenster, das als Vaterfenster für den Dialog benutzt wird.

dwFlags

dwFlags kennzeichnet die Sprachen, die im Dialog angezeigt werden und ausgewählt werden können:

DM_LANGDLG_REMOVE	gesetzt	Außer den Sprachen im Array dwSetLocaleIDs werden alle Sprachen die mit EnumSystemLocales (LCID_SUPPORTED) erfassbar sind angezeigt.
	Nicht gesetzt	Es werden alle Sprachen im Array dwSetLocaleIDs angezeigt.
DM_LANGDLG_ONLY_PRIMARY	gesetzt	Es werden nur die Hauptsprachen (PRIMARY_LANGUAGE) angezeigt.
	Nicht gesetzt	Es werden auch Subsprachen angezeigt.
DM_LANGDLG_NO_NOTIFY		Das Beenden des Dialogs mit "OK" führt nicht zur Benachrichtigung der Applikationen

dwSetLocaleIDs

Array mit den IDs der Sprachen, die im Dialog angezeigt werden sollen.

uSetIDArraySize

Größe des Arrays dwSetLocaleIDs.

lpdwGetLocaleID

Zeiger auf ein DWORD, in dem die ausgewählte Sprach-ID abgelegt werden soll.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Dialog mit "OK" verlassen.

FALSE

Fehler oder Dialog mit "ABBRECHEN" verlassen.

Fehlermeldungen

DM_E_CANCEL	Benutzer hat in einem Dialog "Abbrechen" gewählt
-------------	--

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

2.2.10 Funktionen zur Aktualisierung von Variablen (RT Professional)

2.2.10.1 DMBeginStartVarUpdate (RT Professional)

Verwendung

Aktualisierungsanforderung einleiten. Bei dieser Funktion wird eine Transaktions-ID vergeben, welche die Transaktion identifiziert.

Deklaration

```
BOOL DMBeginStartVarUpdate (  
    LPDWORD          pdwTAID  
    LPCMN_ERROR      lpdmError )
```

Parameter

pdwTAID

Zeiger auf eine Variable, die nach erfolgreichem Aufruf der Funktion die vom Datenmanager vergebene Transaktions-ID enthält.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Transaktions-ID vergeben.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Beispiele

OnTestVariablenBeginstartvarupdate (Seite 497) "TESTCDoc.cpp"

Siehe auch

OnTestVariablenBeginstartvarupdate (Seite 497)

DMEndStartVarUpdate (Seite 453)

DMResumeVarUpdate (Seite 454)

DMStartVarUpdate (Seite 456)

DMStartVarUpdateEx (Seite 460)

DMStopVarUpdate (Seite 472)

DMSuspendVarUpdate (Seite 473)

2.2.10.2 DMEndStartVarUpdate (RT Professional)

Verwendung

Nach dem erfolgreichen Aufruf der Funktion übernimmt der Datenmanager die Aktualisierung der Variablen und leitet die neuen Werte gegebenenfalls an Ihre Applikation weiter.

Deklaration

```
BOOL DMEndStartVarUpdate (  
    DWORD          dwTAID,  
    LPCMN_ERROR    lpdmError );
```

Parameter

dwTAID

dwTAID enthält die beim Aufruf der Funktion DMBeginStartVarUpdate vergebene Transaktions-ID.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktualisierung gestartet.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_TAID	Ungültige Transaktions ID

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMBeginStartVarUpdate (Seite 452)	Aktualisierungsanforderung einleiten
DMStartVarUpdate (Seite 456)	Aktualisierungsvariablen festlegen

Beispiele

OnTestVariablenEndstartvarupdate (Seite 498) "TESTCDoc.cpp"

Siehe auch

OnTestVariablenEndstartvarupdate (Seite 498)

DMBeginStartVarUpdate (Seite 452)

DMStartVarUpdate (Seite 456)

2.2.10.3 DMResumeVarUpdate (RT Professional)

Verwendung

Startet die Aktualisierung aller durch die Transaktion definierten Variablen, wenn durch den Aufruf der Referenzzähler der Transaktion den Wert 0 erreicht.

Die Funktion bildet das Gegenstück zu DMSuspendVarUpdate und sollte mit dieser immer paarweise aufgerufen werden.

Deklaration

```
BOOL DMResumeVarUpdate (  
    DWORD          dwTAID,  
    LPCMN_ERROR    lpdmError);
```

Parameter

dwTAID

dwTAID enthält die beim Aufruf der Funktion `DMBeginStartVarUpdate` vergebene Transaktions-ID.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktualisierung fortsetzen.

FALSE

Fehler.

Bemerkung

Transaktionen verfügen über einen Referenzzähler, der bei jedem Aufruf der Funktion `DMSuspendVarUpdate` inkrementiert wird. Wenn die Aktualisierung der Variablen fortgesetzt werden soll, so muss `DMResumeVarUpdate` so oft aufgerufen werden, bis der Referenzzähler wieder den Stand 0 erreicht.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_TAID	Ungültige Transaktions ID

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMBeginStartVarUpdate (Seite 452)	Aktualisierungsanforderung einleiten
DMSuspendVarUpdate (Seite 473)	Aktualisierung von Variablen aussetzen

Beispiele

OnTestVariablenResumevarupdate (Seite 509) "TESTCDoc.cpp"

Siehe auch

- DMBeginStartVarUpdate (Seite 452)
- DMSuspendVarUpdate (Seite 473)
- OnTestVariablenResumevarupdate (Seite 509)

2.2.10.4 DMStartVarUpdate (RT Professional)

Verwendung

Die Funktion dient der Festlegung der vom Datenmanager zu aktualisierenden Variablen. In der Callback-Funktion übergibt der Datenmanager im Aktualisierungsfall die neuen Variablenwerte an Ihre Applikation.

DMStartVarUpdateEx ist eine weiter entwickelte Funktion, bei der in der Struktur DM_VAR_UPDATE_STRUCTEX in der Callback DM_NOTIFY_VARIABLEEX_PROC zusätzlich der QualityCode mitgeliefert wird.

Deklaration

```
BOOL DMStartVarUpdate (  
    DWORD dwTAID,  
    LPDM_VARKEY lpdmVarKey,  
    DWORD dwItems,  
    DWORD dwCycle,  
    DM_NOTIFY_VARIABLE_PROC lpfnVariable,  
    LPVOID lpvUser,  
    LPCMN_ERROR lpdmError);
```

Parameter

dwTAID

dwTAID enthält die beim Aufruf der Funktion DMBeginStartVarUpdate vergebene Transaktions-ID.

lpdmVarKey

Zeiger auf die erste der Strukturen vom Typ DM_VARKEY (Seite 241) , über welche die zu aktualisierenden Variablen spezifiziert werden.

dwItems

Anzahl der zu aktualisierenden Variablen in den übergebenen Strukturen DM_VARKEY.

dwCycle

Der in dwCycle festgelegte Aktualisierungszyklus gilt für alle über IpdmVarKey festgelegten Variablen. Der Aktualisierungszyklus wird über den Index der Einträge in der Liste der Aktualisierungszyklen definiert.

"Bei Änderung"	azyklisch	Index: 0
"250 ms"	250	Index: 1
"500 ms"	500	Index: 2
"1 s"	1000	Index: 3
"2 s"	2000	Index: 4
"5 s"	5000	Index: 5
"10 s"	10000	Index: 6
"1 min"	60000	Index: 7
"5 min"	300000	Index: 8
"10 min"	600000	Index: 9
"1 h"	3600000	Index: 10
"Anwenderzyklus 1"	(z.B. 2000)	Index: 11
"Anwenderzyklus 2"	(z.B. 3000)	Index: 12
"Anwenderzyklus 3"	(z.B. 4000)	Index: 13
"Anwenderzyklus 4"	(z.B. 5000)	Index: 14
"Anwenderzyklus 5"	(z.B. 10000)	Index: 15

IpfmVariable

Zeiger auf die Callback-Funktion DM_NOTIFY_VARIABLE_PROC zur Datenübergabe durch den Datenmanager.

Wenn IpfmVariable == NULL ist, wird die Aktualisierung im Prozessabbild des Datenmanagers mindestens im geforderten Zyklus durchgeführt, soweit dies physikalisch durch momentane Systemlast, Rechnerperformance, Koppelmedium etc. möglich ist. Die Ermittlung des jeweiligen Werts muss jedoch durch Aufruf der Funktion DMGetValue durch die Applikation selbstständig durchgeführt werden.

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kommt es vor, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

IpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktualisierungsvariablen festgelegt.

FALSE

Fehler.

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMBeginStartVarUpdate (Seite 452)	Aktualisierungsanforderung einleiten
DM_NOTIFY_VARIABLE_PROC (Seite 458)	Aktualisierungsvariablen festlegen (Callback)

Siehe auch

DMEndStartVarUpdate (Seite 453)

DM_VARKEY (Seite 241)

DMBeginStartVarUpdate (Seite 452)

DM_NOTIFY_VARIABLE_PROC (Seite 458)

2.2.10.5 DM_NOTIFY_VARIABLE_PROC (RT Professional)

Beschreibung

Um die vom System ermittelten Daten auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_NOTIFY_VARIABLE_PROC bereitstellen.

Deklaration

```
BOOL ( * DM_NOTIFY_VARIABLE_PROC) (
    DWORD                dwTAID,
    LPDM_VAR_UPDATE_STRUCT lpdmvus,
    DWORD                dwItems,
    LPVOID                lpvUser);
```

Parameter

dwTAID

Transaktions-ID, die vom Datenmanager an die aufrufende Funktion vergeben wurde.

lpdmvus

Zeiger auf die erste der Strukturen vom Typ DM_VAR_UPDATE_STRUCT (Seite 228) , die die Werte der angeforderten Variablen enthalten.

dwItems

Anzahl der in lpdmvus übergebenen Strukturen (entspricht der Anzahl der gelieferten Variablenwerte).

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

Die Rückgabewerte sind abhängig von Ihrer Implementierung

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus der selben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMGetValueWait (Seite 353)	Variablenwerte mit Aktualisierung ermitteln
DMStartVarUpdate (Seite 456)	Aktualisierungsvariablen festlegen

Siehe auch

- DMStartVarUpdate (Seite 456)
- DM_VAR_UPDATE_STRUCT (Seite 228)
- DMGetValueWait (Seite 353)

2.2.10.6 DMStartVarUpdateEx (RT Professional)

Verwendung

Die Funktion dient der Festlegung der vom Datenmanager zu aktualisierenden Variablen. In der Callback-Funktion übergibt der Datenmanager im Aktualisierungsfall die neuen Variablenwerte an die Applikation.

Im Gegensatz zu DMStartVarUpdate wird der Quality Code in der Struktur DM_VAR_UPDATE_STRUCTEX mitgeliefert, die in der Callback DM_NOTIFY_VARIABLEEX_PROC zurückgegeben wird.

Deklaration

```
BOOL DMStartVarUpdateEx (  
    DWORD dwTAID,  
    LPDM_VARKEY lpdmVarKey,  
    DWORD dwItems,  
    DWORD dwCycle,  
    DM_NOTIFY_VARIABLEEX_PROC lpfnVariable,  
    LPVOID lpvUser,  
    LPCMN_ERROR lpdmError);
```

Parameter

dwTAID

dwTAID enthält die beim Aufruf der Funktion DMBeginStartVarUpdate vergebene Transaktions-ID.

lpdmVarKey

Zeiger auf die erste der Strukturen vom Typ DM_VARKEY (Seite 241) , über welche die zu aktualisierenden Variablen spezifiziert werden.

dwItems

Anzahl der zu aktualisierenden Variablen in den übergebenen Strukturen DM_VARKEY.

dwCycle

Der hier festgelegte Aktualisierungszyklus gilt für alle über `IpdmVarKey` festgelegten Variablen. Der Aktualisierungszyklus wird über den Index der Einträge in der Liste der Aktualisierungszyklen definiert.

"Bei Änderung"	Azyklisch	Index: 0
"250 ms"	250	Index: 1
"500 ms"	500	Index: 2
"1 s"	1000	Index: 3
"2 s"	2000	Index: 4
"5 s"	5000	Index: 5
"10 s"	10000	Index: 6
"1 min"	60000	Index: 7
"5 min"	300000	Index: 8
"10 min"	600000	Index: 9
"1 h"	3600000	Index: 10
"Anwenderzyklus 1"	(z.B. 2000)	Index: 11
"Anwenderzyklus 2"	(z.B. 3000)	Index: 12
"Anwenderzyklus 3"	(z.B. 4000)	Index: 13
"Anwenderzyklus 4"	(z.B. 5000)	Index: 14
"Anwenderzyklus 5"	(z.B. 10000)	Index: 15

IpfnVariable

Zeiger auf Callback-Funktion `DM_NOTIFY_VARIABLEEX_PROC` zur Datenübergabe durch den Datenmanager.

Wenn `IpfnVariable == NULL` ist, wird die Aktualisierung im Prozessabbild des Datenmanagers mindestens im geforderten Zyklus durchgeführt, soweit dies physikalisch durch momentane Systemlast, Rechnerperformance, Koppelmedium etc. möglich ist. Die Ermittlung des jeweiligen Werts muss jedoch durch Aufruf der Funktion `DMGetValue` durch die Applikation selbstständig durchgeführt werden.

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

IpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktualisierungsvariablen festgelegt.

FALSE

Fehler.

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMBeginStartVarUpdate (Seite 452)	Aktualisierungsanforderung einleiten
DM_NOTIFY_VARIABLEEX_PROC (Seite 469)	Aktualisierungsvariablen festlegen (Callback)

Siehe auch

DM_VARKEY (Seite 241)

DMBeginStartVarUpdate (Seite 452)

DM_NOTIFY_VARIABLEEX_PROC (Seite 469)

2.2.10.7 DMStartVarUpdateExStr (RT Professional)

Deklaration

```
BOOL DMStartVarUpdateExStr (
    DWORD dwTAID,
    DWORD dwFlags,
    LPVARIANT lpvdmVarKey,
    LPVARIANT lpvCookie,
    DWORD dwCycle,
    DM_NOTIFY_VARIABLEEX_PROC_EXSTR lpfnVariable,
    LPVOID lpvUser,
    LPCMN_ERROR lpdmError);
```

Beschreibung

Die Funktion dient der Festlegung der vom Datenmanager zu aktualisierenden Variablen. In der Callback-Funktion übergibt der Datenmanager im Aktualisierungsfall die neuen Variablenwerte an die Applikation.

Parameter

dwTAID

Enthält die beim Aufruf der Funktion "DMBeginStartVarUpdate" vergebene Transaktions-ID.

dwFlags

Wenn eine Rückgabe des Variablennamen im VARIANT der Struktur "DM_VAR_UPDATE_STRUCT_EXSTR" der "DM_NOTIFY_VARIABLE_PROC_EXSTR" im Format "VT_LPSTR" benötigt wird, kann hier das Flag "DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR" angegeben werden.

lpvdmVarKey

Pointer auf VARIANT mit "VT_ARRAY | VT_VARIANT" zur Übergabe der Variablenliste, die die zu ändernden Variablen kennzeichnen.

Eine VariablenID wird mit dem Typ "VT_I4" und der Variablenname mit "VT_BSTR" in das jeweilige Listenelement eingetragen. Zusätzlich ist noch "VT_LPSTR" (PROPVARIANT) für die Übergabe als allozierter ASCII-String möglich, um z. B. vom Skript aus konstante Namen zu übergeben. "VT_LPSTR" wird dann intern in den geforderten Typ "VT_BSTR" gewandelt.

Bei Multiclient Projekten ab WinCC Version 5.0 ist hier gegebenenfalls bei jedem Namen ein Serverpräfix mit anzugeben (siehe Projekttypen und Versionen).

lpvCookie

Zeiger auf einen VARIANT für ein zusätzliche Liste benutzerabhängiger Daten zu jeder Variable.

Das ist als Ersatz für "lpvUserData" des vorherigen "DM_VARKEY" gedacht und wird in der Struktur "DM_VAR_UPDATE_STRUCT_EXSTR" für jede Variable mit zurückgegeben.

dwCycle

Der hier festgelegte Aktualisierungszyklus gilt für alle über "lpvdmVarKey" festgelegten Variablen. Der Aktualisierungszyklus wird über den Index der Einträge in der Liste der Aktualisierungszyklen definiert.

"Bei Änderung"	azyklisch	Index: 0
"250 ms"	250	Index: 1
"500 ms"	500	Index: 2
"1 s"	1000	Index: 3
"2 s"	2000	Index: 4
"5 s"	5000	Index: 5
"10 s"	10000	Index: 6
"1 min"	60000	Index: 7
"5 min"	300000	Index: 8
"10 min"	600000	Index: 9
"1 h"	3600000	Index: 10
"Anwenderzyklus 1"	(z.B. 2000)	Index: 11
"Anwenderzyklus 2"	(z.B. 3000)	Index: 12
"Anwenderzyklus 3"	(z.B. 4000)	Index: 13

"Anwenderzyklus 4"	(z.B. 5000)	Index: 14
"Anwenderzyklus 5"	(z.B. 10000)	Index: 15

lpfnVariable

Zeiger auf die Rückruffunktion "DM_NOTIFY_VARIABLEEX_PROC" zur Datenübergabe durch den Datenmanager.

Wenn "lpfnVariable" == "NULL" ist, wird die Aktualisierung im Prozessabbild des Datenmanagers mindestens im geforderten Zyklus durchgeführt, soweit physikalisch durch momentane Systemlast, Rechnerperformance, Koppelmedium etc. möglich. Die Ermittlung des jeweiligen Wertes müssen Sie aber mit einem Aufruf der Funktion "DMGetValue" durch die Applikation selbst durchführen.

Wenn ein Programm eine Notify-Routine anmeldet, muss das Programm regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

lpvUser

Zeiger auf applikationsspezifische Daten. Der Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur "CMN_ERROR". Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktualisierungsvariablen festgelegt

FALSE

Fehler

Benötigte Files

dmclient_exstr.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMBeginStartVarUpdate	Aktualisierungsanforderung einleiten
-----------------------	--------------------------------------

Beispiele

Skript-Beispiel "Button DMStartVarUpdateExStr":

```
#include "apdefap.h"
```

```

void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
#pragma code ("OleAut32.dll")
//#include "OleAuto.h"
SAFEARRAY * SafeArrayCreateVector(VARTYPE vt, long lLbound, unsigned
int cElements );
HRESULT SafeArrayPtrOfIndex(SAFEARRAY FAR* psa, long FAR* rgIndices,
void HUGE* FAR* ppvData );
HRESULT SafeArrayLock(SAFEARRAY FAR* psa);
HRESULT SafeArrayUnlock(SAFEARRAY FAR* psa);
#pragma code()

extern BOOL DM_NOTIFY_VARIABLE_PROC_EXSTR
DM_NotifyVariableProcExStr_VarUpdate(
    DWORD dwTAID,
    LPDMS_VAR_UPDATE_STRUCTURE_EXSTR lpdmvus,
    DWORD dwItems,
    LPVOID lpvUser);

VARIANT vVarKey;
VARIANT* pvElem;
SAFEARRAY* parrayKeys;
HRESULT hr;
CMN_ERROR err;
BOOL bRet;
long lInx;
DWORD dwTAID;
DWORD dwFlags;
DWORD dwCycle;

memset(&err, 0, sizeof(err));
bRet = FALSE;
parrayKeys = NULL;
lInx = 0L;
dwFlags = DM_FLAG_RETURN_PROPVARIANT_VT_LPSTR;
//dwCycle = 3; /* 1sec*/
dwCycle = 0; /* on change*/

printf("\r\n\r\n##### enter Test with DMStartVarUpdateExStr
#####");

dwTAID = GetTagDWord("dwUpdTAID");

if (dwTAID)
{
    printf("\r\nStartVarUpdate always running. Stop it before start
again!");
    printf("\r\n##### exit Test with DMStartVarUpdateExStr
(nothing done) #####\r\n");
}

```

```

        return;
    }

    VariantInit (&vVarKey);
    parrayKeys = SafeArrayCreateVector(VT_VARIANT, 0L, 4);
    vVarKey.vt = VT_ARRAY | VT_VARIANT;
    vVarKey.u.parray = parrayKeys;

    SafeArrayLock(parrayKeys);
    lInx = 0L;
    hr = SafeArrayPtrOfIndex(parrayKeys, &lInx, &pvElem);
    pvElem->vt = VT_LPSTR;
    pvElem->u.pbVal = "dwVal_1";
    lInx = 1L;
    hr = SafeArrayPtrOfIndex(parrayKeys, &lInx, &pvElem);
    pvElem->vt = VT_LPSTR;
    pvElem->u.pbVal = "dwVal_2";
    lInx = 2L;
    hr = SafeArrayPtrOfIndex(parrayKeys, &lInx, &pvElem);
    pvElem->vt = VT_LPSTR;
    pvElem->u.pbVal = "dwVal_3";
    lInx = 3L;
    hr = SafeArrayPtrOfIndex(parrayKeys, &lInx, &pvElem);
    pvElem->vt = VT_LPSTR;
    pvElem->u.pbVal = "dwVal_4";
    SafeArrayUnlock(parrayKeys);

    bRet = DMBeginStartVarUpdate(&dwTAID, &err);
    if (FALSE == bRet)
    {
        printf("\r\n  error DMBeginStartVarUpdate, err=%ld,%ld,%ld,%ld,
%ld,[%s]",
            err.dwError1, err.dwError2, err.dwError3,
            err.dwError4, err.dwError5, err.szErrorText);
    }
    else
    {
        printf("\r\n  DMBeginStartVarUpdate (dwTAID=%ld) OK.", dwTAID);
    }
    if (dwTAID)
    {
        SetTagDWord("dwUpdTAID", dwTAID);

        memset(&err, 0, sizeof(err));
        bRet = DMStartVarUpdateExStr(dwTAID, dwFlags, &vVarKey, dwCycle,
            DM_NotifyVariableProcExStr_VarUpdate,
            NULL, &err);
        if (FALSE == bRet)
        {
            printf("\r\n  error DMStartVarUpdateExStr (dwTAID=%ld): err=
%ld,%ld,%ld,%ld,%ld,[%s]",
                dwTAID, err.dwError1, err.dwError2, err.dwError3,

```

```

        err.dwError4, err.dwError5, err.szErrorText);
    }
    else
    {
        printf("\r\n DMStartVarUpdateExStr (dwTAID=%ld) OK.",
dwTAID);
    }

    memset(&err, 0, sizeof(err));
    bRet = DMEndStartVarUpdate(dwTAID, &err);
    if (FALSE == bRet)
    {
        printf("\r\n error DMEndStartVarUpdate (dwTAID=%ld): err=
%ld,%ld,%ld,%ld,%ld, [%s]",
            dwTAID, err.dwError1, err.dwError2, err.dwError3,
            err.dwError4, err.dwError5, err.szErrorText);
    }
    else
    {
        printf("\r\n DMEndStartVarUpdate (dwTAID=%ld) OK.", dwTAID);
    }
}
else
{
    printf("\r\n dwTAID == 0L ???");
}

printf("\r\n##### exit Test with DMStartVarUpdateExStr (dwTAID=
%ld) #####\r\n", dwTAID);

}

```

Skript-Beispiel "Project-Function DM_NotifyVariableProcExStr_VarUpdate":

```

#pragma code ("OleAut32.dll")
// #include "OleAuto.h"
HRESULT VariantChangeType( VARIANTARG FAR* pvargDest, VARIANTARG
FAR* pvarSrc, unsigned short wFlags, VARTYPE vt);
#pragma code()

BOOL DM_NotifyVariableProcExStr_VarUpdate(
    DWORD dwTAID,
    LPDM_VAR_UPDATE_STRUCTURE_EXSTRA lpdmvus,
    DWORD dwItems,
    LPVOID lpvUser)
{
    BOOL bRet = FALSE;
    int i = 0;
    HRESULT hr = 0L;
}

```

```

printf("\r\n*** DM_NotifyVariableProcExStr_VarUpdate entry (dwTAID=
%d) ***", dwTAID);

for (i = 0; i < dwItems; i++)
{
    hr = VariantChangeType((VARIANTARG*)&(lpdmvus[i].vdmValue),
                          (VARIANTARG*)&(lpdmvus[i].vdmValue), 0,
VT_R8);
    if (VT_LPSTR == lpdmvus[i].vdmVarKey.vt)
    {
        printf("\r\n    [%d]:{VarKey=[%s]", i,
(LPCSTR)lpdmvus[i].vdmVarKey.u.pbVal);
    }
    else if (VT_I4 == lpdmvus[i].vdmVarKey.vt)
    {
        printf("\r\n    [%d]:{VarKey=[%ld]", i,
lpdmvus[i].vdmVarKey.u.lVal);
    }
    else
    {
        printf("\r\n    [%d]:{unexpected VarKey.vt=[%d]",
lpdmvus[i].vdmVarKey.vt);
    }
    printf(", dmValue=[%g]", lpdmvus[i].vdmValue.u.dblVal);
    printf(", dwState=[%ld]", lpdmvus[i].dwState);
    printf(", dwType=[%ld]", lpdmvus[i].dmTypeRef.dwType);
    printf(", dwSize=[%ld]", lpdmvus[i].dmTypeRef.dwSize);
    if (lpdmvus[i].dmTypeRef.lpszTypeName)
    {
        printf(", lpszTypeName=[%s]",
lpdmvus[i].dmTypeRef.lpszTypeName);
        printf(", dwNameCharCount=[%ld]",
lpdmvus[i].dmTypeRef.dwNameCharCount);
    }
    else
    {
        printf(", lpszTypeName=[NULL]");
    }
    printf("}");
}

printf("\r\n*** DM_NotifyVariableProcExStr_VarUpdate exit (dwTAID=
%d)***", dwTAID);
return bRet;
}

```

Skript-Beispiel "Button DMStopVarUpdate":

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{

```

```

DWORD dwTAID;
CMN_ERROR err;
BOOL bRet;

dwTAID = GetTagDWord("dwUpdTAID");
memset(&err, 0, sizeof(err));
bRet = FALSE;

if (dwTAID)
{
    bRet = DMStopVarUpdate(dwTAID, &err);
    if (FALSE == bRet)
    {
        printf("\r\n!!!  error DMStopVarUpdate (dwTAID=%ld): err=%ld,
%ld,%ld,%ld,%ld,%ld, [%s]  !!!\r\n",
            dwTAID, err.dwError1, err.dwError2, err.dwError3,
            err.dwError4, err.dwError5, err.szErrorText);
        memset(&err, 0, sizeof(err));
        bRet = DMStopAllUpdates(&err);
        if (FALSE == bRet)
        {
            printf("\r\n!!!  error DMStopAllUpdates (dwTAID=%ld):
err=%ld,%ld,%ld,%ld,%ld,%ld, [%s]  !!!\r\n",
                dwTAID, err.dwError1, err.dwError2, err.dwError3,
                err.dwError4, err.dwError5, err.szErrorText);
        }
        else
        {
            printf("\r\n  DMStopAllUpdates OK.");
        }
    }
    else
    {
        printf("\r\n  DMStopVarUpdate (dwTAID=%ld) OK.", dwTAID);
    }
}

dwTAID = 0L;

SetTagDWord("dwUpdTAID", dwTAID);
}

```

2.2.10.8 DM_NOTIFY_VARIABLEEX_PROC (RT Professional)

Beschreibung

Um die vom System ermittelten Daten auswerten zu können, müssen Sie eine Callback-Funktion vom Typ DM_NOTIFY_VARIABLEEX_PROC bereitstellen.

Deklaration

```
BOOL ( * DM_NOTIFY_VARIABLEEX_PROC) (  
    DWORD                dwTAID,  
    LPDM_VAR_UPDATE_STRUCTEX lpdmvus,  
    DWORD                dwItems,  
    LPVOID               lpvUser);
```

Parameter

dwTAID

Transaktions-ID, die vom Datenmanager an die aufrufende Funktion vergeben wurde.

lpdmvus

Zeiger auf die erste der Strukturen vom Typ DM_VAR_UPDATE_STRUCTEX (Seite 230), die die Werte der angeforderten Variablen enthalten.

dwItems

Anzahl der in lpdmvus übergebenen Strukturen (entspricht der Anzahl der gelieferten Variablenwerte).

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

Die Rückgabewerte sind abhängig von Ihrer Implementierung

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Benötigte Dateien

dmclient.h

Verwandte Funktionen

DMGetValueWaitEx (Seite 356)	Variablenwerte mit Aktualisierung ermitteln
DMStartVarUpdateEx (Seite 460)	Aktualisierungsvariablen festlegen

Siehe auch

DMStartVarUpdateEx (Seite 460)

DM_VAR_UPDATE_STRUCTEX (Seite 230)

DMGetValueWaitEx (Seite 356)

2.2.10.9 DMStopAllUpdates (RT Professional)**Verwendung**

Beendet alle von der Applikation angeforderten Variablenaktualisierungen. Vorzugsweise ist diese Funktion beim Beenden des entsprechenden Moduls aufzurufen.

Deklaration

```
BOOL DMStopAllUpdates (
    LPCMN_ERROR    lpdmError);
```

Parameter**lpdmError**

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Aktualisieren von Variablen beendet.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
--------------------	-----------------------------------

Benötigte Dateien

dmclient.h
dmclient.lib
dmclient.dll

Beispiele

OnTestVariablenStopallupdates (Seite 512) "TESTCDoc.cpp"

Siehe auch

OnTestVariablenStopallupdates (Seite 512)

2.2.10.10 DMStopVarUpdate (RT Professional)

Verwendung

Beendet die Aktualisierung von Variablen einer bestimmten Transaktion.

Deklaration

```
BOOL DMStopVarUpdate (  
    DWORD          dwTAID,  
    LPCMN_ERROR    lpdmError);
```

Parameter

dwTAID

dwTAID enthält die beim Aufruf der Funktion DMBeginStartVarUpdate vergebene Transaktions-ID.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Aktualisieren der Variablen beendet.

FALSE

Fehler.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_TAID	Ungültige Transaktions ID

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMBeginStartVarUpdate (Seite 452)	Aktualisierungsanforderung einleiten
-----------------------------------	--------------------------------------

Beispiele

OnTestVariablenStopvarupdate (Seite 512) "TESTCDoc.cpp"

Siehe auch

OnTestVariablenStopvarupdate (Seite 512)

DMBeginStartVarUpdate (Seite 452)

2.2.10.11 DMSuspendVarUpdate (RT Professional)**Verwendung**

Unterbricht die Aktualisierung aller durch die Transaktion definierten Variablen.

Innerhalb des Prozessabbildes des Datenmanagers werden die Variablen weiterhin aktualisiert. Die aktuellen Werte werden jedoch nicht mehr an die Applikation übertragen.

Deklaration

```
BOOL DMSuspendVarUpdate (  
    DWORD          dwTAID,  
    LPCMN_ERROR    lpdmError);
```

Parameter

dwTAID

dwTAID enthält die beim Aufruf der Funktion DMBeginStartVarUpdate vergebene Transaktions-ID.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktualisieren ausgesetzt.

FALSE

Fehler.

Bemerkung

Transaktionen verfügen über einen Referenzzähler, so dass die Funktion DMSuspendVarUpdate für eine Transaktion mehrmals aufgerufen werden kann. Wenn die Transaktion bereits im Zustand "SUSPENDED" ist, wird lediglich der Referenzzähler inkrementiert.

Zum weiteren Aktualisieren muss DMResumeVarUpdate so oft aufgerufen werden, bis der Referenzzähler wieder den Stand 0 erreicht.

Fehlermeldungen

DM_E_NOT_CONNECTED	Keine Verbindung zum Datenmanager
DM_E_INVALID_TAID	Ungültige Transaktions ID

Benötigte Dateien

dmclient.h

dmclient.lib

dmclient.dll

Verwandte Funktionen

DMBeginStartVarUpdate (Seite 452)	Aktualisierungsanforderung einleiten
DMResumeVarUpdate (Seite 454)	Aktualisierung von Variablen fortsetzen

Beispiele

OnTestVariablenSuspendvarupdate (Seite 513) "TESTCDoc.cpp"

Siehe auch

DMResumeVarUpdate (Seite 454)

DMBeginStartVarUpdate (Seite 452)

OnTestVariablenSuspendvarupdate (Seite 513)

2.2.11 Beispiele (RT Professional)

2.2.11.1 Connection to DM (RT Professional)

Beispiel

```
// =====
// =====
// Desc. : Modul with examples for Data-Manager
// *****
#include "stdafx.h" // if MFC classes
// #include "odkapi.h" // if console application
#include <time.h>
TCHAR g_szProjectFile[255] = {0};
TCHAR g_szDSNName[255] = {0};
#include "DM01.h"

//{{ODK_EXAMPLE}Connection to DM (MCP)}
//{{FUNCTION}DMGetConnectionState (MCP)}
//{{FUNCTION}DMConnect (MCP)}
//{{FUNCTION}DM_NOTIFY_PROC (MCP)}
//{{FUNCTION}DMDisconnect (MCP)}
//{{FUNCTION}(END)}

// =====
// Function: MyDMConnect(void) ODK DM CS
// =====
BOOL MyDMConnect(void)
{
    CMN_ERROR Error;
    BOOL ret = FALSE;
    TCHAR szText[255];
    TCHAR szAppName[255];
    VOID* pvUser = AfxGetApp();
    _tcsncpy_s(szAppName, _countof(szAppName), _T("MyODKApp_23"), _TRUNCATE);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = DMGetConnectionState(&Error);
    if(FALSE == ret) // not connected
    {
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = DMConnect(szAppName, MyDMNotifyCallback, pvUser, &Error);
        if(FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in DMConnect: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMConnect"));
        }
        ODKTrace(szText);
    }
}
```

```

else // already connected
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMGetConnectionState: OK"));
}
else // already connected
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMGetConnectionState: OK"));
    ODKTrace(szText);
}
return(ret);
}

// IMPLEMENTATION
// =====
// Function: MyDMGetConnectionState(void) ODK DM CS
// =====
BOOL MyDMGetConnectionState(void)
{
    CMN_ERROR Error;
    BOOL ret = FALSE;
    TCHAR szText[255];
    TCHAR szAppName[255];
    VOID* pvUser = AfxGetApp();
    _tcsncpy_s(szAppName, _countof(szAppName), _T("MyODKApp_23"), _TRUNCATE);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = DMGetConnectionState(&Error);
    if(FALSE == ret) // not connected
    {
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = DMConnect(szAppName, MyDMNotifyCallback, pvUser, &Error);
        if(FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in DMConnect: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMConnect"));
        }
        ODKTrace(szText);
        //printf("%s\r\n", szText);
    }
    else // already connected
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMGetConnectionState: OK"));
        ODKTrace(szText);
    }
    return(ret);
}

// =====
// Function: MyDMDisconnect(void) ODK DM CS
// =====
BOOL MyDMDisconnect(void)
{

```

```

    CMN_ERROR Error;
    BOOL ret = FALSE;
    TCHAR szText[255];
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = DMDisconnect(&Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in DMConnect: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMDisconnect"));
    }
    ODKTrace(szText);
    return(ret);
}

// =====
// Function: MyDMNotifyCallback
// =====
BOOL MyDMNotifyCallback(DWORD dwNotifyClass, DWORD dwNotifyCode, LPBYTE lpbyData,
    DWORD dwItems, LPVOID lpvUser)
{
    lpvUser;
    lpbyData;
    TCHAR szText[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("***DMNotifyCallback**"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  NotifyClass = 0x
%08X"), dwNotifyClass);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  NotifyCode = 0x%08X"),
dwNotifyCode);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  Items = %d"), dwItems);
    ODKTrace(szText);
    //printf("%s\r\n", szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("*****"));
    ODKTrace(szText);
    return(TRUE);
}

//{{ODK_EXAMPLE} (END) }

```

Siehe auch

DMConnect (Seite 278)

DMDisconnect (Seite 285)

DMGetConnectionState (Seite 297)

2.2.11.2 Enum Data of Tags (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enum Data of Tags (MCP)}
//{{FUNCTION}DMEnumVarData4 (MCP)}
//{{FUNCTION}DM_ENUM_VARIABLE_PROC4 (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: MyDMEnumVarData4(void) ODK DM CS
// =====
// Desc. :
//-----
BOOL MyDMEnumVariable4Callback(LPDM_VARKEY lpdmVarKey, LPDM_VARIABLE_DATA4 lpdmVarData,
LPVOID lpvUser)
{
    lpvUser;
    TCHAR szText[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMEnumVariableCallback"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...szName=%s dwVarProperty=0x
%04X"), lpdmVarKey->szName, lpdmVarData->dwVarProperty);
    ODKTrace(szText);
    return TRUE;
}

void MyDMEnumVarData4()
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    DM_VARKEY dmVarKey[2];
    TCHAR szText[255];
    CMN_ERROR Error;
    VOID* pUser = NULL;
    memset(&Error, 0, sizeof(CMN_ERROR));
    memset(&dmVarKey, 0, 2 * sizeof(DM_VARKEY));
    dmVarKey[0].dwKeyType = DM_VARKEY_NAME;
    _tcsncpy_s(dmVarKey[0].szName, _countof(dmVarKey[0].szName), _T("INT_TEST_VAR"),
    _TRUNCATE);
    dmVarKey[1].dwKeyType = DM_VARKEY_NAME;
    _tcsncpy_s(dmVarKey[1].szName, _countof(dmVarKey[1].szName), _T("EXT_TEST_VAR"),
    _TRUNCATE);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    ret = DMEnumVarData4 (/*PROJ_PATH*/g_szProjectFile,
        &dmVarKey[0],
        2,
        MyDMEnumVariable4Callback,
        pUser,
        &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in DMEnumVarData4: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
}

```

```
    }  
}  
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

[DMEnumVarData4 \(Seite 329\)](#)

[DM_ENUM_VARIABLE_PROC4 \(Seite 331\)](#)

2.2.11.3 Enum open projects (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enum open projects (MCP)}
//{{FUNCTION}DMEEnumOpenedProjects (MCP)}
//{{FUNCTION}DM_ENUM_OPENED_PROJECTS_PROC (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: MyDMEEnumOpenedProjects(void) ODK DM CS
// =====
// Desc. : Inquire project informations
// =====
BOOL MyDMEEnumOpenProjectsCallback( LPDM_PROJECT_INFO lpInfo, LPVOID lpvUser )
{
    lpvUser;
    // this callback is only called once by every call of DMEEnumOpenedProjects
    // because, there can only opened one project at the same time
    TCHAR szText[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("***DMEEnumOpenProjects***"));
    ODKTrace(szText);
    // copy the project file name to g_szProjectFile for global use
    // many API-functions need this name to select the project database
    _tcsncpy_s(g_szProjectFile, _countof(g_szProjectFile), lpInfo->szProjectFile,
    _TRUNCATE);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" ProjectFile = %s"),lpInfo-
    >szProjectFile);
    ODKTrace(szText);

    _tcsncpy_s(g_szDSNName, _countof(g_szDSNName), lpInfo->szDSNName, _TRUNCATE);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" DSNName = %s"),lpInfo-
    >szDSNName);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" DataLocale = 0x%08X"),lpInfo-
    >dwDataLocale);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("*****"));
    ODKTrace(szText);
    return( TRUE ); // only 1. element
}

BOOL MyDMEEnumOpenedProjects(void)
{
    CMN_ERROR Error;
    BOOL ret= FALSE;
    DWORD dwItems;
    TCHAR szText[255];
    VOID* pvUser = AfxGetApp();
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = MyDMConnect();
    if(TRUE == ret)
    {
        ret = DMEEnumOpenedProjects(&dwItems, MyDMEEnumOpenProjectsCallback, pvUser, &Error);
        if(FALSE == ret)
    }
}

```

```
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
DMEnumOpenedProjects: E1= 0x%08lx ; E2= 0x%08lx ; %s"), %s",
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMEnumOpenedProjects"));
    }
    ODKTrace(szText);
}
return(ret);
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

[DMEnumOpenedProjects \(Seite 305\)](#)

[DM_ENUM_OPENED_PROJECTS_PROC \(Seite 306\)](#)

2.2.11.4 Enumerate all structured types (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enumerate all structured types (MCP)}
//{{FUNCTION}GAPIEnumTypes (MCP)}
//{{FUNCTION}DM_ENUM_TYPES_PROC (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: MyGAPIEnumTypes(void) ODK DM CS
// =====
// Desc. :
//-----
BOOL MyDMEnumTypeCallback(LPCSTR lpszTypeName,DWORD dwTypeID,
                          DWORD dwCreatorID, LPVOID lpvUser )
{
    lpvUser;
    TCHAR szText[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("***DMEnumTypeCallback**"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TypeName = %s TypeID = 0x%08x
CreatorID = %d"),
                lpszTypeName,dwTypeID,dwCreatorID);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("*****"));
    ODKTrace(szText);
    return TRUE;
}

void MyGAPIEnumTypes()
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    CMN_ERROR Error;
    BOOL ret = FALSE;
    TCHAR szText[255];
    TCHAR szProjectFile[255];
    VOID* pvUser = NULL;
    ret = FALSE;
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = MyDMGetConnectionState(); //check the connection state to DM
    if(FALSE != ret)
    {
        MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
        _tcsncpy_s(szProjectFile, _countof(szProjectFile), /*PROJ_PATH*/g_szProjectFile,
        _TRUNCATE);
        memset(&Error,0,sizeof(CMN_ERROR));
        ret = GAPIEnumTypes(szProjectFile,
                            MyDMEnumTypeCallback,
                            pvUser,
                            &Error);
        if(FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in DMEnumTypes: E1=
0x%08lx ; E2= 0x%08lx ; %s"),

```

```
        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMEnumTypes"));
    }
    ODKTrace(szText);
}
}
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

[GAPIEnumTypes \(Seite 439\)](#)

[DM_ENUM_TYPES_PROC \(Seite 441\)](#)

2.2.11.5 Enumerate all connections (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enumerate all connections (MCP)}
//{{FUNCTION}DMEnumConnectionData (MCP)}
//{{FUNCTION}DM_ENUM_CONNECTION_PROC (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: MyDMEnumConnectionData(void) ODK DM CS
// =====
// Desc. :
//-----
BOOL MyDMEnumConnectionCallback(LPDM_CONNECTION_DATA lpdmConData, LPVOID lpvUser)
{
    lpvUser;
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMEnumConnectionCallback"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...Connection = %s"),lpdmConData->szConnection);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...UnitName = %s"),lpdmConData->szUnitName);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...Common=%s"),lpdmConData->szCommon);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...Specific=%s"),lpdmConData->szSpecific);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...VarNum=%d"),lpdmConData->dwVarNum);
    ODKTrace(szText);
    return TRUE;
}

void MyDMEnumConnectionData()
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    CMN_ERROR Error;
    TCHAR szText[255];
    DWORD dwItems = 0;
    TCHAR szProjectFile[255];
    DM_CONNKEY ConnKey;
    memset(&ConnKey,0,sizeof(ConnKey));
    memset(&Error,0,sizeof(Error));
    ret = MyDMGetConnectionState();
    if(FALSE != ret)
    {
        MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
        _tcsncpy_s(szProjectFile, _countof(szProjectFile), /*PROJ_PATH*/g_szProjectFile,
        _TRUNCATE);
        ret = DMEnumConnectionData(szProjectFile,

```

```
        &ConnKey,  
        dwItems,  
        MyDMenuConnectionCallback,  
        NULL,  
        &Error);  
if(FALSE == ret)  
{  
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in  
DMenuConnectionData: E1= 0x%08lx ; E2= 0x%08lx ; %s"),  
        Error.dwError1, Error.dwError2, Error.szErrorText);  
}  
else  
{  
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DMenuConnectionData"));  
}  
ODKTrace(szText);  
}  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DMenuConnectionData (Seite 443)

DM_ENUM_CONNECTION_PROC (Seite 446)

2.2.11.6 Inquire project informations (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Inquire project informations (MCP)}
//{{FUNCTION}DMGetProjectInformation (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: DMGetProjectInformation(void) ODK DM CS
// =====
// Desc. : Inquire project informations
// =====
void MyDMGetProjectInformation(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    CMN_ERROR Error;
    BOOL ret = FALSE;
    TCHAR szText[255];
    TCHAR szProjectFile[_MAX_PATH + 1];
    VOID* pvUser = AfxGetApp();
    DM_PROJECT_INFO Info;
    memset(&Error, 0, sizeof(CMN_ERROR));
    memset(&Info, 0, sizeof(DM_PROJECT_INFO));
    ret = MyDMGetConnectionState(); //check the connection state
    if(FALSE != ret)
    {
        MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
        _tcsncpy_s(szProjectFile, _countof(szProjectFile), /*PROJ_PATH*/g_szProjectFile,
        _TRUNCATE);
        ret = DMGetProjectInformation(szProjectFile, &Info, &Error);
        if(FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
DMGetProjectInformation: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
            ODKTrace(szText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("***DMGetProjectInformation:
%d**"), ret);
            ODKTrace(szText);
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" ProjectFile=
%s"), Info.szProjectFile);
            ODKTrace(szText);
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" DSNName=
%s"), Info.szDSNName);
            ODKTrace(szText);
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" DataLocale=0x
%08X"), Info.dwDataLocale);
            ODKTrace(szText);
            _sntprintf_s(szText, _countof(szText), _TRUNCATE,
            _T("*****"), ret);
            ODKTrace(szText);
        }
    }
}

```

```
    }  
  }  
}  
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

DMGetProjectInformation (Seite 309)

2.2.11.7 OnTestDeactivateRuntimeProject (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestDeactivateRuntimeProject (MCP)}  
//{{FUNCTION}DMDeactivateRTProject (MCP)}  
//{{FUNCTION} (END) }  
void CTestCliDoc::OnTestDeactivateRuntimeProject()  
{  
    CCmnError cmnError;  
    if(!::DMDeactivateRTProject(cmnError))  
    {  
        cmnError.Show(__T("DMDeactivateRTProject failed\n"));  
    }  
}  
//{{ODK_EXAMPLE} (END) }
```

2.2.11.8 OnTestEnumGroupsAll (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestEnumGroupsAll (MCP)}  
//{{FUNCTION}DMEnumVarGrpData (MCP)}  
//{{FUNCTION} (END) }  
void CTestCliDoc::OnTestEnumGroupsAll()  
{  
    CCmnError cmnError;  
    if(!::DMEnumVarGrpData((LPSTR) (LPCTSTR) m_strProject, NULL, 0, EnumVarGrpProc, this,  
cmnError))  
    {  
        cmnError.Show(__T("DMEnumVarGrpData failed\n"));  
    }  
}  
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

DMLenumVarGrpData (Seite 332)

2.2.11.9 OnTestEnumVariables (RT Professional)**Beispiel**

```

//{{ODK_EXAMPLE}OnTestEnumVariables (MCP)}
//{{FUNCTION}DMLenumVariables (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestEnumVariables()
{
    DM_VARFILTER dmVarFilter;
    memset( &dmVarFilter, 0, sizeof(DM_VARFILTER) );
    //-----
    // Initialisierung des Filters //
    //-----
    // //
    dmVarFilter.dwFlags = DM_VARFILTER_TYPE;
    dmVarFilter.dwNumTypes = 3;
    DWORD dwFilterTypes[3];
    dwFilterTypes[0] = DM_VARTYPE_BIT;
    dwFilterTypes[1] = DM_VARTYPE_DWORD;
    dwFilterTypes[2] = DM_VARTYPE_DOUBLE;
    dmVarFilter.pdwTypes = dwFilterTypes;
    dmVarFilter.lpszName = __T("VAR_1_BIT");
    dmVarFilter.lpszGroup = __T("VARGROUP_1");
    dmVarFilter.lpszConn = __T("TF_CONN_1");
    // //
    //-----
    CCmnError Error;
    memset(&Error,0,sizeof(CCmnError));
    if(!DMLenumVariables((LPSTR)(LPCTSTR) m_strProject, NULL/*&dmVarFilter*/,
EnumVariablesProc, this, &Error))
    {
        Error.Show(__T("DMLenumVariables failed\n"));
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMLenumVariables (Seite 337)

2.2.11.10 OnTestEnumConnectionDataAll (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestEnumConnectionDataAll (MCP)}  
//{{FUNCTION}DMEEnumConnectionData (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestEnumConnectionDataAll()  
{  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    if( !::DMEEnumConnectionData( m_strProject, NULL, 0, EnumConnectionDataProc, this,  
&Error))  
    {  
        Error.Show(__T("DMEEnumConnectionData failed\n"));  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DMEEnumConnectionData (Seite 443)

2.2.11.11 OnTestMachines (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}OnTestMachines (MCP)}
//{{FUNCTION}DMGetMachineTable (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestMachines()
{
    CMN_ERROR Error;
    memset(&Error,0,sizeof(CMN_ERROR));
    DM_MACHINE_TABLE dmMachineTable;
    memset(&dmMachineTable, 0, sizeof(dmMachineTable));
    if(!DMGetMachineTable(m_strProject, &dmMachineTable, &Error))
    {
        // Error.Show(__TEXT("DMGetMachineTable failed\n"));
        // AfxMessageBox( strError );
    }
    else
    {
        for( int i = 0; i < dmmachinetable.nNumMachines; i++ )
        {
            CString strData;
            strData.Format( _T( "Rechner : %s, Typ : %s, Standort : %s" ),
                dmMachineTable.tm[i].szMachineName,
                dmMachineTable.tm[i].fServer ? _T( "Server" ) : _T( "Client oder ES" ),
                dmMachineTable.tm[i].fLocal ? _T( "local" ) : _T( "remote" ));
            PutStr( strData);
        }
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMGetMachineTable (Seite 301)

2.2.11.12 OnTestProjectInfo (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}OnTestProjectInfo (MCP)}
//{{FUNCTION}DMGetProjectInformation (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestProjectInfo()
{
    CCmnError cmnError;
    DM_PROJECT_INFO ProjectInfo;
    memset(&ProjectInfo, 0, sizeof(DM_PROJECT_INFO));
}

```

```
        if(!DMGetProjectInformation(m_strProject, &ProjectInfo,
cmnError))
        {
            cmnError.Show(__TEXT("DMGetProjectInformation failed\n"));
        }
        else
        {
            CString strData;
            strData.Format(_T("Projekt : %s, Data Source : %s,
DataLocale : %08X"),
                ProjectInfo.szProjectFile,
                ProjectInfo.szDSNName,
                ProjectInfo.dwDataLocale );
            PutStr(strData);
        }
    }
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

[DMGetProjectInformation \(Seite 309\)](#)

2.2.11.13 OnTestProjectPaths (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestProjectPaths (MCP)}
//{{FUNCTION}DMGetProjectDirectory (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestProjectPaths()
{
    DM_DIRECTORY_INFO dmDirInfo;
    CCmnError Error;
    memset(&dmDirInfo, 0, sizeof(DM_DIRECTORY_INFO));
    memset(&Error, 0, sizeof(CCmnError));
    if(!DMGetProjectDirectory(m_strAppName, m_strProject,
        &dmDirInfo, &Error))
    {
        Error.Show(__TEXT("DMGetProjectDirectory failed\n"));
    }
    else
    {
        CString strData;
        strData.Format(_T("szProjectDir = %s"), dmDirInfo.szProjectDir);
        PutStr(strData);
        strData.Format(_T("szProjectAppDir = %s"), dmDirInfo.szProjectAppDir);
        PutStr(strData);
        strData.Format(_T("szProjectGlobalLibDir = %s"), dmDirInfo.szGlobalLibDir);
        PutStr(strData);
        strData.Format(_T("szProjectLibDir = %s"), dmDirInfo.szProjectLibDir);
        PutStr(strData);
        strData.Format(_T("szLokalProjectAppDir = %s"), dmDirInfo.szLokalProjectAppDir);
        PutStr(strData);
    }
}
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

[DMGetProjectDirectory \(Seite 308\)](#)

2.2.11.14 OnTestOpenProject (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestOpenProject (MCP)}  
//{{FUNCTION}DmOpenProject (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestOpenProject()  
{  
    #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"  
    TCHAR szProject[_MAX_PATH + 1];  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    CTestCliView* pView = GetFirstView();  
    memset(szProject,0, sizeof(szProject)); // Delete projectname to call dialog  
        // Or set fixed projectname  
    strcpy( szProject, _T(PROJ_PATH) );  
    if(!DmOpenProject(pView->GetSafeHwnd(),  
        szProject, NELEM(szProject), &Error))  
    {  
        Error.Show(__TEXT("DmOpenProject failed.\n"));  
    }  
    else  
    {  
        AfxMessageBox(szProject);  
        m_strProject = szProject;  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DmOpenProjectPlus (Seite 313)

2.2.11.15 OnTestOpenProjects (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}OnTestOpenProjects (MCP)}
//{{FUNCTION}DMLenumOpenedProjects (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestOpenProjects()
{
    CCmnError Error;
    memset(&Error,0,sizeof(CCmnError));
    if(!DMLenumOpenedProjects(NULL, OpenProjectsProc, this, &Error))
    {
        Error.Show(__TEXT("DMLenumOpenedProjects failed\n"));
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMLenumOpenedProjects (Seite 305)

2.2.11.16 OnTestRuntimeProject (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}OnTestRuntimeProject (MCP)}
//{{FUNCTION}DMLGetRuntimeProject (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestRuntimeProject()
{
    CCmnError Error;
    memset(&Error,0,sizeof(CCmnError));
    TCHAR szBuffer[_MAX_PATH + 1];
    if(!DMLGetRuntimeProject(szBuffer, NELEM(szBuffer), &Error))
    {
        Error.Show(__TEXT("DMLGetRuntimeProject failed\n"));
    }
    else
    {
        PutStr(szBuffer);
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMLGetRuntimeProject (Seite 311)

2.2.11.17 OnTestSystemLocale (RT Professional)**Beispiel**

```

//{{ODK_EXAMPLE}OnTestSystemLocale (MCP)}
//{{FUNCTION}DMGetSystemLocale (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestSystemLocale()
{
    CCmnError Error;
    memset(&Error,0,sizeof(CCmnError));
    DWORD dwLocaleID = 0;
    if(!DMGetSystemLocale(&dwLocaleID, &Error))
    {
        Error.Show(__TEXT("DMGetSystemLocale failed\n"));
    }
    else
    {
        CString strData;
        strData.Format(_T("Systemlocale : %08X"), dwLocaleID);
        PutStr(strData);
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMGetSystemLocale (Seite 448)

2.2.11.18 OnTestUpdateCycles (RT Professional)**Beispiel**

```

//{{ODK_EXAMPLE}OnTestUpdateCycles (MCP)}
//{{FUNCTION}DMLenumUpdateCycles (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestUpdateCycles()
{
    DWORD dwNumCalls = 0;
    CCmnError Error;
    memset(&Error,0,sizeof(CCmnError));
    if( !DMLenumUpdateCycles(m_strProject, &dwNumCalls,
        EnumCyclesProc, this, &Error))
    {
        Error.Show(__TEXT("DMLenumUpdateCycles failed\n"));
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMLenumUpdateCycles (Seite 289)

2.2.11.19 OnTestVariablenBeginstartvarupdate (RT Professional)**Beispiel**

```
//{{ODK_EXAMPLE}OnTestVariablenBeginstartvarupdate (MCP)}  
//{{FUNCTION}DMBeginStartVarUpdate (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenBeginstartvarupdate()  
{  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    if(!DMBeginStartVarUpdate(&m_dwTAID, &Error))  
    {  
        Error.Show(_T("DMBeginStartVarUpdate failed\n"));  
    }  
    else  
    {  
        CString strData;  
        strData.Format(_T("DMBeginStartVarUpdate: TAID:%lu."), m_dwTAID);  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DMBeginStartVarUpdate (Seite 452)

2.2.11.20 OnTestVariablenEndstartvarupdate (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenEndstartvarupdate (MCP)}  
//{{FUNCTION}DMEndStartVarUpdate (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenEndstartvarupdate()  
{  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    if( !DMEndStartVarUpdate(m_dwTAID, &Error))  
    {  
        Error.Show(_T("DMEndStartVarUpdate failed\n"));  
    }  
    else  
    {  
        CString strData;  
        strData.Format(_T("DMEndStartVarUpdate: TAID:%lu."), m_dwTAID);  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DMEndStartVarUpdate (Seite 453)

2.2.11.21 OnTestVariablenGetvalue (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}OnTestVariablenGetvalue (MCP)}
//{{FUNCTION}DMGetValue (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestVariablenGetvalue()
{
    int nNum = GetVarCount();
    LPDM_VARKEY lpdmVarKey = GetVarKeys();

    LPDM_VAR_UPDATE_STRUCT lpdmvus = new DM_VAR_UPDATE_STRUCT[nNum];
    memset(lpdmvus, 0, sizeof(DM_VAR_UPDATE_STRUCT) * nNum);
    // the included VARIANT's in the DM_VAR_UPDATE_STRUCT's then initialized to VT_EMPTY
with the memset,
    // don't do this later again, because VT_BSTR's can be present after DMGetValue(...)!

    CCmnError Error;
    memset(&Error, 0, sizeof(CCmnError));

    if( !DMGetValue(lpdmVarKey, nNum, lpdmvus, &Error))
    {
        Error.Show(_T("DMGetValue failed\n"));
    }
    else
    {
        CString strData;
        for( int i=0; i < nnum; i++)
        {
            LPDM_VAR_UPDATE_STRUCT lpdmvus2 = &lpdmvus[i];
            switch( lpdmvus2->dmTypeRef.dwType )
            {
                case DM_VARTYPE_BIT:
                    if (lpdmvus2->dmValue.boolVal)
                    {
                        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = TRUE"),
                            lpdmvus2->dmVarKey.szName,
                            lpdmvus2->dmVarKey.dwID);
                    }
                    else
                    {
                        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = FALSE"),
                            lpdmvus2->dmVarKey.szName,
                            lpdmvus2->dmVarKey.dwID);
                    }
                    break;
                case DM_VARTYPE_BYTE:
                    strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %u"),
                        lpdmvus2->dmVarKey.szName,
                        lpdmvus2->dmVarKey.dwID,
                        lpdmvus2->dmValue.bVal);
                    break;
                case DM_VARTYPE_SBYTE:

```

```

        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %d"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.bVal);
        break;
    case DM_VARTYPE_WORD:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %u"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.iVal);
        break;
    case DM_VARTYPE_SWORD:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %d"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.iVal);
        break;
    case DM_VARTYPE_DWORD:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %u"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.lVal);
        break;
    case DM_VARTYPE_SDWORD:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %d"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.lVal);
        break;
    case DM_VARTYPE_FLOAT:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %f"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.fltVal);
        break;
    case DM_VARTYPE_DOUBLE:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %f"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.dblVal);
        break;
    case DM_VARTYPE_TEXT_8:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %s"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.bstrVal);
        break;
/*
    case DM_VARTYPE_TEXT_16:
        strData.Format(_T("GetValue: Variable: %s\t( ID = %d ):\tWert = %s"),
        lpdmvus2->dmVarKey.szName,
        lpdmvus2->dmVarKey.dwID,
        lpdmvus2->dmValue.bstrVal);
        break;
*/
    default:

```

```
        strData.Format(_T("Unbekannter Variablentyp !"));
    }
    PutStr(strData);

    // clear the given VARIANT's in every array element, because a VT_BSTR's can be
present // do not only delete the array later without clearing the Variants here because
of memory leak's
        VariantClear(&(lpdmvus2->dmValue));
    }
}
delete []lpdmVarKey;
delete []lpdmvus;
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

[DMGetValue \(Seite 341\)](#)

2.2.11.22 OnTestVariablenGetvaluawait (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}OnTestVariablenGetvaluawait (MCP)}
//{{FUNCTION}DMGetValueWait (MCP)}
//{{FUNCTION}(END)}
//-----< Notification-Callback für GetValueWait >-----
BOOL GetValueWaitNotify(DWORD dwTAID,
                        LPDM_VAR_UPDATE_STRUCT lpdmvus,
                        DWORD dwItems,
                        LPVOID lpvUser)
{
    dwTAID;
    CTestCliDoc* pDoc = (CTestCliDoc*)lpvUser;
    for(DWORD i=0; i <dwItems; i++)
    {
        LPDM_VAR_UPDATE_STRUCT lpdmvus2 = &lpdmvus[i];
        CString strData;
        switch(lpdmvus2->dmTypeRef.dwType)
        {
            case DM_VARTYPE_BIT:
                if (lpdmvus2->dmValue.boolVal)
                {
                    strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = TRUE,
Status=%04x"),
                                lpdmvus2->dmVarKey.szName,
                                lpdmvus2->dmVarKey.dwID,
                                lpdmvus2->dwState);
                }
                else
                {
                    strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = FALSE,
Status=%04x"),
                                lpdmvus2->dmVarKey.szName,
                                lpdmvus2->dmVarKey.dwID,
                                lpdmvus2->dwState);
                }
                break;
            case DM_VARTYPE_BYTE:
                strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %u, Status=
%04x"),
                                lpdmvus2->dmVarKey.szName,
                                lpdmvus2->dmVarKey.dwID,
                                lpdmvus2->dmValue.bVal,
                                lpdmvus2->dwState);
                break;
            case DM_VARTYPE_SBYTE:
                strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %d, Status=
%04x"),
                                lpdmvus2->dmVarKey.szName,
                                lpdmvus2->dmVarKey.dwID,
                                lpdmvus2->dmValue.bVal,
                                lpdmvus2->dwState);
        }
    }
}

```

```

        break;
    case DM_VARTYPE_WORD:
        strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %u, Status=
%04x"),
            lpdmvus2->dmVarKey.szName,
            lpdmvus2->dmVarKey.dwID,
            lpdmvus2->dmValue.iVal,
            lpdmvus2->dwState);
        break;
    case DM_VARTYPE_SWORD:
        strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %d, Status=
%04x"),
            lpdmvus2->dmVarKey.szName,
            lpdmvus2->dmVarKey.dwID,
            lpdmvus2->dmValue.iVal,
            lpdmvus2->dwState);
        break;
    case DM_VARTYPE_DWORD:
        strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %u, Status=
%04x"),
            lpdmvus2->dmVarKey.szName,
            lpdmvus2->dmVarKey.dwID,
            lpdmvus2->dmValue.lVal,
            lpdmvus2->dwState);
        break;
    case DM_VARTYPE_SDWORD:
        strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %d, Status=
%04x"),
            lpdmvus2->dmVarKey.szName,
            lpdmvus2->dmVarKey.dwID,
            lpdmvus2->dmValue.lVal,
            lpdmvus2->dwState);
        break;
    case DM_VARTYPE_FLOAT:
        strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %f, Status=
%04x"),
            lpdmvus2->dmVarKey.szName,
            lpdmvus2->dmVarKey.dwID,
            lpdmvus2->dmValue.fltVal,
            lpdmvus2->dwState);
        break;
    case DM_VARTYPE_DOUBLE:
        strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %f, Status=
%04x"),
            lpdmvus2->dmVarKey.szName,
            lpdmvus2->dmVarKey.dwID,
            lpdmvus2->dmValue.dblVal,
            lpdmvus2->dwState);
        break;
    case DM_VARTYPE_TEXT_8:
        strData.Format(_T("GetValueWait: Variable: %s\t( ID = %d ):\tWert = %s, Status=
%04x"),
            lpdmvus2->dmVarKey.szName,
            lpdmvus2->dmVarKey.dwID,
            lpdmvus2->dmValue.bstrVal,
            lpdmvus2->dwState);

```

```

        break;
    case DM_VARTYPE_RAW:
    {
        LPBYTE pArray;
        HRESULT hr;
        hr = SafeArrayAccessData(lpdmvus2->dmValue.parray, (VOID **) &pArray);
        if (! FAILED(hr))
        {
            strData.Format(_T("Var:%s: Type:Rohdatum\nValues:%02x %02x %02x %02x\n"),
                lpdmvus2->dmVarKey.szName,
                pArray[0],
                pArray[1],
                pArray[2],
                pArray[3]);
            SafeArrayUnaccessData(lpdmvus2->dmValue.parray);
        }
        else
        {
            strData.Format(_T("SafeArrayAccessData fehlgeschlagen\n"));
        }
        break;
    }
    default:
        strData.Format(_T("GetValueWait: Unbekannter Variablentyp !"));
    }
    pDoc->PutStr(strData);
}
return(TRUE);
}

//-----< OnTestVariablenGetvaluwait >-----
void CTestCliDoc::OnTestVariablenGetvaluwait()
{
    int nNum = GetVarCount();
    LPDM_VARKEY lpdmVarKey = GetVarKeys();
    LPDM_VAR_UPDATE_STRUCT lpdmvus = new DM_VAR_UPDATE_STRUCT[nNum];
    memset(lpdmvus, 0, sizeof(DM_VAR_UPDATE_STRUCT) * nNum);
    DWORD dwTAID = 5;
    DWORD dwTimeOut = 5000;
    CCmnError Error;
    memset(&Error, 0, sizeof(CCmnError));
    if(!DMGetValueWait(&dwTAID,
        lpdmVarKey,
        nNum,
        FALSE,
        dwTimeOut,
        GetValueWaitNotify,
        this,
        &Error))
    {
        Error.Show(_T("DMGetValueWait (FALSE) failed\n"));
    }
    delete []lpdmVarKey;
    delete []lpdmvus;
}

```

```
//-----< OnTestVariablenGetvaluewait >-----  
void CTestCliDoc::OnTestVariablenGetvaluewaitTrue()  
{  
    int nNum = GetVarCount();  
    LPDM_VARKEY lpdmVarKey = GetVarKeys();  
    LPDM_VAR_UPDATE_STRUCT lpdmvus = new DM_VAR_UPDATE_STRUCT[nNum];  
    memset(lpdmvus, 0, sizeof(DM_VAR_UPDATE_STRUCT) * nNum);  
    DWORD dwTAID = 5;  
    DWORD dwTimeOut = 5000;  
    CCmnError Error;  
    memset(&Error, 0, sizeof(CCmnError));  
    if( !DMGetValueWait(&dwTAID,  
                        lpdmVarKey,  
                        nNum,  
                        TRUE,  
                        dwTimeOut,  
                        GetValueWaitNotify,  
                        this,  
                        &Error))  
    {  
        Error.Show(_T("DMGetValueWait (TRUE) failed\n"));  
    }  
    delete []lpdmVarKey;  
    delete []lpdmvus;  
}  
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

[DMGetValueWait \(Seite 353\)](#)

2.2.11.23 OnTestVariablenGetVarInfo (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenGetVarInfo (MCP)}  
//{{FUNCTION}DMGetVarInfo (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenGetVarInfo()  
{  
    int nNum = GetVarCount();  
    LPDM_VARKEY lpdmVarKey = GetVarKeys();  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    if(!::DMGetVarInfo(m_strProject, lpdmVarKey, nNum, &Error))  
    {  
        Error.Show(_T("DMGetVarInfo failed\n"));  
    }  
    else  
    {  
        CString strData;  
        for(int i=0; i<nnum; i++)  
        {  
            strData.Format(_T("DMGetVarInfo: Variablenname: %s, VarID:%lu."),  
                lpdmVarKey[i].szName,  
                lpdmVarKey[i].dwID);  
            PutStr(strData);  
        }  
        delete []lpdmVarKey;  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

[DMGetVarInfo \(Seite 360\)](#)

2.2.11.24 OnTestVariablenGetvarlimits (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}OnTestVariablenGetvarlimits (MCP)}
//{{FUNCTION}DMGetVarLimits (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestVariablenGetvarlimits()
{
    int nNum = GetVarCount();
    LPDM_VARLIMIT lpdmVarLimit = new DM_VARLIMIT[nNum];
    memset(lpdmVarLimit, 0, sizeof(DM_VARLIMIT) * nNum);
    LPDM_VARKEY lpdmVarKey = GetVarKeys();
    CCmnError Error;
    memset(&Error, 0, sizeof(CCmnError));
    if(!::DMGetVarLimits(m_strProject, lpdmVarKey, nNum,
        lpdmVarLimit, &Error))
    {
        Error.Show(__T("DMGetVarLimits failed\n"));
    }
    else
    {
        CString strData;
        for(int i=0; i<nnum; i++)
        {
            strData.Format(__T("DMGetVarLimits: Var:%s, MaxRange:%lf, MinRange:%lf, MinLimit:
%lf, MaxLimit:%lf."),
                lpdmVarKey[i].szName,
                lpdmVarLimit[i].dmMaxRange.dblVal,
                lpdmVarLimit[i].dmMinRange.dblVal,
                lpdmVarLimit[i].dmMaxLimit.dblVal,
                lpdmVarLimit[i].dmMinLimit.dblVal);
            PutStr(strData);
        }
    }
    delete []lpdmVarLimit;
    delete []lpdmVarKey;
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

[DMGetVarLimits \(Seite 372\)](#)

2.2.11.25 OnTestVariablenGetvartype (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenGetvartype (MCP)}  
//{{FUNCTION}DMGetVarType (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenGetvartype()  
{  
    int nNum = GetVarCount();  
    LPDM_TYPEREF lpdmTypeRef = new DM_TYPEREF[nNum];  
    memset(lpdmTypeRef, 0, sizeof(DM_TYPEREF) * nNum);  
    LPDM_VARKEY lpdmVarKey = GetVarKeys();  
    CCmnError Error;  
    memset(&Error, 0, sizeof(CCmnError));  
    if(!::DMGetVarType(m_strProject, lpdmVarKey, nNum,  
        lpdmTypeRef, &Error))  
    {  
        Error.Show(_T("DMGetVarType failed\n"));  
    }  
    else  
    {  
        CString strData;  
        for(int i=0; i<nnum; i++)  
        {  
            strData.Format(_T("DMGetVarType: Var:%s, Size:%lu, Name:%s."),  
                lpdmVarKey[i].szName,  
                lpdmVarKey[i].dwID);  
            PutStr(strData);  
        }  
    }  
    delete []lpdmVarKey;  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

[DMGetVarType \(Seite 378\)](#)

2.2.11.26 OnTestVariablenResumevarupdate (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenResumevarupdate (MCP)}  
//{{FUNCTION}DMResumeVarUpdate (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenResumevarupdate()  
{  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    if(!DMResumeVarUpdate(m_dwTAID, &Error))  
    {  
        Error.Show(_T("DMResumeVarUpdate failed.\n"));  
    }  
    else  
    {  
        CString strData;  
        strData.Format(_T("DMResumeVarUpdate ok."));  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DMResumeVarUpdate (Seite 454)

2.2.11.27 OnTestVariablenSetvalue (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenSetvalue (MCP)}  
//{{FUNCTION}DMSetValue (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenSetvalue()  
{  
    int nNum = GetVarCount();  
    LPDM_VARKEY lpdmVarKey = GetVarKeys();  
    LPDWORD lpdmVarState = new DWORD[nNum];  
    INT i;  
    CString strData;  
    CCmnError cmnError;  
    for (i = 0; i < 1; i++)</p>  
    {  
        DWORD dwStart = GetTickCount();  
        DWORD dwEnd;  
        if (! DMSetValue(lpdmVarKey, nNum, m_varValues, lpdmVarState, cmnError))  
        {  
            break;  
        }  
        dwEnd = GetTickCount();  
        strData.Format("DMSetValue OK, Dauer %d ms", dwEnd - dwStart);  
        PutStr(strData);  
    }  
    delete []lpdmVarKey;  
    delete []lpdmVarState;  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

[DMSetValue \(Seite 385\)](#)

2.2.11.28 OnTestVariablenSetvaluawait (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenSetvaluawait (MCP)}  
//{{FUNCTION}DMSetValueWait (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenSetvaluawait()  
{  
    int nNum = GetVarCount();  
    LPDM_VARKEY lpdmVarKey = GetVarKeys();  
    LPDWORD lpdmVarState = new DWORD[nNum];  
    DWORD dwTimeOut = 1000L;  
    CCmnError cmnError;  
    if(!::DMSetValueWait(&m_dwTAID, lpdmVarKey, nNum, m_varValues, dwTimeOut,  
        CompletionProc, this, cmnError))  
    {  
        cmnError.Show(_T("DMSetValueWait failed\n"));  
    }  
    else  
    {  
        CString strData;  
        strData.Format(_T("DMSetValueWait ok.));  
    }  
    delete []lpdmVarKey;  
    delete []lpdmVarState;  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

[DMSetValueWaitMessage \(Seite 400\)](#)

[DMSetValueWait \(Seite 394\)](#)

2.2.11.29 OnTestVariablenStopallupdates (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenStopallupdates (MCP)}  
//{{FUNCTION}DMStopAllUpdates (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenStopallupdates()  
{  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    if(!DMStopAllUpdates(&Error))  
    {  
        Error.Show(_T("DMStopAllUpdates failed\n"));  
    }  
    else  
    {  
        CString strData;  
        strData.Format(_T("DMStopAllUpdates ok."));  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DMStopAllUpdates (Seite 471)

2.2.11.30 OnTestVariablenStopvarupdate (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}OnTestVariablenStopvarupdate (MCP)}  
//{{FUNCTION}DMStopVarUpdate (MCP)}  
//{{FUNCTION}(END)}  
void CTestCliDoc::OnTestVariablenStopvarupdate()  
{  
    CCmnError Error;  
    memset(&Error,0,sizeof(CCmnError));  
    if( !DMStopVarUpdate(m_dwTAID, &Error))  
    {  
        Error.Show(_T("DMStopVarUpdate failed\n"));  
    }  
    else  
    {  
        CString strData;  
        strData.Format(_T("DMStopVarUpdate: TAID:%lu."), m_dwTAID);  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

DMStopVarUpdate (Seite 472)

2.2.11.31 OnTestVariablenSuspendvarupdate (RT Professional)**Beispiel**

```

//{{ODK_EXAMPLE}OnTestVariablenSuspendvarupdate (MCP)}
//{{FUNCTION}DMSuspendVarUpdate (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestVariablenSuspendvarupdate()
{
    CCmnError Error;
    memset(&Error,0,sizeof(CCmnError));
    if(!DMSuspendVarUpdate(m_dwTAID, &Error))
    {
        Error.Show(_T("DMSuspendVarUpdate failed\n"));
    }
    else
    {
        CString strData;
        strData.Format(_T("DMSuspendVarUpdate ok."));
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMSuspendVarUpdate (Seite 473)

2.2.11.32 OnTestWinCCShutdown (RT Professional)**Beispiel**

```

//{{ODK_EXAMPLE}OnTestWinCCShutdown (MCP)}
//{{FUNCTION}DMEExitWinCC (MCP)}
//{{FUNCTION}(END)}
void CTestCliDoc::OnTestWinCCShutdown()
{
    DMEExitWinCC();
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DMEExitWinCC (Seite 292)

2.2.11.33 Open project by means of dialog (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Open projectby means of dialog (MCP)}
//{{FUNCTION}DmOpenProject (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: MyDMOpenProject(void) ODK DM CS
// =====
// Desc. : open project, give name through WinCC-Dialog-Box
// =====
void MyDMOpenProject(void)
{
    CMN_ERROR Error;
    BOOL ret = FALSE;
    DWORD dwSize = 255;
    HWND handle = NULL;
    TCHAR szProjFile[255];
    TCHAR szText[255];
    _tcsncpy_s(szProjFile, _countof(szProjFile), _T("C:\\Siemens\\ODK\\Samples\\Projects\\
\\Demo\\odk.mcp"), _TRUNCATE);
    memset(&Error, 0, sizeof( CMN_ERROR ));
    ret = MyDMConnect(); // check connection to DataManager
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in MyDMConnect"));
        ODKTrace(szText);
    }
    else
    {
        memset(&Error, 0, sizeof( CMN_ERROR ));
        ret = DmOpenProject(handle, szProjFile, dwSize, &Error);
        if(ret == FALSE)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in DmOpenProject:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("DmOpenProject (%s)",
szProjFile);
        }
        ODKTrace(szText);
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

DmOpenProjectPlus (Seite 313)

2.2.11.34 Read tag (RT Professional)

Beispiel

```
// =====
// =====
// short : Modul with examples to DataManager-API
// RUNTIME
// SC: PO = Projekt opened PG= Projekt closed
// *****
#include "stdafx.h" // if MFC classes
// #include "odkapi.h" // if console application
#include <time.h>
#include "dm02.h"

// =====
// 2.1 Interface IMPORT
// =====
//extern void ODKTrace(LPCTSTR);
extern BOOL MyDMConnect (void); // Connect
extern BOOL MyDMGetConnectionState(void);

// =====
// 2.2 Interface EXPORT
// =====
// 2.3 Interface LOCAL
// =====
// 3. Definitions
// =====

//{{ODK_EXAMPLE}Read tag (MCP)}
//{{FUNCTION}DMGetRuntimeProject (MCP)}
//{{FUNCTION}DMGetValue (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: MyDMGetValue(void) ODK DM CS
// =====
// short : Read tag
// =====
void MyDMGetValue(void)
{
    CMN_ERROR Error;
    BOOL ret = FALSE;
    DWORD dwSize = _MAX_PATH;
    TCHAR szText[255];
    TCHAR szProjFile[_MAX_PATH + 1];
    TCHAR BstrValue[255]; //
    const short int nNum = 11; // tagcount
    const short int nTextMax = 20; // textlength
    TCHAR TagNames[nNum][nTextMax]=
    {
        "VAR_1_BOOL", // boolVal; // VT_BOOL.

```

2.2 Funktionen der Datenhaltung (RT Professional)

```

"VAR_2_BYTE", // bVal; // VT_UI1
"VAR_3_SBYTE", // iVal; // VT_I2
"VAR_4_WORD", // lVal // VT_I4
"VAR_5_SWORD", // iVal // VT_I2
"VAR_6_DWORD", // dblVal // VT_R8
"VAR_7_SDWORD", // lVal // VT_I4
"VAR_8_FLOAT", // fltVal // VT_R4
"VAR_9_DOUBLE", // dblVal // VT_R8
"VAR_A_TEXT8", // bstrVal // VT_BSTR
"VAR_B_TEXT16" // bstrVal // VT_BSTR
};
DM_VARKEY VarKey[nNum];

DM_VAR_UPDATE_STRUCT VarUp[nNum];
memset(&VarUp, 0, sizeof(DM_VAR_UPDATE_STRUCT) * nNum);
// the included VARIANT's in the DM_VAR_UPDATE_STRUCT's then initialized to VT_EMPTY
with the memset,
// don't do this later again, because VT_BSTR's can be present after DMGetValue(...)!

memset(&VarKey,0, sizeof(DM_VARKEY) * nNum);
memset(&Error,0,sizeof(Error));
ret = MyDMGetConnectionState();
if(FALSE != ret)
{
    memset(&Error,0,sizeof(Error));
    // RunTime project
    ret = DMGetRuntimeProject(szProjFile, dwSize, &Error);
    if(FALSE != ret)
    {
        // fill out DM_VARKEY
        for(int iRead = 0; iRead <nNum; iRead++)
        {
            VarKey[iRead].dwKeyType = DM_VARKEY_NAME;
            VarKey[iRead].dwID = 0;
            strcpy( VarKey[iRead].szName, &TagNames[iRead][0]);
            VarKey[iRead].lpvUserData = (VOID *) iRead;
        }
        memset(&Error,0,sizeof(Error));
        ret = DMGetValue(VarKey, nNum, VarUp, &Error);
        if(FALSE != ret)
        {
            for(int iOut=0; iOut <nNum; iOut++)
            {
                switch(VarUp[iOut].dmTypeRef.dwType)
                {
                    case DM_VARTYPE_BIT: //vt = 3 VT_I4 = 3
                    {
                        sprintf(szText, "Index=%d Name=%s ID=%d Value=%d",
                            iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
                            VarUp[iOut].dmValue.boolVal);
                        break;
                    }
                    case DM_VARTYPE_BYTE: //vt = 17 VT_UI1 = 17
                    {
                        sprintf(szText, "Index=%d Name=%s ID=%d Value=%d",

```

```

        iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.bVal);
        break;
    }
    case DM_VARTYPE_SBYTE: //vt = 2 VT_I2 = 2
    {
        sprintf(szText, "Index=%d Name=%s ID=%d Value=%d",
iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.iVal);
        break;
    }
    case DM_VARTYPE_WORD: //vt = 3 VT_I4 = 3
    {
        sprintf(szText, "Index=%d Name=%s ID=%d Value=%d",
iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.lVal);
        break;
    }
    case DM_VARTYPE_SWORD: //vt = 2 VT_I2 = 2
    {
        sprintf(szText, "Index=%d Name=%s ID=%d Value=%d",
iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.iVal);
        break;
    }
    case DM_VARTYPE_DWORD: //vt = 5 VT_R8 = 5
    {
        sprintf(szText, "Index=%d Name=%s ID=%d Value=%d",
iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.dblVal);
        break;
    }
    case DM_VARTYPE_SDWORD: //vt = 3 VT_I4 = 3
    {
        sprintf(szText, "Index=%d Name=%s ID=%d Value=%d",
iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.lVal);
        break;
    }
    case DM_VARTYPE_FLOAT: //vt = 4 VT_R4 = 4
    {
        sprintf(szText, "Index=%d Name=%s ID=%d Value=%f",
iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.fltVal);
        break;
    }
    case DM_VARTYPE_DOUBLE: //vt = 5 VT_R8 = 5
    {
        sprintf(szText, "Index=%d Name=%s ID=%d Value=%f",

```

```

        iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID,
        VarUp[iOut].dmValue.dblVal);
        break;
    }
    case DM_VARTYPE_TEXT_8: //vt = 8 VT_BSTR = 8
    {
        ret = WideCharToMultiByte( CP_ACP,
(DWORD)0, VarUp[iOut].dmValue.bstrVal,
        -1, (LPSTR) &BstrValue[0], 128, NULL, NULL);
        sprintf(szText, "Index=%d Name=%s ID=%d StrValue=%s",
            iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID, BstrValue);
        break;
    }
    case DM_VARTYPE_TEXT_16: // vt = 8 VT_BSTR = 8
    {
        ret = WideCharToMultiByte( CP_ACP,
(DWORD)0, VarUp[iOut].dmValue.bstrVal,
        -1, (LPSTR) &BstrValue[0], 128, NULL, NULL);
        sprintf(szText, "Index=%d Name=%s ID=%d StrValue=%s",
            iOut, VarUp[iOut].dmVarKey.szName,
VarUp[iOut].dmVarKey.dwID, BstrValue);
        break;
    }
    default:
        break;
    } // end switch case
    ODKTrace(szText);
    //printf("%s\r\n", szText);

    VariantClear( &VarUp[iOut].dmValue );
} //end for
}
else
{
    sprintf(szText, "Error in DMGetValue: E1= 0x%08lx ; E2= 0x%08lx ; %s",
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    //printf("%s\r\n", szText);
}
}
else
{
    sprintf(szText, "Error in DMGetRuntimeProject: E1= 0x%08lx ; E2= 0x%08lx ; %s",
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    //printf("%s\r\n", szText);
}
}
else
{
    sprintf(szText, "Error in MyDMGetConnectionState: E1= 0x%08lx ; E2= 0x%08lx ; %s",
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    //printf("%s\r\n", szText);
}
}

```

```
    }  
}  
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

[DMGetRuntimeProject \(Seite 311\)](#)

[DMGetValue \(Seite 341\)](#)

2.2.11.35 Write tag (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Write tag (MCP)}
//{{FUNCTION}DMGetRuntimeProject (MCP)}
//{{FUNCTION}DMSetValue (MCP)}
//{{FUNCTION}(END)}
// =====
// Function: MyDMSetValue(void) ODK DM CS
// =====
// short : Write tag
// =====
void MyDMSetValue(void)
{
    CMN_ERROR Error;
    BOOL ret = FALSE;
    DWORD dwSize = _MAX_PATH;
    TCHAR szText[255];
    TCHAR szProjFile[_MAX_PATH +1];
    TCHAR szTextValue[255]; // for TEXT8 Values
    const short int nNum = 11; // tagcount
    const short int nTextMax = 20; // textlength
    TCHAR TagNames[nNum][nTextMax]=
    {
        "VAR_1_BOOL", // boolVal; // VT_BOOL.
        "VAR_2_BYTE", // bVal; // VT_UI1
        "VAR_3_SBYTE", // iVal; // VT_I2
        "VAR_4_WORD", // lVal // VT_I4
        "VAR_5_SWORD", // iVal // VT_I2
        "VAR_6_DWORD", // dblVal // VT_R8
        "VAR_7_SDWORD", // lVal // VT_I4
        "VAR_8_FLOAT", // fltVal // VT_R4
        "VAR_9_DOUBLE", // dblVal // VT_R8
        "VAR_A_TEXT8", // bstrVal // VT_BSTR
        "VAR_B_TEXT16" // bstrVal // VT_BSTR
    };
    DM_VARKEY VarKey[nNum];
    DM_VAR_UPDATE_STRUCT VarUp[nNum];
    VARIANT VarVal[nNum];
    DWORD VarSta[nNum];
    static short int Tmp[20];
    BSTR bstrText1, bstrText2;
    memset(&Error, 0, sizeof(CMN_ERROR));
    memset(&VarUp, 0, sizeof(DM_VAR_UPDATE_STRUCT) * nNum);
    memset(&VarKey, 0, sizeof(DM_VARKEY) * nNum);
    ret = MyDMGetConnectionState();
    if(FALSE != ret)
    {
        // RunTime project
        ret = DMGetRuntimeProject(szProjFile, dwSize, &Error);
        if(FALSE != ret)
        {
            // read tag

```

```

for(int i = 0; i < nNum; i++)
{
    VarKey[i].dwKeyType = DM_VARKEY_NAME;
    VarKey[i].dwID = 0;
    _tcsncpy_s(VarKey[i].szName, _countof(VarKey[i].szName), &TagNames[i][0],
_TRUNCATE);
    VarKey[i].lpvUserData = (VOID *) i;
}
// init tag
// "VAR_1_BOOL",
VarVal[0].vt = VT_BOOL;
VarVal[0].boolVal = TRUE;
// "VAR_2_BYTE"
VarVal[1].vt = VT_UI1;
VarVal[1].bVal = 23;
// "VAR_3_SBYTE",
VarVal[2].vt = VT_I2;
VarVal[2].iVal = -23;
// "VAR_4_WORD",
VarVal[3].vt = VT_I4;
VarVal[3].lVal = 89;
// "VAR_5_SWORD",
VarVal[4].vt = VT_I2;
VarVal[4].iVal = -89;
// "VAR_6_DWORD",
VarVal[5].vt = VT_R8;
VarVal[5].dblVal = 1200;
// "VAR_7_SDWORD",
VarVal[6].vt = VT_I4;
VarVal[6].lVal = -1200;
// "VAR_8_FLOAT",
VarVal[7].vt = VT_R4;
VarVal[7].fltVal = (float)3.789;
// "VAR_9_DOUBLE",
VarVal[8].vt = VT_R8;
VarVal[8].dblVal = (double)34.789;
// "VAR_A_TEXT8",
_tcsncpy_s(szTextValue, _countof(szTextValue), _T("VAR_A_TEXT8 Value"),
_TRUNCATE);
ret = MultiByteToWideChar(CP_ACP, (DWORD)0, (LPCSTR) &szTextValue, -1,
(LPWSTR) &Tmp[0], 30);
// SysFreeString(pBSTR);
bstrText1 = SysAllocString((OLECHAR FAR*)&Tmp[0]);
VarVal[9].vt = VT_BSTR;
VarVal[9].bstrVal = bstrText1;
// "VAR_B_TEXT16"
_tcsncpy_s(szTextValue, _countof(szTextValue), _T("VAR_B_TEXT16 Value"),
_TRUNCATE);
ret = MultiByteToWideChar(CP_ACP, (DWORD)0, (LPCSTR) &szTextValue, -1,
(LPWSTR) &Tmp[0], 30);
// SysFreeString(pB2);
bstrText2 = SysAllocString((OLECHAR FAR*)&Tmp[0]);
VarVal[10].vt = VT_BSTR;
VarVal[10].bstrVal = bstrText2;
// set tag
memset(&Error, 0, sizeof(CMN_ERROR));

```

2.3 Funktionen des Grafiksystems (RT Professional)

```

ret = DMSetValue(&VarKey[0], nNum, &VarVal[0], &VarSta[0], &Error);
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in DMSetValue:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("write Value to WinCC"));
    ODKTrace(szText);
}
SysFreeString(bstrText1);
SysFreeString(bstrText2);
}
}
}
//{{ODK_EXAMPLE}(END)}
    
```

Siehe auch

- DMGetRuntimeProject (Seite 311)
- DMSetValue (Seite 385)

2.3 Funktionen des Grafiksystems (RT Professional)

2.3.1 Grundlagen (RT Professional)

2.3.1.1 Übersicht über die Funktionen (RT Professional)

Übersicht

PDLRTShowApp (Seite 566)	Grafiksystem anzeigen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion der Runtime-Funktionen
PDLRTCclosePicture (Seite 555)	Bild schließen
PDLRTDisableClosePicture (Seite 557)	Bild schließen verhindern
PDLRTEnableClosePicture (Seite 558)	Bild zum Schließen freigeben
PDLRTGetCursorKeys (Seite 570)	Cursorsteuerungstasten abfragen
PDLRTGetDefPropEx (Seite 578)	Objekteigenschaften Defaultwerte abfragen

PDLRTGetPropEx (Seite 581)	Objekteigenschaften Defaultwerte abfragen
PDLRTGetFocus (Seite 572)	Eingabefocus abfragen
PDLRTGetLink (Seite 588)	Verknüpfung Objekteigenschaft-Variable abfragen
PDLRTGotoPicture (Seite 560)	Bildspeicher abfragen
PDLRTInquireFreeArea (Seite 561)	freien Bereich eines Fensters ermitteln
PDLRTOpenPicture (Seite 563)	Grundbildwechsel im Runtime
PDLRTPictureNavigation (Seite 565)	Bildnavigation ein- und ausschalten
PDLRTSetCursorKeys (Seite 574)	Cursorsteuerungstasten festlegen
PDLRTSetFocus (Seite 576)	Eingabefocus festlegen
PDLRTSetLink (Seite 590)	Verknüpfung Objekteigenschaft-Variable festlegen
PDLRTSetMultiLink (Seite 592)	Verknüpfung Objekteigenschaft-Variable festlegen (mehrere Variablen)
PDLRTSetPropEx (Seite 584)	Objekteigenschaft festlegen

2.3.1.2 Übersicht über die Strukturen (RT Professional)

Übersicht

FOCUSINFO (Seite 554)	Eingabefocusdaten
LINKINFO (Seite 551)	Verknüpfung von Objekteigenschaften und Variablen
MULTILINK (Seite 552)	Verknüpfung von Objekteigenschaften und Variablen
MULTILINKINFO (Seite 553)	Verknüpfung von Objekteigenschaften und Variablen

2.3.1.3 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

Konfiguration

PDLCS_ELIASES_NO_ERROR	0	Es ist kein Fehler aufgetreten
PDLCS_ERROR_NO_DISPATCH	5	Ole-Automation-Ausführung nicht möglich
PDLCS_ERROR_NO_PROJECT	10	Projekt existiert nicht
PDLCS_ERROR_NO_PICTURE	15	Bild existiert nicht
PDLCS_ERROR_NO_OBJECT	20	Objekt existiert nicht
PDLCS_ERROR_NO_PROPERTY	25	Property existiert nicht
PDLCS_ERROR_NO_OBJTYPE	30	Objekttyp ist nicht vorhanden.

2.3 Funktionen des Grafiksystems (RT Professional)

PDLCS_ERROR_NO_DYNAMIC	32	Keine Dynamik vorhanden
PDLCS_ERROR_ALREADY_OPEN	35	angegebenes Bild ist bereits geöffnet
PDLCS_ERROR_WRITE_PROTECTED	40	Angegebenes Bild ist schreibgeschützt
PDLCS_ERROR_PICTURE_SAVE	45	Angegebenes Bild konnte nicht gesichert werden
PDLCS_ERROR_PICTURE_SAVE_AS	50	angegebenes Bild konnte nicht gesichert werden
PDLCS_ERROR_PICTURE_CLOSE	55	angegebenes Bild konnte nicht geschlossen werden
PDLCS_ERROR_PICTURE_CLOSE_ALL	60	Bilder konnten nicht geschlossen werden
PDLCS_ERROR_PICTURE_IMPORT	65	Bilder konnten nicht importiert werden
PDLCS_ERROR_PICTURE_CREATE	70	Allgemeiner Fehler beim Erzeugen des Bildes.
PDLCS_ERROR_PICTURE_EXISTS	75	Bild existiert bereits
PDLCS_ERROR_NEW_OBJECT_NAME	80	Objektname schon vorhanden
PDLCS_ERROR_NEW_OBJECT	85	Objekt konnte nicht erzeugt werden
PDLCS_ERROR_DELETE_OBJECT	90	Objekt konnte nicht gelöscht werden
PDLCS_ERROR_SET_PROPERTY	95	Property konnte nicht gesetzt werden
PDLCS_ERROR_GET_PROPERTY	96	Property konnte nicht gelesen werden
PDLCS_ERROR_ENUM_OBJECTTYPE	100	Fehler bei der Enumeration der Objekttypen.
PDLCS_ERROR_ENUM_PROPERTY	105	Fehler bei der Enumeration der Objekteigenschaften.
PDLCS_ERROR_CREATE_PROPERTY	110	Property konnte nicht definiert werden
PDLCS_ERROR_SET_LINK	115	Verbindung konnte nicht hergestellt werden.
PDLCS_ERROR_REMOVE_LINK	120	Verbindung konnte nicht gelöscht werden.
PDLCS_ERROR_SET_ACTION	125	Aktion konnte nicht hinterlegt werden.
PDLCS_ERROR_GET_ACTION	126	Aktion konnte nicht geholt werden.
PDLCS_ERROR_ENUM_DYNAMICS	127	Dynamiken konnten nicht enumeriert werden.
PDLCS_ERROR_SET_DIRECTCONNECT	128	Direktverbindung konnte nicht hinterlegt werden.
PDLCS_ERROR_AUTOMATION	130	Fehler bei der Übertragung
PDLCS_ERROR_MEMORY	200	Kein Speicher

Runtime

PDLRT_OK	0	Es ist kein Fehler aufgetreten
PDLRT_APP_NOT_RUNNING	1	PDL-Applikation nicht gestartet
PDLRT_NO_TRANSFER	2	PDL-Applikation nicht ansprechbar
PDLRT_NO_PIC	3	Bild nicht angewählt
PDLRT_NO_OBJ	4	Objekt nicht gefunden
PDLRT_NO_PROP	5	Property nicht gefunden
PDLRT_NO_MET	6	Methode am Objekt nicht gefunden
PDLRT_LINK_NOT_SET	7	Property konnte nicht verbunden werden
PDLRT_WNAME_NOT_UNIQUE	8	Instanzname des neuen Fensters nicht eindeutig
PDLRT_WCREATION_FAILED	9	Fenster konnte nicht kreiert werden
PDLRT_PICTURE_NOT_LOADED	10	Bilddatei konnte nicht geladen werden

PDLRT_ILLEGAL_USERLEVEL	11	Unzureichende Bedienberechtigungsstufe
PDLRT_NO_POSITION	12	Rechteck kann nicht platziert werden.
PDLRT_POSITION_WITH_MODY	13	Rechteck wurde modifiziert
PDLRT_E_ALREADY_CONNECTED	14	eine andere Verbindung zum PDLRT besteht bereits
PDLRT_PICTURE_ALREADY_OPEN	15	Bild ist bereits geöffnet
PDLRT_BAD_OLE_CONVERSION	16	Fehler bei Wandlung über OLE Automation
PDLRT_NO_LINK	17	Property hat keine Dynamik
PDLRT_IND_LINK_READ_ERR	18	Fehler beim lesen der indirekten Variablen
PDLRT_NO_IND_LINK	19	Property hat keine indirekte Dynamik
PDLRT_FAILURE_PARAM	20	Fehlerhafte Parameter

2.3.1.4 Konstanten (RT Professional)

Adressierungsmodus

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt erfolgt relativ
PDLRT_AM_PICTABS	1	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objektes ist absolut

Die Werte PDLRT_AM_PICTABS und PDLRT_AM_OBJABS können verodert werden.

PDLRTInquireFreeArea

PDLRT_IQ_ONLY	0x01	Nur Nachfrage mit dem Ergebnis: "passt/passt nicht"
PDLRT_IQ_MODY_POSX	0x02	Die x-Position des Rechtecks darf so modifiziert werden, dass es platziert werden kann.
PDLRT_IQ_MODY_POSY	0x04	Die y-Position des Rechtecks darf so modifiziert werden, dass es platziert werden kann.
PDLRT_IQ_MODY_HEIGHT	0x08	Die Höhe des Rechtecks darf so modifiziert werden, dass es platziert werden kann.
PDLRT_IQ_MODY_WIDTH	0x10	Die Breite des Rechtecks darf so modifiziert werden, dass es platziert werden kann.
PDLRTOpenPicture		
PDLRT_WS_STYLE_FROM_PIC	0	Es werden die Anzeigeattribute aus den Bilddaten übernommen
PDLRT_WS_BORDER	WS_BORDER	Fenster mit Rahmen
PDLRT_WS_CAPTION	WS_CAPTION	Fenster mit Titel und Rahmen
PDLRT_WS_THICKFRAME	WS_THICKFRAME	Fenster mit dickem Rahmen. Kann durch Mausbedienung in der Größe verändert werden.

PDLRT_WS_HSCROLL	WS_HSCROLL	Fenster mit horizontalem Slider
PDLRT_WS_VSCROLL	WS_VSCROLL	Fenster mit vertikalem Slider

Aktualisierungszyklen

Name	Index
"Bei Änderung"	0
"250 ms"	1
"500 ms"	2
"1 s"	3
"2 s"	4
"5 s"	5
"10 s"	6
"1 min"	7
"5 min"	8
"10 min"	9
"1 h"	10
"Anwenderzyklus 1"	11
"Anwenderzyklus 2"	12
"Anwenderzyklus 3"	13
"Anwenderzyklus 4"	14
"Anwenderzyklus 5"	15
"Fensterzyklus"	235
"Bildzyklus"	255

2.3.1.5 Liste der Objekteigenschaften (A-K) (Graphics Designer) (RT Professional)

Übersicht

Hinweis

Bei vielen API-Funktionen des Graphics Designer müssen Sie den Parameter `IpszPropName` versorgen. Dabei muss der englische Name einer Objekteigenschaft angegeben werden.

Properties vom Typ `VT_USERDEFINED` sind nicht mit den Get/Set-Property-Funktionen zu bearbeiten.

Es darf kein `VT_DISPATCH` und andere Referenzen verwendet werden, lediglich die normalen Typen und max. ein `VT_VARIANT` für ein Array-Property einfacher Typen.

OLE-Automation Name	Datentyp		Property name	Attributname	Objektzugehörigkeit
"ActualPointLeft"	VT_I4		Current Value X	Aktueller Wert X	Polygon,

"ActualPointTop"	VT_I4		Current Value Y	Aktueller Wert Y	Polygon,
"AdaptBorder"	VT_BOOLEAN		Adapt Border	Rahmenanpassung	Button, Statischer Text, E/A-Feld, Textliste, Check-Box, Radio-Box,
"AdaptPicture"	VT_BOOLEAN		Adapt Picture	Bild anpassen	Bildfenster,
"AdaptSize"	VT_BOOLEAN		Adapt Size	Fenster anpassen	Bildfenster,
"AdaptWidth"					
"AlarmHigh"	VT_R8		Limit AH	Obergrenze AH	Balken,
"AlarmLow"	VT_R8		Limit AL	Untergrenze AL	Balken,
"Alignment"	VT_BOOLEAN		Alignment	Ausrichtung	Balken,
"AlignmentLeft"	VT_I4		X Alignment	X-Ausrichtung	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"AlignmentTop"	VT_I4		Y Alignment	Y-Ausrichtung	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"Analog"	VT_BOOLEAN	OCX	Analog	Analog	DAClockCtrl,
"AngleAlpha"	VT_I4	BPC	Alpha	Alpha	3D-Balken,
"AngleBeta"	VT_I4	BPC	Beta	Beta	3D-Balken,
"AngleMax"	VT_R8	OCX	AngleMax	AngleMax	XGaugeCtrl,
"AngleMin"	VT_R8	OCX	AngleMin	AngleMin	XGaugeCtrl,
"Application"	VT_BSTR		WindowContents	Fensterinhalt	Druckauftrag/Skriptdiagnose,
"ApplicationDataPath"					
"Assignments"	VT_BSTR		Assignments	Zuordnungen	Textliste,
"AssumeOnExit"	VT_BOOLEAN		Apply on Exit	Übernahme bei Verlassen	E/A-Feld, Textliste,
"AssumeOnFull"	VT_BOOLEAN		Apply on Full	Übernahme bei vollst. Eingabe	E/A-Feld,
"AutoScroll"	VT_BOOLEAN	OCX	AutoScroll	AutoScroll	CCAlgWinCtrl,
"AutoSize"	VT_I4	OCX	AutoSize	AutoSize	PButtonCtrl,
"Average"	VT_BOOLEAN		Average Value	Mittelwert	Balken,
"Axe"	VT_I4				
"AxisSection"	VT_R8		Axis Section	Achsenabschnitt	Balken,

2.3 Funktionen des Grafiksystems (RT Professional)

"BackBorderWidth"	VT_I4		3D Border Weight	3D-Rahmenbreite	Button, Sammelanzeige, Rundbutton, Slider-Objekt,
"BackColor"	VT_I4		Background Color	Hintergrundfarbe	Button, Ellipse, Ellipsesegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"BackColor"	VT_UI4		BackColor	BackColor	DAClockCtrl, XGaugeCtrl, PButtonCtrl, SliderCtrl,
"BackColor2"	VT_I4		Bar Color	Balkenfarbe	Balken,
"BackColor3"	VT_I4		Bar Background Color	Balkenhintergrundfarbe	Balken,
"BackColorBottom"	VT_I4		Low Limit Background Color	Untere Hintergrundfarbe	Slider-Objekt,
"BackColorTop"	VT_I4		High Limit Background Color	Obere Hintergrundfarbe	Slider-Objekt,
"BackFlashColorOff"	VT_I4		Flashing Background Color Off	Hintergrundblinkfarbe Aus	Button, Ellipse, Ellipsesegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"BackFlashColorOn"	VT_I4		Flashing Background Color On	Hintergrundblinkfarbe Ein	Button, Ellipse, Ellipsesegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"Background"	VT_BOOLEAN		Background	Hintergrund	3D-Balken,
"BackgroundPicture"	VT_USEDEFINED		BackgroundPicture	BackgroundPicture	XGaugeCtrl,
"BarBackColor"	VT_UI4		BarBackColor	BarBackColor	SliderCtrl,
"BarFillColor"	VT_UI4		BarFillColor	BarFillColor	SliderCtrl,

2.3 Funktionen des Grafiksystems (RT Professional)

"BarDepth"	VT_I4		Bar Depth	Balkentiefe	3D-Balken,
"BarHeight"	VT_I4		Bar Height	Balkenhöhe	3D-Balken,
"BarWidth"	VT_I4		Bar Width	Balkenbreite	3D-Balken,
"BasePicReferenced"	VT_BOOLEAN		Basic Picture Referenced	Grundbild referenziert	Zustandsanzeige,
"BasePicTransparentColor"	VT_I4		Basic Picture Transparent Color	Grundbild Transparentfarbe	Zustandsanzeige,
"BasePicture"	VT_BITMAP		Basic Picture	Grundbild	Zustandsanzeige,
"BasePicUseTransparentColor"	VT_BOOLEAN		Basic Picture Transparent Color On	Grundbild Transparentfarbe Ein	Zustandsanzeige,
"BaseX"	VT_I4		Base X	Basis X	3D-Balken,
"BaseY"	VT_I4		Base Y	Basis Y	3D-Balken,
"BevelInner"	VT_I4		BevelInner	Bevellner	SliderCtrl,
"BevelColorDown"	VT_UI4		BevelColorDown	BevelColorDown	SliderCtrl,
"BevelColorUp"	VT_UI4		BevelColorUp	BevelColorUp	SliderCtrl,
"BevelOuter"	VT_I4		BevelOuter	BevelOuter	XGaugeCtrl,
"BevelWidth"	VT_I4		BevelWidth	BevelWidth	XGaugeCtrl,
"BitNumber"	VT_I4			Bitnummer	Zustandsanzeige,
"BorderBackColor"	VT_I4		Border Background Color, Line BackgroundColor	Rahmen-Hintergrundfarbe, Linien-Hintergrundfarbe	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"BorderColor"	VT_I4		Border Color, Line Background Color	Rahmenfarbe, Linienfarbe	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Balken, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"BorderColorBottom"	VT_I4		3D Shadow Color	3D-Rahmenfarbe unten	Button, Rundbutton,

2.3 Funktionen des Grafiksystems (RT Professional)

"BorderColorTop"	VT_I4		3D Border Color	3D-Rahmenfarbe oben	Button, Rundbutton,
"BorderEndStyle"	VT_UI4		Line End Style	Linienenden	Linie, Polygonzug,
"BorderFlashColorOff"	VT_I4		Flashing Border Color Off, Flashing Line Color Off	Rahmenblinkfarbe Aus, Linienblinkfarbe Aus	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Slider-Objekt,
"BorderFlashColorOn"	VT_I4		Flashing Border Color On, Flashing Line Color On	Rahmenblinkfarbe Ein, Linienblinkfarbe Ein	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"BorderStyle"	VT_UI4		Line Style	Linienart	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,

"BorderWidth"	VT_I4		Line Weight	Linienbreite	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Balken, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"BorderWidth"	VT_I4	OCX	BorderWidth	BorderWidth	XGaugeCtrl,
"BoxAlignment"	VT_BOOLEAN		Box Alignment	Feldausrichtung	Check-Box, Radio-Box,
"BoxCount"	VT_I4		Number of Boxes	Anzahl Felder	Check-Box, Radio-Box,
"BoxType"	VT_I4		Field Type	Feldtyp	E/A-Feld, Textliste,
"Button1Width"	VT_I4		Button 1 Width	Breite Button 1	Sammelanzeige,
"Button2Width"	VT_I4		Button 2 Width (Error: Button 1 Width 2)	Breite Button 2	Sammelanzeige,
"Button3Width"	VT_I4		Button 3 Width (Error: Button 1 Width 3)	Breite Button 3	Sammelanzeige,
"Button4Width"	VT_I4		Button 4 Width	Breite Button 4	Sammelanzeige,
"ButtonColor"	VT_I4		Button Color	Knopffarbe	Slider-Objekt,
"ButtonCommand"	VT_I4	OCX	ButtonCommand	ButtonCommand	CCAlgWinCtrl,
"Caption"	VT_BOOLEAN		Title	Titel	Druckauftrag/Skriptdiagnose, Bildfenster,
"Caption"	VT_BOOLEAN	OCX	Caption	Caption	XGaugeCtrl, PButtonCtrl, SliderCtrl,
"CaptionColor"	VT_UI4	OCX	CaptionColor	CaptionColor	XGaugeCtrl,
"CaptionFont"	VT_USEDEFINED	OCX	CaptionFont	CaptionFont	XGaugeCtrl,
"CaptionOffset"	VT_R8	OCX	CaptionOffset	CaptionOffset	XGaugeCtrl,
"CellCut"	VT_BOOLEAN	OCX	CellCut	CellCut	CCAlgWinCtrl,
"CenterColor"	VT_UI4	OCX	CenterColor	CenterColor	XGaugeCtrl,
"CenterScale"	VT_R4	OCX	CenterScale	CenterScale	XGaugeCtrl,
"CheckAlarmHigh"	VT_BOOLEAN		Monitoring AH	Überwachung AH	Balken,
"CheckAlarmLow"	VT_BOOLEAN		Monitoring AL	Überwachung AL	Balken,
"CheckLimitHigh4"	VT_BOOLEAN		Monitoring RH4	Überwachung RH4	Balken,

2.3 Funktionen des Grafiksystems (RT Professional)

"CheckLimitHigh5"	VT_BO-OL		Monitoring RH5	Überwachung RH5	Balken,
"CheckLimitLow4"	VT_BO-OL		Monitoring RL4	Überwachung RL4	Balken,
"CheckLimitLow5"	VT_BO-OL		Monitoring RL5	Überwachung RL5	Balken,
"CheckTolerance-High"	VT_BO-OL		Monitoring TH	Überwachung TH	Balken,
"CheckTolerance-Low"	VT_BO-OL		Monitoring TL	Überwachung TL	Balken,
"CheckWarningHigh"	VT_BO-OL		Monitoring WH	Überwachung WH	Balken,
"CheckWarningLow"	VT_BO-OL		Monitoring WL	Überwachung WL	Balken,
"ClearOnError"	VT_BO-OL		Clear on Invalid Input	Löschen bei Fehleingabe	E/A-Feld,
"ClearOnNew"	VT_BO-OL		Clear on New Input	Löschen bei Neueingabe	E/A-Feld,
"CloseButton"	VT_BO-OL		Can be Closed	Schließbar	Druckauftrag/Skriptdiagnose, Bildfenster,
"CollectValue"	VT_UI4		Group Value	Sammelwert	Sammelanzeige
"ColMove"	VT_BO-OL	OCX	ColMove	ColMove	CCAlgWinCtrl,
"ColorAlarmHigh"	VT_I4		Bar Color AH	Balkenfarbe AH	Balken,
"ColorAlarmLow"	VT_I4		Bar Color AL	Balkenfarbe AL	Balken,
"ColorBottom"	VT_I4		Low Limit Color	Farbe unterer Anschlag	Slider-Objekt,
"ColorChangeType"	VT_BO-OL		Change Color	Farbumschlag	Balken,
"ColorLimitHigh4"	VT_I4		Bar Color RH4	Balkenfarbe RH4	Balken,
"ColorLimitHigh5"	VT_I4		Bar Color RH5	Balkenfarbe RH5	Balken,
"ColorLimitLow4"	VT_I4		Bar Color RL4	Balkenfarbe RL4	Balken,
"ColorLimitLow5"	VT_I4		Bar Color RL5	Balkenfarbe RL5	Balken,
"ColorTolerance-High"	VT_I4		Bar Color TH	Balkenfarbe TH	Balken,
"ColorTolerance-Low"	VT_I4		Bar Color TL	Balkenfarbe TL	Balken,
"ColorTop"	VT_I4		High Limit Color	Farbe oberer Anschlag	Slider-Objekt,
"ColorWarningHigh"	VT_I4		Bar Color WH	Balkenfarbe WH	Balken,
"ColorWarningLow"	VT_I4		Bar Color WL	Balkenfarbe WL	Balken,
"ColTitle"	VT_BO-OL	OCX	ColTitle	ColTitle	CCAlgWinCtrl,
"ColWidth"	VT_BO-OL	OCX	ColWidth	ColWidth	CCAlgWinCtrl,
"CursorControl"	VT_BO-OL		Cursor Control	Cursorsteuerung	E/A-Feld, Textliste,

"CursorMode"	VT_BO-OL				
"Danger"	VT_R4	OCX	Danger	Danger	XGaugeCtrl,
"DangerColor"	VT_UI4	OCX	DangerColor	DangerColor	XGaugeCtrl,
"DataFormat"	VT_I4		Data Format	Datenformat	E/A-Feld,
"DataLanguage"	VT_I4		Configured Languages	Sprachen projiziert	Button, Statischer Text, Check-Box, Radio-Box,
"Delta"	VT_R4		Delta	Delta	XGaugeCtrl,
"DesignMode"					
"DigitNumber"					
"Direction"	VT_I4		Bar Alignment	Balkenrichtung	3D-Balken, Slider-Objekt, Balken,
"EditAtOnce"	VT_BO-OL		Immediate Input	Eingabe sofort	E/A-Feld, Textliste,
"EnableTime"					
"EnableWriting"					
"EndAngle"	VT_I4		End Angle	Endwinkel	Ellipsenbogen, Ellipsensegment, Kreisbogen, Kreissegment,
"Exponent"	VT_BO-OL		Exponent Display	Exponentendarstellung	Balken,
"ExtendedOperation"	VT_BO-OL		Extended Operation	Erweiterte Bedienung	Slider-Objekt,
"FillBackColor"					
"FillColor"	VT_I4		Fill Pattern Color	Füllmusterfarbe	Button, Ellipse, Ellipsensegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"Filling"	VT_BO-OL		Dynamic Filling	Dynamisches Füllen	Button, Ellipse, Ellipsensegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, Graphik-Objekt, Check-Box, Radio-Box, Rundbutton,

2.3 Funktionen des Grafiksystems (RT Professional)

"FillingIndex	VT_I4		Fill Level	Füllstand	Button, Ellipse, Ellipsensegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, Graphik-Objekt, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"FillStyle"	VT_UI4		Fill Pattern	Füllmuster	Button, Ellipse, Ellipsensegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"FillStyle2"	VT_UI4		Bar Pattern	Balkenmuster	Balken,
"FlashBackColor"	VT_BOOLEAN		Flashing Background Active	Blinken Hintergrund aktiv	Button, Ellipse, Ellipsensegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"FlashBorderColor"	VT_BOOLEAN		Flashing Line Active	Blinken Rahmen aktiv	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"FlashBorderColorEx"					
"FlashFlashPicture"	VT_BOOLEAN		Flashing Flash Picture Active	Blinken Blinkbild aktiv	Zustandsanzeige,
"FlashForeColor"	VT_BOOLEAN		Flashing Text Active	Blinken Schrift aktiv	Button, Statischer Text, E/A-Feld, Textliste, Check-Box, Radio-Box,
"FlashPicReferenced"	VT_BOOLEAN		Flash Picture Referenced	Blinksbild referenziert	Zustandsanzeige,

"FlashPicTransColor"	VT_I4		Flash Picture Transparent Color	Blinkbild Transparentfarbe	Zustandsanzeige,
"FlashPicture"	VT_BOOL		Flash Picture	Blinkbild	Zustandsanzeige,
"FlashPicUseTransColor"			Flash Picture Transparent Color On	Blinkbild Transparentfarbe Ein	Zustandsanzeige,
"FlashRate"	VT_I4		Flash Frequency	Blinkfrequenz	Sammelanzeige,
"FlashRateBackColor"	VT_I4		Background Flash Frequency	Blinkfrequenz Hintergrund	Button, Ellipse, Ellipsensegment, Kreis, Kreissegment, Polygon, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"FlashRateBorderColor"	VT_I4		Line Flash Frequency	Blinkfrequenz Rahmen	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, E/A-Feld, Balken, Graphik-Objekt, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"FlashRateFlashPic"	VT_I4		Flash Picture Flash Frequency	Blinkfrequenz Blinkbild	Zustandsanzeige
"FlashRateForeColor"	VT_I4		Text Flash Frequency	Blinkfrequenz Schrift	Button, Statischer Text, E/A-Feld, Textliste, Check-Box, Radio-Box,
"Font"	VT_USEDEFINED	OCX	Font	Font	DAClockCtrl, PButtonCtrl, SliderCtrl,
"FontBold"	VT_BOOLEAN		Bold	Fett	Button, Statischer Text, E/A-Feld, Balken, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"FontBold"	VT_BOOLEAN	OCX	FontBold	FontBold	PButtonCtrl,

2.3 Funktionen des Grafiksystems (RT Professional)

"FontItalic"	VT_BO-OL		Italic	Kursiv	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"FontItalic"	VT_BO-OL		FontItalic	FontItalic	PButtonCtrl,
"FontName"	VT_BSTR		Font	Zeichensatz	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"FontName"	VT_BSTR	OCX	FontName	FontName	PButtonCtrl,
"FontPosition"	VT_USERDEFINED	OCX	FontPosition	FontPosition	SliderCtrl
"FontSize"	VT_I4		Font Size	Zeichensatzgröße	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"FontSize"	VT_BSTR	OCX	FontSize	Font Size	PButtonCtrl,
"FontStrikeThru"	VT_BO-OL	OCX	FontStrikeThru	FontStrikeThru	PButtonCtrl,
"FontUnderline"	VT_BO-OL		Underline	Unterstrichen	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"FontUnderline"	VT_BO-OL	OCX	FontUnderline	FontUnderline	PButtonCtrl,
"ForceInput"					
"ForeColor"	VT_I4		Font Color	Schriftfarbe	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"ForeColor"	VT_UI4	OCX	ForeColor	ForeColor	DAClockCtrl, PButtonCtrl, SliderCtrl,
"ForeFlashColorOff"	VT_I4		Flashing Text Color Off	Schriftblinkfarbe Aus	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"ForeFlashColorOn"	VT_I4		Flashing Text Color On	Schriftblinkfarbe Ein	Button, Statischer Text, E/A-Feld, Sammelanzeige, Textliste, Check-Box, Radio-Box,
"FrameColor"	VT_UI4				

"FrameColorDown"	VT_UI4	OCX	FrameColor-Down	FrameColor-Down	PButtonCtrl,
"FrameColorUp"	VT_UI4	OCX	FrameColorUp	FrameColorUp	PButtonCtrl,
"FramePicture"	VT_USE RDEFI- NED	OCX	FramePicture	FramePicture	XGaugeCtrl,
"FrameScale"	VT_R4	OCX	FrameScale	FrameScale	XGaugeCtrl,
"FrameSize"		OCX	FrameSize	FrameSize	PButtonCtrl,
"FrameWidth"	VT_I4	OCX	FrameWidth	FrameWidth	PButtonCtrl,
"Freeze"	VT_BO- OL		Freeze	Freeze	WTVctrlCtrl,
"Grid"	VT_BO- OL				
"GridColor"	VT_I4				
"GridHeight"	VT_I4				
"GridLineHorz"	VT_BO- OL	OCX	GridLineHorz	GridLineHorz	CCAlgWinCtrl,
"GridLineVert"	VT_BO- OL	OCX	GridLineVert	GridLineVert	CCAlgWinCtrl,
"GridWidth"	VT_I4				
"Height"	VT_I4		Height	Höhe	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, DAClockCtrl, XGaugeCtrl, CCAlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"Height"	VT_I4		Window Height	Fensterhöhe	Druckauftrag/Skriptdiagnose, Bildfenster,
"HiddenInput"	VT_BO- OL		Hidden Input	Verdeckte Eingabe	E/A-Feld,
"Hotkey"	VT_UI4		Hot Key	Hotkey	Button,
"HourNeedleHeight"	VT_I4		HourNeedle- Height	HourNeedle- Height	DAClockCtrl,
"HourNeedleWidth"	VT_I4		HourNeedle- Width	HourNeedle- Width	DAClockCtrl,
"Hysteresis"	VT_BO- OL		Hysteresis	Hysteresese	Balken,
"HysteresisRange"	VT_R8		Hysteresis Range	Hystereseebe- reich	Balken,

"Index"	VT_I4		Index	Index	Polygon, Polygonzug, Zustandsanzeige, Check-Box, Radio-Box,
"InnerBevelOffset"	VT_I4	OCX	InnerBevelOffset	InnerBevelOffset	SliderCtrl,
"InnerBevelStyle"	VT_I4	OCX	InnerBevelStyle	InnerBevelStyle	SliderCtrl,
"InnerBevelWidth"	VT_I4	OCX	InnerBevelWidth	InnerBevelWidth	SliderCtrl,
"ItemBorderBackColor"	VT_I4		Dividing Background Color	Trennlinienhintergrundfarbe	Textliste,
"ItemBorderColor"	VT_I4		Dividing Line Color	Trennlinienfarbe	Textliste,
"ItemBorderStyle"	VT_UI4		Dividing Line Style	Trennlinienart	Textliste,

2.3.1.6 Liste der Objekteigenschaften (L-Z) (Graphics Designer) (RT Professional)

Übersicht

Hinweis

Bei vielen API-Funktionen des Graphics Designer müssen Sie den Parameter IpszPropName versorgen. Dabei muss der englische Name einer Objekteigenschaft angegeben werden.

Properties vom Typ VT_USERDEFINED sind nicht mit den Get/Set-Property-Funktionen zu bearbeiten.

Es darf kein VT_DISPATCH und andere Referenzen verwendet werden, lediglich die normalen Typen und max. ein VT_VARIANT für ein Array-Property einfacher Typen.

OLE-Automation Name	Datentyp		Property name	Attributname	Objektzugehörigkeit
"LanguageSwitch"	VT_BOOL		Language Switch	Sprachumschaltung	Textliste,
"LastChange"	VT_BSTR				

"Layer"	VT_I4		Layer	Ebene	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, Druckauftrag/Skriptdiagnose, Bildfenster, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, XGaugeCtrl, CCAlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"Layer00Checked"	VT_BOOL	BPC	Monitoring 0	Überwachung 0	3D-Balken,
"Layer01Checked"	VT_BOOL	BPC	Monitoring 1	Überwachung 1	3D-Balken,
"Layer02Checked"	VT_BOOL	BPC	Monitoring 2	Überwachung 2	3D-Balken,
"Layer03Checked"	VT_BOOL	BPC	Monitoring 3	Überwachung 3	3D-Balken,
"Layer04Checked"	VT_BOOL	BPC	Monitoring 4	Überwachung 4	3D-Balken,
"Layer05Checked"	VT_BOOL	BPC	Monitoring 5	Überwachung 5	3D-Balken,
"Layer06Checked"	VT_BOOL	BPC	Monitoring 6	Überwachung 6	3D-Balken,
"Layer07Checked"	VT_BOOL	BPC	Monitoring 7	Überwachung 7	3D-Balken,
"Layer08Checked"	VT_BOOL	BPC	Monitoring 8	Überwachung 8	3D-Balken,
"Layer09Checked"	VT_BOOL	BPC	Monitoring 9	Überwachung 9	3D-Balken,
"Layer10Checked"	VT_BOOL	BPC	Monitoring 10	Überwachung 10	3D-Balken,
"Layer00Color"	VT_I4	BPC	Bar Color 0	Balkenfarbe 0	3D-Balken,
"Layer01Color"	VT_I4	BPC	Bar Color 1	Balkenfarbe 1	3D-Balken,
"Layer02Color"	VT_I4	BPC	Bar Color 2	Balkenfarbe 2	3D-Balken,
"Layer03Color"	VT_I4	BPC	Bar Color 3	Balkenfarbe 3	3D-Balken,
"Layer04Color"	VT_I4	BPC	Bar Color 4	Balkenfarbe 4	3D-Balken,
"Layer05Color"	VT_I4	BPC	Bar Color 5	Balkenfarbe 5	3D-Balken,
"Layer06Color"	VT_I4	BPC	Bar Color 6	Balkenfarbe 6	3D-Balken,
"Layer07Color"	VT_I4	BPC	Bar Color 7	Balkenfarbe 7	3D-Balken,
"Layer08Color"	VT_I4	BPC	Bar Color 8	Balkenfarbe 8	3D-Balken,
"Layer09Color"	VT_I4	BPC	Bar Color 9	Balkenfarbe 9	3D-Balken,
"Layer10Color"	VT_I4	BPC	Bar Color 10	Balkenfarbe 10	3D-Balken,

2.3 Funktionen des Grafiksystems (RT Professional)

"Layer00Value"	VT_R8	BPC	Limit 0	Grenze 0	3D-Balken,
"Layer01Value"	VT_R8	BPC	Limit 1	Grenze 1	3D-Balken,
"Layer02Value"	VT_R8	BPC	Limit 2	Grenze 2	3D-Balken,
"Layer03Value"	VT_R8	BPC	Limit 3	Grenze 3	3D-Balken,
"Layer04Value"	VT_R8	BPC	Limit 4	Grenze 4	3D-Balken,
"Layer05Value"	VT_R8	BPC	Limit 5	Grenze 5	3D-Balken,
"Layer06Value"	VT_R8	BPC	Limit 6	Grenze 6	3D-Balken,
"Layer07Value"	VT_R8	BPC	Limit 7	Grenze 7	3D-Balken,
"Layer08Value"	VT_R8	BPC	Limit 8	Grenze 8	3D-Balken,
"Layer09Value"	VT_R8	BPC	Limit 9	Grenze 9	3D-Balken,
"Layer10Value"	VT_R8	BPC	Limit 10	Grenze 10	3D-Balken,
"Left"	VT_I4		Position X	Position X	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, Druckauftrag/Skriptdiagnose, Bildfenster, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, XGaugeCtrl, CCAlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"LeftComma"	VT_I4		Digits to the Left of the Decimal Point	Vorkommastellen	Balken,
"LightEffect"	VT_BOOL		Light Effect	Lichteffekt	3D-Balken,
"LimitHigh4"	VT_R8		Limit RH4	Obergrenze RH4	Balken,
"LimitHigh5"	VT_R8		Limit RH5	Obergrenze RH5	Balken,
"LimitLow4"	VT_R8		Limit RL4	Obergrenze RL4	Balken,
"LimitLow5"	VT_R8		Limit RL5	Obergrenze RL5	Balken,
"LimitMax"	VT_R8				
"LimitMin"	VT_R8				
" LineFont "	VT_BOOL	OCX	LineFont	LineFont	CCAlgWinCtrl,
"LineHeight"	VT_BOOL	OCX	LineHeight	LineHeight	CCAlgWinCtrl,
"LineTitle"	VT_BOOL	OCX	LineTitle	LineTitle	CCAlgWinCtrl,
"ListType"	VT_I4		List Type	Listenart	Textliste,
"Locale"					
"LockBackColor"	VT_I4		Lock Background Color	Sperre Hintergrundfarbe	Sammelanzeige,

"LockStatus"	VT_BOOL		Lock Display	Sperre Anzeige	Sammelanzeige,
"LockText"	VT_BOOL		Lock Display Text	Sperre Anzeigetext	Sammelanzeige,
"LockTextColor"	VT_I4		Lock Font Color	Sperre Schriftfarbe	Sammelanzeige,
"LongStrokes-Bold"	VT_BOOL		Long Axis Section	Lange Abschnitte	Balken,
"LongStrokesOnly"	VT_BOOL		Only Long Axis Section	Nur lange Abschnitte	Balken,
"LongStrokes-Size"	VT_UI4		Length of Axis Section	Länge Abschnitt	Balken,
"LongStrokes-TextEach"	VT_UI4		Label Each	Beschrifte jeden	Balken,
"Marker"	VT_BOOL		Limit Marker	Grenzwertmarkierer	Balken,
"Max"	VT_R8		Maximum Value	Maximalwert	3D-Balken, Slider-Objekt, Balken,
"MaximizeButton"	VT_BOOL		Can be Maximized	Maximierbar	Druckauftrag/Skriptdiagnose, Bildfenster,
"MCGUBackColorOff"	VT_I4		Went Out Unacknowledge-Background Color Off	Gegangen Unquittiert - Hintergrundfarbe Aus	Sammelanzeige,
"MCGUBackColorOn"	VT_I4		Went Out Unacknowledge-Background Color On	Gegangen Unquittiert - Hintergrundfarbe Ein	Sammelanzeige,
"MCGUBackFlash"	VT_BOOL		Went Out Unacknowledge - Background Flashing	Gegangen Unquittiert - Hintergrund blinken	Sammelanzeige,
"MCGUTextColorOff"	VT_I4		Went Out Unacknowledge-Text Color Off	Gekommen - Hintergrundfarbe Aus	Sammelanzeige,
"MCGUTextColorOn"	VT_I4		Went Out Unacknowledge-Text Color On	Gekommen - Hintergrundfarbe Ein	Sammelanzeige,
"MCGUTextFlash"	VT_BOOL		Went Out Unacknowledge-Text Flashing	Gegangen Unquittiert - Hintergrund blinken	Sammelanzeige,
"MCKOBackColorOff"	VT_I4		Came In - Background Color Off	Gekommen - Hintergrundfarbe Aus	Sammelanzeige,
"MCKOBackColorOn"	VT_I4		Came In - Background Color On	Gekommen - Hintergrundfarbe Ein	Sammelanzeige,
"MCKOBackFlash"	VT_BOOL		Came In - Background Flashing	Gekommen - Hintergrund blinken	Sammelanzeige,

2.3 Funktionen des Grafiksystems (RT Professional)

"MCKOTextColorOff"	VT_I4		Came In - Text Color Off	Gekommen - Schriftfarbe Aus	Sammelanzeige,
"MCKOTextColorOn"	VT_I4		Came In - Text Color On	Gekommen - Schriftfarbe Ein	Sammelanzeige,
"MCKOTextFlash"	VT_BOOL		Came In - Text Flashing	Gekommen - Schrift blinken	Sammelanzeige,
"MCKQBackColorOff"	VT_I4		Came In Acknowledge - Background Color Off	Gekommen Quittiert - Hintergrundfarbe Aus	Sammelanzeige,
"MCKQBackColorOn"	VT_I4		Came In Acknowledge - Background Color On	Gekommen Quittiert - Hintergrundfarbe Ein	Sammelanzeige,
"MCKQBackFlash"	VT_BOOL		Came In Acknowledge - Background Flashing	Gekommen Quittiert - Hintergrund blinken	Sammelanzeige,
"MCKQTextColorOff"	VT_I4		Came In Acknowledge - Text Color Off	Gekommen Quittiert - Schriftfarbe Aus	Sammelanzeige,
"MCKQTextColorOn"	VT_I4		Came In Acknowledge - Text Color On	Gekommen Quittiert - Schriftfarbe Ein	Sammelanzeige,
"MCKQTextFlash"	VT_BOOL		Came In Acknowledge - Text Flashing	Gekommen Quittiert - Schrift blinken	Sammelanzeige,
"MCText"	VT_BOOL		Display Text	Anzeigetext	Sammelanzeige,
"MessageClass"	VT_I4		Message Type	Meldeart	Sammelanzeige,
"Min"	VT_R8		Minimum Value	Minimalwert	Sammelanzeige,
"MinuteNeedleHeight"	VT_I4	OCX	MinuteNeedleHeight	MinuteNeedleHeight	Sammelanzeige,
"MinuteNeedleWidth"	VT_I4	OCX	MinuteNeedleWidth	MinuteNeedleWidth	Sammelanzeige,
"MsgFilter"	VT_BSTR	OCX		MsgFilter	
"MsgFilter2"	VT_BSTR	OCX		MsgFilter2	
"Moveable"	VT_BOOL		Moveable	Verschiebbar	Druckauftrag/Skriptdiagnose, Bildfenster,
"NeedleColor"	VT_UI4	OCX	NeedleColor	NeedleColor	XGaugeCtrl,
"NormalColor"	VT_UI4	OCX	NormalColor	NormalColor	XGaugeCtrl,
"NumberLines"	VT_I4		Number of visible Lines	Anzahl sichtbarer Zeilen	Textliste,

"ObjectName"	VT_BSTR		Object Name	Objektname	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, Druckauftrag/Skriptdiagnose, E/A-Feld, Balken, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, DAClockCtrl, XGaugeCtrl, CCAlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"OffsetLeft"	VT_I4		Picture Offset X	Bildoffset X	Bildfenster,
"OffsetTop"	VT_I4		Picture Offset Y	Bildoffset Y	Bildfenster,
"OnTop"	VT_BOOL		Foreground	Vordergrund	Druckauftrag/Skriptdiagnose, Bildfenster,
"Operation"	VT_BOOL		Operator-Control Enable	Bedienfreigabe	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, DAClockCtrl, XGaugeCtrl, CCAlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"OperationMessage"	VT_BOOL		Operator Input Message	Bedienmeldung	E/A-Feld, Textliste, Check-Box, Radio-Box, Slider-Objekt,
"OperationReport"	VT_BOOL		Operator Actions Report	Bedienprotokoll	E/A-Feld, Textliste, Slider-Objekt,
"OperationSteps"			Operation Steps	Bedienschritte	Slider-Objekt,
"Orientation"	VT_BOOL		Text Orientation	Schreibrichtung	Statischer Text, E/A-Feld, Textliste, Check-Box, Radio-Box,
"OuterBevelOffset"	VT_I4	OCX	OuterBevelOffset	OuterBevelOffset	SliderCtrl,
"OuterBevelStyle"	VT_I4	OCX	OuterBevelStyle	OuterBevelStyle	SliderCtrl,

2.3 Funktionen des Grafiksystems (RT Professional)

"OuterBevel-Width"	VT_I4	OCX	OuterBevel-Width	OuterBevel-Width	SliderCtrl,
"Outline"	VT_BOOL	OCX	Outline	Outline	PButtonCtrl,
"OutputFormat"	VT_BSTR		Output Format	Ausgabeformat	E/A-Feld,
"OutputValue"	VT_R8		Output Value	Ausgabewert	E/A-Feld, Textliste,
"PasswordLevel"	VT_UI4		User Level	Passwort	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt,
"PicDeactReferenced"	VT_BOOL		Picture Deact. Referenced	Bild Deakt. Referenziert	Rundbutton,
"PicDeactTransparent"	VT_I4		Picture Deact. Transparent Color	Bild Deakt. Transparentfarbe	Rundbutton,
"PicDeactUseTransColor"	VT_BOOL		Picture Deact. Transparent Color On	Bild Deakt. Transparentfarbe Ein	Rundbutton,
"PicDownReferenced"	VT_BOOL		Picture On Referenced	Bild Ein Referenziert	Rundbutton,
"PicDownTransparent"	VT_I4		Picture On Transparent Color	Bild Ein Transparentfarbe	Rundbutton,
"PicDownUseTransColor"	VT_BOOL		Picture On Transparent Color On	Bild Ein Transparentfarbe Ein	Rundbutton,
"PicReferenced"	VT_BOOL		Picture Referenced	Bild Referenziert	Graphic-Objekt,
"PicTransColor"	VT_I4		Picture Transparent Color	Bild Transparentfarbe	Graphic-Objekt,
"Picture"	VT_USER_DEFINED	OCX	Picture	Picture	DAClockCtrl,
"PictureBack"	VT_USER_DEFINED	OCX	PictureBack	PictureBack	SliderCtrl,
"PictureDeactivated"	VT_BSTR		Picture Status Deactivated	Bild Zustand Deaktiviert	Rundbutton,
"PictureDown"	VT_BSTR		Picture Status On	Bild Zustand Ein	Button, Rundbutton,
"PictureName"	VT_BSTR		Picture Name	Bildname	Bildfenster, Graphik-Objekt
"PictureSelected"	VT_USER_DEFINED	OCX	PictureSelected	PictureSelected	PButtonCtrl,

2.3 Funktionen des Grafiksystems (RT Professional)

"PictureUnselected"	VT_USER DEFINED	OCX	Picture Unselected	PictureUnselected	PButtonCtrl,
"PictureObjectPath"					
"PictureThumb"	VT_USER DEFINED		PictureThumb	PictureThumb	SliderCtrl,
"PictureUp"	VT_BSTR		Picture Status Off	Bild Zustand Aus	Button, Rundbutton,
"PicUpReferenced"	VT_BOOL		Picture Off Referenced	Bild Aus Referenziert	Rundbutton,
"PicUpTransparent"	VT_I4		Picture Off Transparent Color	Bild Aus Transparentfarbe	Rundbutton,
"PicUpUseTransColor"	VT_BOOL		Picture Off Transparent Color On	Bild Aus Transparentfarbe Ein	Rundbutton,
"PicUseTransColor"	VT_BOOL		Picture Transparent Color On	Transparentfarbe Ein	Graphic-Objekt,
"PointCount"	VT_I4		Number of Corners	Anzahl Eckpunkte	Polygon, Polygonzug,
"Position"	VT_I4	OCX	Positions	Position	SliderCtrl,
"PredefinedAngles"	VT_I4	BPC	Angle Settings	Winkелеinstellungen	3D-Balken
"Pressed"	VT_BOOL		Pressed	gedrückt	Rundbutton,
"Process"	VT_R8		Process Driver Connection	Prozessanschluss	3D-Balken, Slider-Objekt, Balken,
"Process"	VT_R8		Selected Boxes	Selektierte Felder	Check-Box, Radio-Box
"ProfileName"					
"ProjectName"					
"ProjectPath"	VT_BSTR	OCX	ProjectPath	ProjectPath	CCAlgWinCtrl,
"Radius"	VT_I4		Radius	Radius	Kreis, Kreisbogen, Kreissegment, Rundbutton,
"RadiusHeight"	VT_I4		Radius Y	Radius Y	Ellipse, Ellipsenbogen, Ellipsesegment,
"RadiusWidth"	VT_I4		Radius X	Radius X	Ellipse, Ellipsenbogen, Ellipsesegment,
"RangeMax"	VT_I4	OCX	RangeMax	RangeMax	SliderCtrl,
"RangeMin"	VT_I4	OCX	RangeMin	RangeMin	SliderCtrl,
"ReferenceMoveLeft"					
"ReferenceMoveTop"					
"ReferenceRotationLeft"	VT_I4		Rotation Reference X	Rotationsreferenz X	Linie, Polygon, Polygonzug,
"ReferenceRotationTop"	VT_I4		Rotation Reference Y	Rotationsreferenz Y	Linie, Polygon, Polygonzug,

2.3 Funktionen des Grafiksystems (RT Professional)

"RefreshTimer-Period"	VT_I4	OCX	RefreshTimer-Period	RefreshTimer-Period	WTVctrlCtrl,
"Relevant"	VT_BOOL		Group Relevant	Sammelrelevant	Sammelanzeige,
"RightComma"	VT_I4		Digits to the Right of the Decimal Point	Nachkommastellen	Balken,
"RotationAngle"	VT_I4		Rotation Angle	Rotationswinkel	Linie, Polygon, Polygonzug,
"RoundCorner-Height"	VT_I4		Corner Radius Y	Eckradius Y	Rundrechteck,
"RoundCorner-Width"	VT_I4		Corner Radius X	Eckradius X	Rundrechteck,
"SameSize"	VT_BOOL		Same Size	Gleiche Größe	Sammelanzeige,
"ScaleColor"	VT_I4		Scale Color	Skalenfarbe	Balken,
"ScaleTicks"	VT_I4		Scale Marks	Skaleneinteilung	Balken,
"Scaling"	VT_BOOL		Scale	Skala	Balken,
"ScalingType"	VT_I4		Bar Scalling	Balkenskalierung	Balken,
"ScrollBars"	VT_BOOL		Scroll Bar	Rollbalken	Bildfenster,
"SecondNeedle-Height"	VT_I4	OCX	SecondNeedleHeight	SecondNeedleHeight	DAClockCtrl,
"SecondNeedle-Width"	VT_I4	OCX	SecondNeedleWidth	SecondNeedleWidth	DAClockCtrl,
"SelBGColor"	VT_I4		Selection Background Color	Selektions-Hintergrundfarbe	Textliste,
"SelectionMode"	VT_I2	OCX	SelectionMode	SelectionMode	CCAIgWinCtrl,
"SelTextColor"	VT_I4		Selection Font Color	Selektions-Schrifffarbe	Textliste,
"ServerName"	VT_BSTR		Server Name	Servername	DAClockCtrl, XGaugeCtrl, CCAIgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"Sign"					
"Significant-Mask"	VT_UI4		Bit Pattern Group Display	Bitmuster Sammelanzeige	Sammelanzeige,
"Sizeable"	VT_BOOL		Sizeable	Größe veränderbar	Druckauftrag/Skriptdiagnose, Bildfenster
"ShowBar"	VT_BOOL	OCX	ShowBar	ShowBar	SliderCtrl,
"ShowDanger"	VT_BOOL	OCX	ShowDanger	ShowDanger	XGaugeCtrl,
"ShowDecimal-Point"	VT_BOOL	OCX	ShowDecimal-Point	ShowDecimal-Point	XGaugeCtrl,
"ShowNormal"	VT_BOOL	OCX	ShowNormal	ShowNormal	XGaugeCtrl,
"ShowPeak"	VT_BOOL	OCX	ShowPeak	ShowPeak	XGaugeCtrl,
"ShowPosition"	VT_BOOL	OCX	ShowPosition	ShowPosition	SliderCtrl,
"ShowThumb"	VT_BOOL	OCX	ShowThumb	ShowThumb	SliderCtrl,

"SmallWarning"	VT_I4			SmallWarning	Slider-Objekt,
"StartAngle"	VT_I4		Start Angle	Anfangswinkel	Ellipsenbogen, Ellipsensegment, Kreisbogen, Kreissegment,
"TabOrderAlpha"					
"TabOrderSwitch"					
"Template"	VT_BSTR		Template	Vorlage	Druckauftrag/Skriptdiagnose,
"TemplateName"	VT_BSTR	OCX	TemplateName	TemplateName	CCAlgWinCtrl,
"Text"	VT_BSTR		Text	Text	Button, Statischer Text, Check-Box, Radio-Box,
"ThumbBackColor"	VT_UI4	OCX	ThumbBackColor	ThumbBackColor	SliderCtrl,
"TicColor"	VT_UI4	OCX	TicColor	TicColor	XGaugeCtrl,
"TicFont"	VT_USER_DEFINED	OCX	TicFont	TicFont	XGaugeCtrl,
"Ticks"	VT_BOOL	OCX	Ticks	Ticks	DAClockCtrl,
"TicksColor"	VT_UI4	OCX	TicksColor	TicksColor	DAClockCtrl,
"TickStyle"	VT_I4	OCX	TickStyle	TickStyle	SliderCtrl,
"TicOffset"	VT_R8	OCX	TicOffset	TicOffset	XGaugeCtrl,
"TicTextColor"	VT_UI4	OCX	TicTextColor	TicTextColor	XGaugeCtrl,
"TicTextOffset"	VT_R8	OCX	TicTextOffset	TicTextOffset	XGaugeCtrl,
"TicWidth"	VT_R8	OCX	TicWidth	TicWidth	XGaugeCtrl,
"TitleCut"	VT_BOOL	OCX	TitleCut	TitleCut	CCAlgWinCtrl,
"Toggle"	VT_BOOL		Latch Down	Einrasten	Rundbutton,
"ToleranceHigh"	VT_R8		Limit TH	Obergrenze TH	Balken,
"ToleranceLow"	VT_R8		Limit TL	Untergrenze TL	Balken,
"ToolTipText"	VT_BSTR		Tooltip Text	Tooltiptext	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, DAClockCtrl, XGaugeCtrl, CCAlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,

2.3 Funktionen des Grafiksystems (RT Professional)

"Top"	VT_I4		Position Y	Position Y	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, Druckauftrag/Skriptdiagnose, Bildfenster, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, DAClockCtrl, XGaugeCtrl, CCAIlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"Trend"	VT_BOOL		Trend	Trend	Balken,
"TrendColor"	VT_I4		Trend Color	Trendfarbe	Balken,
"TypeAlarmHigh"	VT_BOOL		Type AH	Art AH	Balken,
"TypeAlarmLow"	VT_BOOL		Type AL	Art AL	Balken,
"TypeLimitHigh4"	VT_BOOL		Type RH4	Art RH4	Balken,
"TypeLimitHigh5"	VT_BOOL		Type RH5	Art RH5	Balken,
"TypeLimitLow4"	VT_BOOL		Type RL4	Art RL4	Balken,
"TypeLimitLow5"	VT_BOOL		Type RL5	Art RL5	Balken,
"TypeToleranceHigh"	VT_BOOL		Type TH	Art TH	Balken,
"TypeToleranceLow"	VT_BOOL		Type TL	Art TL	Balken,
"TypeWarningHigh"	VT_BOOL		Type WH	Art WH	Balken,
"TypeWarningLow"	VT_UI4		Type WL	Art WL	Balken,
"UnitColor"	VT_USER_DEFINED	OCX	UnitColor	UnitColor	XGaugeCtrl,
"UnitFont"	VT_BSTR	OCX	UnitFont	UnitFont	XGaugeCtrl,
"UnitText"	VT_BSTR	OCX	UnitText	UnitText	XGaugeCtrl,
"UnitOffset"	VT_R8	OCX	UnitOffset	UnitOffset	XGaugeCtrl,
"UnselBGColor"	VT_I4		List Background Color	Auswahl-Hintergrundfarbe	Textliste,
"UnselTextColor"	VT_I4		List Font Color	Auswahl-Schriftfarbe	Textliste,
"UpdateCycle"	VT_I4		Update Cycle	Aktualisierungszeitraster	Bildfenster,
"UseRefreshTimer"	VT_BOOL	OCX	UseRefreshTimer	UpdateCycle	WTVctrlCtrl,
"UserValue1"	VT_UI4		User Value 1	Benutzerwert 1	Sammelanzeige,
"UserValue2"	VT_UI4		User Value 2	Benutzerwert 2	Sammelanzeige,
"UserValue3"	VT_UI4		User Value 3	Benutzerwert 3	Sammelanzeige,

"UserValue4"	VT_UI4		User Value 4	Benutzerwert 4	Sammelanzeige,
"Value"	VT_R4	OCX	Value	Value	XGaugeCtrl,
"ValueMax"	VT_R4	OCX	ValueMax	ValueMax	XGaugeCtrl,
"ValueMin"	VT_R4	OCX	ValueMin	ValueMin	XGaugeCtrl,
"VideoEnabled"	VT_BOOL	OCX	VideoEnabled	VideoEnabled	WTVctrlCtrl,
"VideoSource"	VT_I2	OCX	VideoSource	VideoSource	WTVctrlCtrl,
"Visible"	VT_BOOL		Display	Anzeige	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, Druckauftrag/Skriptdiagnose, Bildfenster, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, DAClockCtrl, XGaugeCtrl, CCAIlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"Warning"	VT_R4	OCX	Warning	Warning	XGaugeCtrl,
"WarningColor"	VT_UI4	OCX	WarningColor	WarningColor	XGaugeCtrl,
"WarningHigh"	VT_R8		Limit WH	Obergrenze WH	Balken,
"WarningLow"	VT_R8		Limit WL	Untergrenze WL	Balken,
"Width"	VT_I4		Width	Breite	Button, Ellipse, Ellipsenbogen, Ellipsensegment, Kreis, Kreisbogen, Kreissegment, Linie, Polygon, Polygonzug, Rechteck, Rundrechteck, Statischer Text, 3D-Balken, Druckauftrag/Skriptdiagnose, Bildfenster, E/A-Feld, Balken, Graphik-Objekt, Sammelanzeige, Textliste, Zustandsanzeige, Check-Box, Radio-Box, Rundbutton, Slider-Objekt, DAClockCtrl, XGaugeCtrl, CCAIlgWinCtrl, PButtonCtrl, SliderCtrl, WTVctrlCtrl,
"WindowBorder"	VT_BOOL		Border	Rahmen	Druckauftrag/Skriptdiagnose, Bildfenster,
"WindowsStyle"	VT_BOOL		Windows Style	Windows-Stil	Button, Slider-Objekt,
"WindowType"	VT_I2	OCX	WindowType	WindowType	CCAIGWinCtrl,

"WinTVInformation"	VT_BSTR	OCX		WinTVInformation	WTVctrlCtrl,
"WithAxes"	VT_BOOL	OCX	WithAxes	WithAxes	SliderCtrl,
"WithLabels"	VT_BOOL	OCX	WithLabels	WithLabels	SliderCtrl,
"ZeroLeading"					
"ZeroPoint"	VT_UI4		Zero Point	Nullpunkt	Balken,
"ZeroPointValue"	VT_R8		Zero Point Value	Nullpunktwert	3D-Balken, Balken,
Zoom	VT_I4		Scaling Factor	Skalierungsfaktor	Bildfenster

2.3.1.7 API-Aufrufe in OCXen (RT Professional)

Übersicht

API wurde für Anwendungen in eigenen Applikationen entwickelt. Bei API-Aufrufen in OCXen, die in Graphics-Bilder eingebunden werden, kann es dagegen zu Problemen bis hin zu Deadlocks kommen.

Wenn unbedingt ein OCX in einem Graphics-Bild gebraucht wird, prüfen Sie bitte auf jeden Fall, ob sich die Aufgabe auch mit IndustrialX lösen läßt. Das ist für solche Zwecke konzipiert.

Für den Fall, dass sich die Aufgabe nur mit API lösen läßt, hier einige Hinweise:

Besonders betroffen sind dabei Runtime-API-Funktionen von Alarmlogging und Taglogging.

Eine Ursache ist, daß API in Runtime per Window-Message Daten von WinCC-Programmen anfordert und auf die Antwort wartet. Wenn das OCX im Kontext ausgerechnet eines solchen WinCC-Programm läuft, so kann dieses die Anforderung nicht bearbeiten, weil es ja im API-Aufruf hängt.

Abhilfe:

In diesem Fall genügt es oft, den API-Aufruf in einen anderen Thread zu legen.

Diese Abhilfe sollte jedoch nur benutzt werden, wenn es wirklich ein Problem gibt. Manche andere Aufrufe haben nämlich Probleme, wenn sie nicht im Hauptthread laufen.

2.3.2 Strukturen (RT Professional)

2.3.2.1 LINKINFO (RT Professional)

Deklaration

```
typedef struct {
    LINKTYPE    LinkType;
    DWORD       dwCycle;
    TCHAR       szLinkName[256];
}
LINKINFO;
```

Members

LinkType

Für den Typ LINKTYPE sollten die in der Datei "trigger.h" definierten Werte für enum LinkType verwendet werden.

enum LinkType	aktuell definierter Wert	Verknüpfung
BUBRT_LT_NOLINK	0 (definierter Startwert)	Keine Verknüpfung
BUBRT_LT_VARIABLE_DIRECT	1 (folgend)	direkte Variable
BUBRT_LT_VARIABLE_INDIRECT	2 (folgend)	indirekte Variable
BUBRT_LT_ACTION	3 (folgend)	C-Aktion
BUBRT_LT_ACTION_WIZARD	4 (folgend)	Dynamic Dialog
BUBRT_LT_DIRECT_CONNECTION	5 (folgend)	Direktverbindung
BUBRT_LT_ACTION_WIZARD_INPROC	6 (folgend)	Dynamic Dialog

Bei Verwendung in PDLRTSetLink dürfen nur BUBRT_LT_VARIABLE_DIRECT und BUBRT_LT_VARIABLE_INDIRECT verwendet werden. Bei PDLRTGetLink können alle Verknüpfungstypen zurückgeliefert werden.

Hinweis

Bei zukünftigen Erweiterungen der enum LinkType Definition in trigger.h müssen neue Konstanten immer hinten angefügt werden, da sich sonst die oben aufgelistete Wertigkeit ändert. Daher sollte bei Enumerationen immer die Definition und nicht der Wert verwendet werden.

dwCycle

Zykluszeit der Aktualisierung. dwCycle kennzeichnet die Ordnung innerhalb der Liste der Aktualisierungszyklen.

"Bei Änderung"	Index: 0
"250ms"	Index: 1

:	:
"Anwenderzyklus 5"	Index: 15
"Fensterzyklus"	Index: 235
"Bildzyklus"	Index: 255

szLinkName

Name der Variablenverbindung

Benötigte Dateien

- pdlrtapi.h
- trigger.h

API - Funktionen

PDLRTGetLink (Seite 588)	Verknüpfung Objekteigenschaft-Variable abfragen
PDLRTSetLink (Seite 590)	Verknüpfung Objekteigenschaft-Variable festlegen

Siehe auch

- PDLRTGetLink (Seite 588)
- PDLRTSetLink (Seite 590)

2.3.2.2 MULTILINK (RT Professional)

Deklaration

```
typedef struct {  
    DWORD wArraySize;  
    LPMULTILINKINFO pLinkArray;  
}  
MULTILINK;
```

Members

wArraySize

Anzahl der Verknüpfungen (Anzahl der Strukturen vom Typ MULTILINKINFO).

pLinkArray

Zeiger auf die erste von wArraySize Strukturen vom Typ MULTILINKINFO (Seite 553) mit den Informationen zur Verknüpfung einer Objekteigenschaft und einer Variablen.

Benötigte Dateien

pdLrtapi.h

API - Funktionen

PDLRTSetMultiLink (Seite 592)	Verknüpfung Objekteigenschaft-Variable festlegen (mehrere Variablen)
-------------------------------	--

Siehe auch

PDLRTSetMultiLink (Seite 592)

MULTILINKINFO (Seite 553)

2.3.2.3 MULTILINKINFO (RT Professional)**Deklaration**

```
typedef struct {
    char*      pszObjectName,
    char*      pszPropertyName,
    LINKTYPE   LinkType;
    DWORD      dwCycle;
    char*      pszLinkName;
}
MULTILINKINFO;
```

Members**IpszObjectName**

Zeiger auf den Namen des Objektes.

IpszPropertyName

Zeiger auf den Namen der der zu verknüpfenden Objekteigenschaft.

LinkType

Kennzeichnet die Art der Verbindung.

LT_VARIABLE_DIRECT	Objekteigenschaft und Variable direkt verbinden
LT_VARIABLE_INDIRECT	Objekteigenschaft und Variable indirekt verbinden

dwCycle

Zykluszeit der Aktualisierung. dwCycle kennzeichnet die Ordnung innerhalb der Liste der Aktualisierungszyklen.

"Bei Änderung"	Index: 0
"250ms"	Index: 1
:	:
"Anwenderzyklus 5"	Index: 15

pszLinkName

Zeiger auf Name der Verbindung

Bemerkungen

Die Struktur MULTILINKINFO ist Teil der Struktur MULTILINK (Seite 552).

Benötigte Dateien

pdlrtapi.h

Siehe auch

MULTILINK (Seite 552)

2.3.2.4 FOCUSINFO (RT Professional)

Beschreibung

In der Struktur wird hinterlegt welches Objekt eines Bildes den Runtimecursor besitzt.

Deklaration

```
typedef struct {  
    WCHAR    szPicture[256];  
    WCHAR    szObject[256];  
}  
FOCUSINFO;
```

Members

szPicture

Bildname

szObject

Objektname

API - Funktionen

PDLRTGetFocus (Seite 572)	Eingabefocus abfragen
---------------------------	-----------------------

Siehe auch

PDLRTGetFocus (Seite 572)

2.3.3 Allgemeine Funktionen (RT Professional)**2.3.3.1 PDLRTClosePicture (RT Professional)****Verwendung**

Schließt ein Bild im Runtime-Modus.

Deklaration

```
BOOL PDLRTClosePicture(  
    ADRMODE          adrMode,  
    LPCSTR           lpszPictureName,  
    LPCSTR           lpszWName,  
    PDLRT_CALLBACK  pfn,  
    LPVOID           pvUser,  
    PCMN_ERROR       pError );
```

Parameter**adrMode**

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lpszPictureName

Zeiger auf den projektierten Namen des Bildes, ohne Pfad und ohne Extension.

Groß- und Kleinbuchstaben werden unterschieden.

lpszWName

Name des Fensterobjekts, das geschlossen wird.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Bild geschlossen.

FALSE

Fehler.

Benötigte Dateien

pdlrtapi.h

pdlrt_s.lib

pdlrtapi.dll

Verwandte Funktionen

PDLRT_CALLBACK (Seite 568)	Call-back-Funktion der Runtime-Funktionen
----------------------------	---

Siehe auch

PDLRT_CALLBACK (Seite 568)

Übersicht über die Funktionen (Seite 522)

2.3.3.2 PDLRTDisableClosePicture (RT Professional)

Verwendung

Sperrt ein Bild im Runtime-Modus gegen Schließen. Der Mehrfachaufruf ist möglich. Das Bild kann erst wieder geschlossen werden, wenn ebenso viele Aufrufe der Funktion PDLRTEnableClosePicture gemacht wurden.

Deklaration

```
BOOL PDLRTDisableClosePicture(  
    ADRMODE          adrMode,  
    LPCSTR           lpszPictureName,  
    PDLRT_CALLBACK   pfn,  
    LPVOID           pvUser,  
    PCMN_ERROR       pError );
```

Parameter

adrMode

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lpszPictureName

Zeiger auf den projektierten Namen des Bildes, ohne Pfad und ohne Extension.

Groß- und Kleinbuchstaben werden unterschieden.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Funktion erfolgreich beendet.

FALSE

Fehler.

Benötigte Dateien

pdlrtapi.h

pdlrt_s.lib

pdlrtapi.dll

Verwandte Funktionen

PDLRT_CALLBACK (Seite 568)	Call-back-Funktion der Runtime-Funktionen
----------------------------	---

Siehe auch

PDLRT_CALLBACK (Seite 568)

Übersicht über die Funktionen (Seite 522)

2.3.3.3 PDLRTEnableClosePicture (RT Professional)

Verwendung

Gibt ein Bild im Runtime-Modus zum Schließen frei.

Das Bild kann jedoch erst geschlossen werden, wenn ebenso viele Aufrufe von PDLRTEnableClosePicture gemacht wurden, wie vorher von der Funktion PDLRTDisableClosePicture.

Ein vorher erfolgter PDLRTCclosePicture wird vorgemerkt und dann nach dem letzten PDLRTEnableClosePicture nachgeholt.

Deklaration

```
BOOL PDLRTEnableClosePicture (
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    PDLRT_CALLBACK   pfn,
    LPVOID           pvUser,
    PCMN_ERROR       pError );
```

Parameter

adrMode

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lpszPictureName

Zeiger auf den projektierten Namen des Bildes, ohne Pfad und ohne Extension.

Groß- und Kleinbuchstaben werden unterschieden.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Funktion erfolgreich beendet.

FALSE

Fehler.

Benötigte Dateien

pdLrtapi.h

pdLrt_s.lib

pdLrtapi.dll

Verwandte Funktionen

PDLRT_CALLBACK (Seite 568)	Call-back-Funktion der Runtime-Funktionen
----------------------------	---

Siehe auch

- PDLRT_CALLBACK (Seite 568)
- Übersicht über die Funktionen (Seite 522)

2.3.3.4 PDLRTGotoPicture (RT Professional)

Verwendung

Lädt das angegebene Bild.

Deklaration

```
BOOL PDLRTGotoPicture (  
    PDLRTGotoPict  nextPict,  
    PCMN_ERROR     pError );
```

Parameter

nextPict

Über diesen Parameter wird die Funktion dieser Methode gesteuert. Es sind folgende Werte möglich:

PDLRTPictureHome	Lädt das Startbild.
PDLRTPictureNext	Lädt das nächste Bild im Bildspeicher.
PDLRTPicturePrev	Lädt das vorherige Bild im Bildspeicher.
PDLRTPictureStored	Lädt das gemerkte Bild im Bildspeicher.
PDLRTStorePicture	Merkt sich das aktuelle Bild im Bildspeicher.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Bild geladen.

FALSE

Fehler.

Fehlermeldungen

PDLRT_NO_PIC	Bild nicht angewählt
--------------	----------------------

Benötigte Dateien

pdlrtapi.h
 pdlrt_s.lib
 pdlrtapi.dll

Verwandte Funktionen

PDLRTPictureNavigation (Seite 565)	Bildnavigation ein- und ausschalten
------------------------------------	-------------------------------------

Siehe auch

PDLRTPictureNavigation (Seite 565)
 Übersicht über die Funktionen (Seite 522)

2.3.3.5 PDLRTInquireFreeArea (RT Professional)**Verwendung**

Ermittelt, ob ein vorgegebenes Rechteck auf dem Bildschirm platziert werden kann, ohne einen "geschützten Bereich" zu überdecken. "Geschützte Bereiche" sind zum Beispiel andere im Vordergrund liegende Bilder.

Deklaration

```

BOOL PDLRTInquireFreeArea(
    LPRECT          lpScreenRect,
    DWORD           dwModus,
    PDLRT_CALLBACK  pfn,
    LPVOID          pvUser,
    PCMN_ERROR      pError );

```

Parameter

lpScreenRect

Zeiger auf eine Windows-spezifische Struktur vom Typ RECT.

dwModus

Inquiremodus: Es werden mehrere Inquiremöglichkeiten angeboten, die beliebig verodert werden können:

PDLRT_IQ_ONLY	(Wert: 0x01)	Nur Nachfrage mit dem Ergebnis "passt/passt nicht"
PDLRT_IQ_MODY_POSX	(Wert: 0x02)	Kann das vorgegebene Rechteck nicht platziert werden, so darf die x-Position des Rechtecks so modifiziert werden, dass es platziert werden kann.
PDLRT_IQ_MODY_POSY	(Wert: 0x04)	Kann das vorgegebene Rechteck nicht platziert werden, so darf die y-Position des Rechtecks so modifiziert werden, dass es platziert werden kann.
PDLRT_IQ_MODY_HEIGHT	(Wert: 0x08)	Kann das vorgegebene Rechteck nicht platziert werden, so darf die Höhe des Rechtecks so modifiziert werden, dass es platziert werden kann.
PDLRT_IQ_MODY_WIDTH	(Wert: 0x10)	Kann das vorgegebene Rechteck nicht platziert werden, so darf die Breite des Rechtecks so modifiziert werden, dass es platziert werden kann.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Fensterbereich ist frei.

FALSE

Fehler.

Benötigte Files

pdlrtapi.h
 pdlrt_s.lib
 pdlrtapi.dll

Verwandte Funktionen

PDLRT_CALLBACK (Seite 568)	Callback-Funktion der Runtime-Funktionen
----------------------------	--

Siehe auch

PDLRT_CALLBACK (Seite 568)
 Übersicht über die Funktionen (Seite 522)

2.3.3.6 PDLRTOpenPicture (RT Professional)**Verwendung**

Die Funktion führt einen Grundbildwechsel im Runtime durch.

Deklaration

```

BOOL PDLRTOpenPicture(
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    LPCSTR           lpszWName,
    LPCSTR           lpszPictureFileName,
    DWORD            dwWinStyle,
    LONG             lxPos,
    LONG             lyPos,
    LONG             lWidth,
    LONG             lHeight,
    PDLRT_CALLBACK   pfn,
    LPVOID           pvUser,
    PCMN_ERROR       pError );

```

Parameter**adrMode**

Zeiger auf den projektierten Namen des Bildes, ohne Pfad und ohne Extension.

Groß- und Kleinbuchstaben werden unterschieden.

lpzWName

Dieser Parameter ist zur Zeit mit dem Bildnamen zu versorgen.

lpzPictureFileName

Dieser Parameter ist zur Zeit mit dem Bildnamen zu versorgen.

dwWinStyle

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lxPos

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lyPos

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lWidth

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lHeigth

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Grundbildwechsel durchgeführt.

FALSE

Fehler.

Fehlermeldungen

PDLRT_PICTURE_ALREADY_OPEN	Bild ist bereits geöffnet
PDLRT_NO_PIC	Bild nicht angewählt
PDLRT_BAD_OLE_CONVERSION	Fehler bei Wandlung über OLE Automation

Benötigte Dateien

pdLrtapi.h

pdLrt_s.lib

pdLrtapi.dll

Verwandte Funktionen

PDLRT_CALLBACK (Seite 568)	Callback-Funktion der Runtime-Funktionen
----------------------------	--

Siehe auch

PDLRT_CALLBACK (Seite 568)

Übersicht über die Funktionen (Seite 522)

2.3.3.7 PDLRTPictureNavigation (RT Professional)**Verwendung**

Schaltet die Bildnavigation ein und aus. Im ausgeschalteten Zustand werden beim Bildwechsel keine weiteren Bilder in den Stack aufgenommen, sondern es wird der alte Zustand eingefroren.

Deklaration

```

BOOL PDLRTPictureNavigation (
    PDLRT_PNFLAGS    flags,
    PCMN_ERROR       pError );

```

Parameter

flags

Über diesen Parameter wird die Funktion gesteuert. Es sind folgende Werte möglich:

PDLRT_PNF_ENABLE	Schaltet die Bildnavigation ein.
PDLRT_PNF_DISABLE	Schaltet die Bildnavigation aus.
PDLRT_PNF_ENABLE_LOAD_LAST_PICTURE	Schaltet die Bildnavigation ein und lädt das letzte Bild im Bildspeicher.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Funktion erfolgreich beendet.

FALSE

Fehler.

Ursache: Bildnavigation war bereits ein- oder ausgeschaltet.

Benötigte Dateien

pdlrtapi.h

pdlrt_s.lib

pdlrtapi.dll

Verwandte Funktionen

PDLRTGotoPicture (Seite 560)	Spezifiziertes Bild laden
------------------------------	---------------------------

Siehe auch

PDLRTGotoPicture (Seite 560)

Übersicht über die Funktionen (Seite 522)

2.3.3.8 PDLRTShowApp (RT Professional)

Verwendung

Durch die Funktion wird das eventuell im Hintergrund liegende Runtime-Fenster an die Oberfläche des Desktop gebracht.

Deklaration

```
BOOL PDLRTShowApp (
    PDLRT_CALLBACK    pfn,
    LPVOID            pvUser,
    PCMN_ERROR        pError );
```

Parameter

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Runtime-Fenster im Vordergrund.

FALSE

Fehler.

Benötigte Dateien

pdLrtapi.h

pdLrt_s.lib

pdLrtapi.dll

Verwandte Funktionen

PDLRT_CALLBACK (Seite 568)	Callback-Funktion
----------------------------	-------------------

Siehe auch

PDLRT_CALLBACK (Seite 568)

Übersicht über die Funktionen (Seite 522)

2.3.3.9 PDLRT_CALLBACK (RT Professional)

Beschreibung

Wenn Ihre Applikation in Runtime über die Durchführung von API-Funktionen asynchron benachrichtigt werden soll, müssen Sie Callback-Funktionen vom Typ PDLRT_CALLBACK bereitstellen.

Diese Funktion wird von allen Funktionen des PDLRT als Callback-Funktion genutzt.

Deklaration

```
void ( * PDLRT_CALLBACK ) (
    LPVOID      pvUser,
    PCMN_ERROR  pError );
```

Parameter

pvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird als Parameter der API-Funktion zur Verfügung gestellt, und hier wieder unverändert zurückgegeben.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

Diese Callback hat keinen Rückgabewert.

Bemerkungen

Wenn die API-Funktion ohne Callback, also mit pfn = NULL aufgerufen wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden. Ohne Synchronisierung besteht auch die Gefahr, dass die asynchrone Funktion noch Daten über den nicht mehr gültigen Daten-Zeiger schreibt, wenn die rufende Bearbeitung schon beendet ist, bzw. den Gültigkeitsbereich verlassen hat, was zur Ausnahmebehandlung führt.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

In seltenen Fällen kann es sogar vorkommen, dass die Callback geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Benötigte Dateien

pdlrtapi.h

Siehe auch

PDLRTCclosePicture (Seite 555)
PDLRTDisableClosePicture (Seite 557)
PDLRTEnableClosePicture (Seite 558)
PDLRTInquireFreeArea (Seite 561)
PDLRTOpenPicture (Seite 563)
PDLRTShowApp (Seite 566)
PDLRTSetLink (Seite 590)
PDLRTSetMultiLink (Seite 592)
PDLRTGetLink (Seite 588)
PDLRTSetPropEx (Seite 584)
PDLRTGetPropEx (Seite 581)
PDLRTGetDefPropEx (Seite 578)
PDLRTSetFocus (Seite 576)
PDLRTSetCursorKeys (Seite 574)
PDLRTGetFocus (Seite 572)
PDLRTGetCursorKeys (Seite 570)
Übersicht über die Funktionen (Seite 522)

2.3.4 Funktionen zur Beeinflussung des Runtime-Cursors (RT Professional)

2.3.4.1 PDLRTGetCursorKeys (RT Professional)

Verwendung

Mit der Funktion werden die Tasten für die Cursorsteuerung abgefragt, die mit PDLRTSetCursorKeys gesetzt wurden.

Deklaration

```
BOOL PDLRTGetCursorKeys (  
    long*          pKeyUp,  
    long*          pKeyDown,  
    long*          pKeyLeft,  
    long*          pKeyRight,  
    long*          pKeyState,  
    long*          pTabMode,  
    PDLRT_CALLBACK pfn,  
    LPVOID         pvUser,  
    PCMN_ERROR     pError );
```

Parameter

KeyUp

Über diesen Zeiger wird der virtuelle Tastencode (VK_...) der Taste geliefert, die den Cursor aufwärts bewegt.

pKeyDown

Über diesen Zeiger wird der virtuelle Tastencode (VK_...) der Taste geliefert, die den Cursor abwärts bewegt.

pKeyLeft

Über diesen Zeiger wird der virtuelle Tastencode (VK_...) der Taste geliefert, die den Cursor nach links bewegt.

pKeyRight

Über diesen Zeiger wird der virtuelle Tastencode (VK_...) der Taste geliefert, die den Cursor nach rechts bewegt.

pKeyState

Über diesen Zeiger wird der Tastaturstatus geliefert:

HOTKEYF_SHIFT	(Wert: 0x01)	"SHIFT"-Taste gedrückt
HOTKEYF_CONTROL	(Wert: 0x02)	"STRG"-Taste gedrückt
HOTKEYF_ALT	(Wert: 0x04)	"ALT"-Taste gedrückt

Die Werte können beliebig verodert sein.

pTabMode

Über diesen Zeiger wird der TabMode geliefert, welches Objekt bei der nächsten Betätigung einer der Cursorsteuerungstasten angesprungen wird:

O:	xxBildxx
1:	xxBildxx
10:	xxBildxx
	xxBildxx

pfm

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfm = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Cursorsteuerungstasten ermittelt.

FALSE

Fehler.

Benötigte Dateien

pdlrtapi.h

pdlrt_s.lib

pdlrtapi.dll

Verwandte Funktionen

PDLRTSetCursorKeys (Seite 574)	Cursorsteuerungstasten festlegen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

- PDLRTSetCursorKeys (Seite 574)
- PDLRT_CALLBACK (Seite 568)
- Übersicht über die Funktionen (Seite 522)

2.3.4.2 PDLRTGetFocus (RT Professional)

Verwendung

Die Funktion liefert das Bild und den Objektnamen des Objekts, das den Eingabefokus besitzt.

Deklaration

```
BOOL PDLRTGetFocus (
    ADRMODE          adrMode,
    LPFOCUSINFO      pFocusInfo,
    PDLRT_CALLBACK   pfn,
    LPVOID           pvUser,
    PCMN_ERROR       pError );
```

Parameter

adrMode

Über den Parameter adrMode wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt erfolgt relativ
PDLRT_AM_PICTABS	1	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objekts ist absolut

Die Werte PDLRT_AM_PICTABS und PDLRT_AM_OBJABS können verodert werden.

pFocusInfo

Zeiger auf die Struktur FOCUSINFO (Seite 554) , in der die Ergebnisse abgelegt werden.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Eingabefokus ermittelt.

FALSE

Fehler.

Benötigte Dateien

pdLrtapi.h
pdLrt_s.lib
pdLrtapi.dll

Verwandte Funktionen

PDLRTSetFocus (Seite 576)	Eingabefokus festlegen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

FOCUSINFO (Seite 554)
PDLRTSetFocus (Seite 576)
PDLRT_CALLBACK (Seite 568)
Übersicht über die Funktionen (Seite 522)

2.3.4.3 PDLRTSetCursorKeys (RT Professional)

Verwendung

Mit der Funktion werden die Tasten für die Cursorsteuerung festgelegt.

Deklaration

```
BOOL PDLRTSetCursorKeys (  
    long        KeyUp,  
    long        KeyDown,  
    long        KeyLeft,  
    long        KeyRight,  
    long        KeyState,  
    long        TabMode,  
    PDLRT_CALLBACK pfn,  
    LPVOID      pvUser,  
    PCMN_ERROR   pError );
```

Parameter

KeyUp

Virtueller Tastencode (VK_...) der Taste, die den Cursor aufwärts bewegt.

KeyDown

Virtueller Tastencode (VK_...) der Taste, die den Cursor abwärts bewegt.

KeyLeft

Virtueller Tastencode (VK_...) der Taste, die den Cursor nach links bewegt.

KeyRight

Virtueller Tastencode (VK_...) der Taste, die den Cursor nach rechts bewegt.

KeyState

KeyState gibt den Tastaturstatus wieder:

HOTKEYF_SHIFT	(Wert: 0x01)	"SHIFT"-Taste gedrückt
HOTKEYF_CONTROL	(Wert: 0x02)	"STRG"-Taste gedrückt
HOTKEYF_ALT	(Wert: 0x04)	"ALT"-Taste gedrückt

Die Werte können beliebig verodert werden.

TabMode

Über TabMode wird festgelegt, welches Objekt bei der nächsten Betätigung einer der Cursorsteuerungstasten angesprungen wird:

0:	xxBildxx
1:	xxBildxx
10:	xxBildxx
	xxBildxx

pfm

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfm = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Cursorsteuerungstasten definiert.

FALSE

Fehler.

Benötigte Dateien

pdLrtapi.h

pdLrt_s.lib

pdLrtapi.dll

Verwandte Funktionen

PDLRTGetCursorKeys (Seite 570)	Cursorsteuerungstasten abfragen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

- PDLRTGetCursorKeys (Seite 570)
- PDLRT_CALLBACK (Seite 568)
- Übersicht über die Funktionen (Seite 522)

2.3.4.4 PDLRTSetFocus (RT Professional)

Verwendung

Mit der Funktion wird der Eingabefokus gesetzt. Das Objekt ist durch `lpszPictureName` und `lpszObjectName` bestimmt.

Deklaration

```
BOOL PDLRTSetFocus (
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    LPCSTR           lpszObjectName,
    PDLRT_CALLBACK   pfn,
    LPVOID           pvUser,
    PCMN_ERROR       pError );
```

Parameter

adrMode

Über den Parameter `adrMode` wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt ist relativ
PDLRT_AM_PICTABS	1	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objekts ist absolut

Die Werte `PDLRT_AM_PICTABS` und `PDLRT_AM_OBJABS` können verodert werden.

lpszPictureName

Zeiger auf den projektierten Namen des Bildes entsprechend dem über `adrMode` definierten Adressierungsmodus, ohne Extension. Groß- und Kleinbuchstaben werden unterschieden.

IpszObjectName

Zeiger auf den projizierten Objektnamen entsprechend dem über adrMode definierten Adressierungsmodus.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Eingabefokus gesetzt

FALSE

Fehler

Fehlermeldungen

PDLRT_NO_PIC	Bild nicht angewählt
PDLRT_NO_OBJ	Objekt nicht gefunden

Benötigte Dateien

pdlrtapi.h

pdlrt_s.lib

pdlrtapi.dll

Verwandte Funktionen

PDLRTGetFocus (Seite 572)	Eingabefokus abfragen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

- PDLRTGetFocus (Seite 572)
- PDLRT_CALLBACK (Seite 568)
- Übersicht über die Funktionen (Seite 522)

2.3.5 Funktionen zum Bearbeiten von Objekteigenschaften (RT Professional)

2.3.5.1 PDLRTGetDefPropEx (RT Professional)

Verwendung

Einen Default-Property-Wert anfordern. Wenn der angeforderte Datentyp nicht mit dem Datentyp des Properties übereinstimmt, wird der Wert beim Übertragen in den Anwender-Puffer pvProp entsprechend dem angeforderten Format gewandelt, sofern möglich.

Deklaration

```
BOOL PDLRTGetDefPropEx(  
    ADRMODE        adrMode,  
    LPCSTR         lpszPictureName,  
    LPCSTR         lpszObjectName,  
    LPCSTR         lpszPropName,  
    VARTYPE        vt,  
    LPVOID         pvProp,  
    PDLRT_CALLBACK pfn,  
    LPVOID         pvUser,  
    DWORD          dwFlags,  
    LPVOID         pData,  
    PCMN_ERROR     pError );
```

Parameter

adrMode

Über den Parameter adrMode wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt ist relativ
PDLRT_AM_PICTABS	0	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objekts ist absolut

Die Werte PDLRT_AM_PICTABS und PDLRT_AM_OBJABS können verodert werden.

IpszPictureName

Zeiger auf den projektierten Namen des Bildes, entsprechend dem über adrMode definierten Adressierungsmodus. Groß- und Kleinbuchstaben werden unterschieden.

IpszObjectName

Zeiger auf den projektierten Objektnamen, entsprechend dem über adrMode definierten Adressierungsmodus.

IpszPropName

Zeiger auf den projektierten Namen der Objekteigenschaft.

vt

Datentyp des unter pvProp übergebenen Werts. Die erlaubten Typen sind in der zum Compiler gehörenden Include-Datei "oidl.h" und "wtypes.h" definiert. Für Skripte können Sie die WinCC Include-Datei APLIB\Defines.h verwenden.

pvProp

Zeiger auf eine Variable, in welcher der Property-Wert abgelegt wird. Der Datentyp des Werts ist durch vt bestimmt:

vt	pvProp
VT_BSTR	BSTR*
VT_LPSTR	LPSTR*
VT_LPWSTR	LPWSTR*
VT_UI4	LONG*
VT_xxxx	pvProp muss auf einen Puffer zeigen

Bei Typen mit Puffer (z. B. BSTR) wird der Puffer von der Funktion allokiert und ist von der aufrufenden Applikation wieder frei zu geben.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

pData

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbesetzt werden.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Defaultwerte der Objekteigenschaft ermittelt.

FALSE

Fehler.

Bemerkungen

Um allokierte Speicherbereiche wieder frei zu geben, benutzen Sie im Fall VT_BSTR die Windows-API-Funktion SysFreeString(...) und im Falle VT_LPSTR, VT_LPWSTR die Windows-API-Funktion LocalFree(...).

Fehlermeldungen

PDLRT_NO_PROP	Property nicht gefunden
PDLRT_BAD_OLE_CONVERSION	Fehler bei Wandlung über OLE Automation

Benötigte Dateien

- pdlrtapi.h
- pdlrt_s.lib
- pdlrtapi.dll

Verwandte Funktionen

PDLRT_CALLBACK (Seite 568)	Callback-Funktion
----------------------------	-------------------

Siehe auch

PDLRT_CALLBACK (Seite 568)

Übersicht über die Funktionen (Seite 522)

2.3.5.2 PDLRTGetPropEx (RT Professional)

Verwendung

Anfordern eines aktuellen Property-Werts. Die Quelle ist durch lpszPictureName, lpszObjectName und lpszPropName bestimmt.

Wenn der angeforderte Datentyp nicht mit dem Datentyp des Property übereinstimmt, wird der Wert beim Übertragen in den Anwender-Puffer pvProp entsprechend dem angeforderten Format gewandelt, sofern möglich.

Deklaration

```

BOOL PDLRTGetPropEx (
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    LPCSTR           lpszObjectName,
    LPCSTR           lpszPropName,
    VARTYPE          vt,
    LPVOID           pvProp,
    PDLRT_CALLBACK   pfn,
    LPVOID           pvUser,
    DWORD            dwFlags,
    LPVOID           pData,
    PCMN_ERROR       pError );

```

Parameter

adrMode

Über den Parameter adrMode wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt erfolgt relativ
PDLRT_AM_PICTABS	0	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objekts ist absolut

Die Werte PDLRT_AM_PICTABS und PDLRT_AM_OBJABS können verodert werden.

lpzPictureName

Zeiger auf den projektorierten Namen des Bildes entsprechend dem über adrMode definierten Adressierungsmodus. Groß- und Kleinbuchstaben werden unterschieden.

lpzObjectName

Zeiger auf den projektorierten Namen des Objekts entsprechend dem über adrMode definierten Adressierungsmodus.

lpzPropName

Zeiger auf den projektorierten Namen der Objekteigenschaft.

vt

Datentyp des unter pvProp übergebenen Werts. Die erlaubten Typen sind in der zum Compiler gehörenden Include-Datei "oaidl.h" und "wtypes.h" definiert. Für Skripte können Sie die WinCC Include-Datei APLIB\Defines.h verwenden.

pvProp

Zeiger auf eine Variable, in welcher der Property-Wert abgelegt wird. Der Datentyp des Werts ist durch vt bestimmt:

vt	pvProp
VT_BSTR	BSTR*
VT_LPSTR	LPSTR*
VT_LPWSTR	LPWSTR*
VT_UI4	LONG*
VT_xxxx	pvProp muss auf einen Puffer zeigen

Bei Typen mit Puffer (z. B. BSTR) wird der Puffer von der Funktion allokiert und ist von der aufrufenden Applikation wieder frei zu geben.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

pData

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbesetzt werden.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Objekteigenschaften ermittelt.

FALSE

Fehler.

Bemerkungen

Um allokierte Speicherbereiche wieder frei zu geben, benutzen Sie im Fall VT_BSTR die Windows-API-Funktion SysFreeString(...) und im Falle VT_LPSTR, VT_LPWSTR die Windows-API-Funktion LocalFree(...).

Es darf kein VT_DISPATCH und andere Referenzen verwendet werden, lediglich die normalen Typen und max. ein VT_VARIANT für ein Array-Property einfacher Typen.

Fehlermeldungen

PDLRT_NO_PIC	Bild nicht angewählt
PDLRT_BAD_OLE_CONVERSION	Fehler bei Wandlung über OLE Automation
PDLRT_NO_PROP	Property nicht gefunden

Benötigte Dateien

pdIrtapi.h
pdIrt_s.lib
pdIrtapi.dll

Verwandte Funktionen

PDLRTSetPropEx (Seite 584)	Property setzen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

- PDLRT_CALLBACK (Seite 568)
- PDLRTSetPropEx (Seite 584)
- Übersicht über die Funktionen (Seite 522)

2.3.5.3 PDLRTSetPropEx (RT Professional)

Verwendung

Setzen eines Property. Das Ziel ist durch lpszPictureName, lpszObjectName und lpszPropName bestimmt. Stimmt der übergebene Datentyp nicht mit dem Datentyp des Property überein, so wird der Wert beim Übertragen auf das Property gewandelt, sofern möglich.

Es können nur solche Eigenschaften gesetzt werden, die auch dynamisierbar sind.

Deklaration

```

BOOL PDLRTSetPropEx (
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    LPCSTR           lpszObjectName,
    LPCSTR           lpszPropName,
    VARTYPE          vt,
    LPVOID           pvProp,
    PDLRT_CALLBACK  pfn,
    LPVOID           pvUser,
    DWORD            dwFlags,
    LPVOID           pData,
    PCMN_ERROR       pError);

```

Parameter

adrMode

Über den Parameter adrMode wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt ist relativ
PDLRT_AM_PICTABS	0	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objekts ist absolut

Die Werte PDLRT_AM_PICTABS und PDLRT_AM_OBJABS können verodert werden.

lpszPictureName

Zeiger auf den projektierten Namen des Bildes entsprechend dem über adrMode definierten Adressierungsmodus. Groß- und Kleinbuchstaben werden unterschieden.

IpszObjectName

Zeiger auf den projizierten Objektnamen entsprechend dem über adrMode definierten Adressierungsmodus.

IpszPropName

Zeiger auf den projizierten Namen der Objekteigenschaft.

vt

Datentyp des unter pvProp übergebenen Werts. Die erlaubten Typen sind in der zum Compiler gehörenden Include-Datei "oidl.h" und "wtypes.h" definiert. Für Skripte können Sie die WinCC Include-Datei APLIB\Defines.h verwenden.

pvProp

Zeiger auf einen Puffer, in dem der neue Property-Wert abgelegt ist. Der Datentyp des Werts ist durch vt bestimmt:

vt	pvProp
VT_BSTR	BSTR*
VT_LPSTR	LPSTR*
VT_LPWSTR	LPWSTR*
VT_UI4	LONG*
VT_xxxx	pvProp muss auf einen Puffer zeigen

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

pData

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbesetzt werden.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Objekteigenschaften gesetzt.

FALSE

Fehler.

Bemerkungen

Es darf kein VT_DISPATCH und andere Referenzen verwendet werden, lediglich die normalen Typen und max. ein VT_VARIANT für ein Array-Property einfacher Typen.

Fehlermeldungen

PDLRT_NO_PIC	Bild nicht angewählt
PDLRT_PICTURE_NOT_LOADED	Bilddatei konnte nicht geladen werden
PDLRT_FAILURE_PARAM	Fehlerhafte Parameter
PDLRT_NO_OBJ	Objekt nicht gefunden
PDLRT_NO_PROP	Property nicht gefunden
PDLRT_BAD_OLE_CONVERSION	Fehler bei Wandlung über OLE Automation

Benötigte Dateien

pdlrtapi.h

pdlrt_s.lib

pdlrtapi.dll

Verwandte Funktionen

PDLRTGetPropEx (Seite 581)	Property abfragen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Beispiel

```

#pragma code("pdlrtapi.dll")

#include "apdefap.h"

#define CP_ACP 0
#define MB_PRECOMPOSED 1

void ButtonRT(char* text)
{
    DM_DIRECTORY_INFO dirInfo;
    CMN_ERROR error;
    BOOL bRet;
    char szProjectFile[222] = "";
    WCHAR wstring[99];
    BSTR bstrval;
    WORD *pw = NULL;
    int i;

    printf("Funktion ButtonRT() with text '%s'\r\n", text);

    bRet = DMGetRuntimeProject
(&szProjectFile[0], sizeof(szProjectFile), &error);
    printf("DMGetRuntimeProject: %s\r\n", szProjectFile);

    bRet = PDLRTOpenPicture(0, "Start.PDL", NULL, NULL, 0, 0, 0, 0, 0, NULL, NULL,
&error);
    printf("PDLRTOpenPicture: %s,%ld, %s\r\n", bRet?"TRUE":"FALSE",
error.dwError1, error.szErrorText);

    bRet = PDLRTSetPropEx(0, "Start.PDL", "Button4", "Text", VT_LPSTR,
&text, NULL, NULL, 0, NULL, &error);
    printf("PDLRTSetPropEx: %s,%ld, %s\r\n", bRet?"TRUE":"FALSE",
error.dwError1, error.szErrorText);

    SysFreeString( bstrval );
    printf("SysFreeString done\r\n");
}

```

Siehe auch

[PDLRT_CALLBACK \(Seite 568\)](#)

[PDLRTGetPropEx \(Seite 581\)](#)

[Übersicht über die Funktionen \(Seite 522\)](#)

2.3.6 Funktionen zum Bearbeiten von Dynamiken (RT Professional)

2.3.6.1 PDLRTGetLink (RT Professional)

Verwendung

Erfragen eines Links von einem Property. Wenn das Property eine indirekte Variablenverbindung besitzt, wird diese zurückgeliefert. Wenn bei indirekter Variablenverbindung die momentan aktuelle direkte Verbindung gewünscht ist, kann diese durch Angabe von `LinkType = BUBRT_LT_VARIABLE_DIRECT` in der Struktur `LINKINFO` angefordert werden.

Deklaration

```
BOOL PDLRTGetLink (
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    LPCSTR           lpszObjectName,
    LPCSTR           lpszPropName,
    LPLINKINFO       pLink,
    PDLRT_CALLBACK  pfn,
    LPVOID           pvUser,
    PCMN_ERROR       pError );
```

Parameter

adrMode

Über den Parameter `adrMode` wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

<code>PDLRT_AM_DEFAULT</code>	0	Die Adressierung von Bild und Objekt erfolgt relativ
<code>PDLRT_AM_PICTABS</code>	1	Die Adressierung des Bildes ist absolut
<code>PDLRT_AM_OBJABS</code>	2	Die Adressierung des Objekts ist absolut

Die Werte `PDLRT_AM_PICTABS` und `PDLRT_AM_OBJABS` können verodert werden.

lpszPictureName

Zeiger auf den projektorierten Namen des Bildes entsprechend dem über `adrMode` definierten Adressierungsmodus, ohne Extension. Groß- und Kleinbuchstaben werden unterschieden.

lpszObjectName

Zeiger auf den projektorierten Objektnamen entsprechend dem über `adrMode` definierten Adressierungsmodus.

lpszPropName

Zeiger auf den Namen der Objekteigenschaft.

pLink

Zeiger auf die Struktur LINKINFO (Seite 551) , in der die Linkinformation abgelegt wird.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Verknüpfung ermittelt.

FALSE

Fehler.

Fehlermeldungen

PDLRT_NO_LINK	Property hat keine Dynamik
---------------	----------------------------

Benötigte Dateien

pdLrtapi.h

pdLrt_s.lib

pdLrtapi.dll

Verwandte Funktionen

PDLRTSetLink (Seite 590)	Verknüpfung Objekteigenschaft-Variable festlegen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

- PDLRTSetLink (Seite 590)
- PDLRT_CALLBACK (Seite 568)
- LINKINFO (Seite 551)
- Übersicht über die Funktionen (Seite 522)

2.3.6.2 PDLRTSetLink (RT Professional)

Verwendung

Setzen einer Verknüpfung zwischen einem Property und einer Variablen.

Deklaration

```

BOOL PDLRTSetLink (
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    LPCSTR           lpszObjectName,
    LPCSTR           lpszPropName,
    LPLINKINFO       pLink,
    PDLRT_CALLBACK   pfn,
    LPVOID           pvUser,
    PCMN_ERROR       pError );

```

Parameter

adrMode

Über den Parameter adrMode wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt ist relativ
PDLRT_AM_PICTABS	1	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objekts ist absolut

Die Werte PDLRT_AM_PICTABS und PDLRT_AM_OBJABS können verodert werden.

lpszPictureName

Zeiger auf den projektierten Namen des Bildes entsprechend dem über adrMode definierten Adressierungsmodus, ohne Extension. Groß- und Kleinbuchstaben werden unterschieden.

IpszObjectName

Zeiger auf den projizierten Objektnamen entsprechend dem über adrMode definierten Adressierungsmodus.

IpszPropName

Zeiger auf den Namen der Objekteigenschaft.

pLink

Zeiger auf die Struktur LINKINFO (Seite 551) , in der die Linkinformation abgelegt ist.

In LinkType dürfen nur BUBRT_LT_VARIABLE_DIRECT und BUBRT_LT_VARIABLE_INDIRECT verwendet werden, da hier nur Verknüpfung mit Variablen zugelassen sind.

pfn

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfn = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Verknüpfung gesetzt

FALSE

Fehler

Fehlermeldungen

PDLRT_FAILURE_PARAM	Fehlerhafte Parameter
PDLRT_NO_PIC	Bild nicht angewählt

PDLRT_NO_OBJ	Objekt nicht gefunden
PDLRT_LINK_NOT_SET	Property konnte nicht verbunden werden

Benötigte Dateien

pdlrtapi.h
pdlrt_s.lib
pdlrtapi.dll

Verwandte Funktionen

PDLRTGetLink (Seite 588)	Verknüpfung Objekteigenschaft-Variable ermitteln
PDLRTSetMultiLink (Seite 592)	Verknüpfung Objekteigenschaft-Variable festlegen (mehrere Variablen)
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

PDLRTGetLink (Seite 588)
PDLRTSetMultiLink (Seite 592)
PDLRT_CALLBACK (Seite 568)
LINKINFO (Seite 551)
Übersicht über die Funktionen (Seite 522)

2.3.6.3 PDLRTSetMultiLink (RT Professional)

Verwendung

Setzen mehrerer Verknüpfungen zwischen einem Property und Variablen.

Deklaration

```
BOOL PDLRTSetMultiLink (
    ADRMODE          adrMode,
    LPCSTR           lpszPictureName,
    LPMULTILINK      pMultiLink,
    PDLRT_CALLBACK   pfn,
    LPVOID           pvUser,
    PCMN_ERROR       pError );
```

Parameter

adrMode

Über den Parameter adrMode wird der Adressierungsmodus des zu bearbeitenden Bildes festgelegt.

PDLRT_AM_DEFAULT	0	Die Adressierung von Bild und Objekt ist relativ
PDLRT_AM_PICTABS	1	Die Adressierung des Bildes ist absolut
PDLRT_AM_OBJABS	2	Die Adressierung des Objekts ist absolut

Die Werte PDLRT_AM_PICTABS und PDLRT_AM_OBJABS können verodert werden.

lpszPictureName

Zeiger auf den projektierten Namen des Bildes entsprechend dem über adrMode definierten Adressierungsmodus, ohne Extension. Groß- und Kleinbuchstaben werden unterschieden.

pMultiLink

Zeiger auf die Struktur MULTILINK (Seite 552) , in der ein Feld mit Linkinformationen abgelegt ist.

pfm

Zeiger auf Ihre Callback-Funktion. Diese wird aufgerufen, um Ihnen den Erfolg bzw. Misserfolg des Auftrags mitzuteilen.

Wenn pfm = NULL übergeben wird, ist der Aufruf synchron. Die aufrufende Applikation wird so lange wartend gesetzt, bis von PDLRT ein Fehler oder Erfolg gemeldet wird. Diese Art des Aufrufs sollte für voneinander abhängige Aufruf-Sequenzen bevorzugt verwendet werden.

Wenn nachfolgende API-Aufrufe darauf angewiesen sind, dass diese Funktion beendet wurde und diese Funktion asynchron mit Callback verwendet wird, sollte die API-Aufrufe mit Semaphoren synchronisiert werden.

Es ist sinnvoll, für jede API-Funktion eine eigene Callback zu verwenden.

pvUser

Zeiger auf applikationsspezifische Daten, der an die Callback-Funktion weitergereicht wird. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Der Rückgabewert TRUE wird nur geliefert, wenn alle Verknüpfungen erfolgreich gesetzt werden konnten. Wenn auch nur eine Verknüpfung nicht gesetzt werden konnte wird FALSE geliefert.

FALSE

Fehler.

Fehlermeldungen

PDLRT_FAILURE_PARAM	Fehlerhafte Parameter
PDLRT_NO_PIC	Bild nicht angewählt
PDLRT_NO_OBJ	Objekt nicht gefunden
PDLRT_LINK_NOT_SET	Property konnte nicht verbunden werden

Benötigte Files

pdlrtapi.h
pdlrt_s.lib
pdlrtapi.dll

Verwandte Funktionen

PDLRTSetLink (Seite 590)	Verknüpfung Objekteigenschaft-Variable festlegen
PDLRT_CALLBACK (Seite 568)	Callback-Funktion

Siehe auch

PDLRTSetLink (Seite 590)
PDLRT_CALLBACK (Seite 568)
MULTILINK (Seite 552)
Übersicht über die Funktionen (Seite 522)

2.4 Funktionen des Scripting (RT Professional)

2.4.1 Grundlagen (RT Professional)

2.4.1.1 Übersicht über die Funktionen (RT Professional)

Übersicht

AP_RT_PROC (Seite 611)	Benachrichtigungsfunktion der Runtime-Funktionen
APActive (Seite 622)	Aktion aktivieren
APCompile (Seite 613)	Quellcode compilieren
APCompileEx (Seite 616)	Quellcode compilieren
APConnect (Seite 605)	Applikation anmelden
APDisconnect (Seite 607)	Applikation abmelden

APEndAct (Seite 624)	Aktion bei der Aktionssteuerung abmelden
APFreeResultStruct (Seite 626)	Aktionsergebnisspeicher freigeben
APInactive (Seite 627)	Aktionen deaktivieren
APSetLanguage (Seite 610)	Sprache der Fehlertexte festlegen
APStart (Seite 629)	Aktion starten
APTransact (Seite 631)	Aktionen zur Ausführung anmelden
GSCGenCompile (Seite 618)	Aktion übersetzen
GSCGenCompileUserFunctions (Seite 620)	Anwender-Funktionen übersetzen
GSCGenGetActionStream (Seite 621)	Aktionsstream ermitteln

2.4.1.2 Übersicht über die Strukturen (RT Professional)

Übersicht

AP_ACT_KEY (Seite 599)	Aktionsidentifikation
AP_ACT_RESULT_STRUCT (Seite 602)	Aktionsergebnis
CREATE_USER_HEADER_FILE (Seite 603)	Header-Datei von Anwender-Funktionen anlegen
GENERATE_COMPILE (Seite 604)	Aktion erzeugen
GET_ACTION_STREAM (Seite 604)	Aktionsstream ermitteln

2.4.1.3 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

Allgemeine Fehlermeldungen

AP_ALREADY_CONNECTED	2	Applikation ist mit der Aktionssteuerung schon verbunden
AP_NO_CONNECTION	3	Applikation kann keine Verbindung mit der Aktionssteuerung aufnehmen.
AP_ERROR_IPC_SEND	4	Fehler bei der Kommunikation zur Aktionssteuerung. Der Auftrag konnte nicht abgesetzt werden.
AP_FAILURE_UNKNOWN	5	Nicht näher beschriebener Fehler
AP_FAILURE_PARAM	6	Fehlerhafte Parameterversorgung
AP_NO_ACT_PROGRAM	7	Aktionssteuerung konnte nicht gestartet werden
AP_TIMEOUT	8	Zeitüberschreitung, momentan noch nicht realisiert.
AP_ACT_QUIT	9	Aktionssteuerung hat sich beendet
AP_INSTALL_SERV_ERROR	10	Servicekanal konnte nicht installiert werden

AP_ENDACT_UNKNOWN_ORDER	11	Für den EndAct wurde eine nicht bekannte Auftragsnummer verwendet
AP_ACTION_FAILED	12	Eine Aktion konnte nicht fehlerfrei ausgeführt werden. Die Rückgabeergebnisse sind ungültig.
AP_FAILURE_IN_SERVER	13	Vom Server wird ein Fehler gemeldet.
AP_TO_MANY_CLIENTS	14	Die maximale Anzahl von Verbindungen ist erreicht. Keine neuen Verbindungen möglich.
AP_TRANSACTID_UNKNOWN	15	Transaktions-ID ist nicht gültig. Fehler tritt beim Aufruf von AP_EndAct auf, wenn versucht wird, eine Transaktion zu beenden, die vorher nicht angemeldet wurde.
AP_NO_MEMORY	16	Es steht nicht mehr genügend Speicher für diese Operation zu Verfügung
AP_TRANSACT_ERROR	17	Innerhalb der Transaktion ist ein Fehler aufgetreten aktionsbezogener Fehler in dwerror des AP_ACT_KEY
AP_RESULT_TRANS_ERROR	18	Bei den Ergebnissen ist ein Fehler aufgetreten
AP_RESULT_START_ERROR	19	Bei den Ergebnissen ist ein Fehler aufgetreten
AP_NO_UPDATE_WRONG_FORMAT	50	Für das Datenformat besteht keine Updatemöglichkeit Aktion beinhaltet keinen Interpretercode.
AP_ERR_WRONG_FORMAT	202	Aktion besitzt falsches Datenformat.
AP_NO_VALID_FUNCTION_VALUE	203	Funktionsrückgabewert kann nicht in den Datentyp VARIANT gewandelt werden.

Fehler

die bei der Durchführung einer Aktion auftreten können

AP_CISS_ERR_EXIT_OVERFLOW	1001	Im Aktionsinterpreter ist ein Stapelüberlauf während der Ausführung aufgetreten. Weitere Ausführung der Aktion wird abgebrochen.
AP_CISS_ERR_EXIT_DIVIDE0	1002	Innerhalb der Ausführung einer Aktion ist eine Division durch 0 aufgetreten. Aktion wird abgebrochen.
AP_CISS_ERR_EXIT_UNRESOLVED	1003	Innerhalb der Aktion wurde während der Ausführung auf ein nicht vorhandenes Symbol referenziert.
AP_CISS_ERR_EXIT_GPF	1004	Innerhalb der Aktion wurde während der Ausführung auf nicht definierten Speicher zugegriffen.
AP_CISS_ERR_EXIT_BREAKPOINT	1005	Der Aktionsinterpreter lief auf einen Breakpoint.
AP_CISS_ERR_EXIT_STEP	1006	Der Aktionsinterpreter wurde im Debugger um einen Bearbeitungsschritt fortgesetzt.

Meldungen

des Aktionsinterpreters

AP_CISS_ERR_CREATE_PCH_FROM_PCH	8001	Eine precompiled Headerdatei kann nicht aus einer precompiled Headerdatei generiert werden.
AP_CISS_ERR_MODULE_IN_USE	8002	Auf die Aktion kann nicht zugegriffen werden. Das Modul wird momentan benutzt.
AP_CISS_ERR_INVALID_PROGRAM	8003	Das Programm ist ungültig.
AP_CISS_ERR_INVALID_MODULE	8004	Die Aktion ist ungültig.
AP_CISS_ERR_CANNOT_CREATE_FILE	8005	Datei konnte von der Aktionssteuerung nicht angelegt werden.
AP_CISS_ERR_CANNOT_NO_MEMORY	8006	Der Aktionsinterpreter hat nicht mehr genügend Speicher zu Verfügung.
AP_CISS_ERR_INVALID_FILE_FORMAT	8007	Dateiformat ist für die Aktionssteuerung ungültig.
AP_CISS_ERR_CANNOT_OPEN_FILE	8008	Datei konnte von der Aktionssteuerung nicht geöffnet werden.
AP_CISS_ERR_PROGRAM_IS_LOCKED	8009	Programm ist momentan von der Aktionssteuerung gesperrt.
AP_CISS_ERR_MODULE_ALREADY_INSERTED	8010	Aktion wurde bereits der Aktionssteuerung zur Bearbeitung gegeben.
AP_CISS_ERR_CONFLICT_WITH_OTHER	8011	In der Aktion ist ein Konflikt mit einer anderen Aktion aufgetreten.
AP_CISS_ERR_MODULE_NOT_FOUND	8013	Aktion wurde von der Aktionssteuerung nicht gefunden.
AP_CISS_ERR_FUNCTION_NOT_FOUND	8014	Funktion wurde von der Aktionssteuerung nicht gefunden.
AP_CISS_ERR_INVALID_LINE	8015	Angegebene Zeileninformation ist ungültig.
AP_CISS_ERR_INVALID_SCOPE	8016	Angegebenes Symbol ist außerhalb des Gültigkeitsbereichs.
AP_CISS_ERR_BUFFER_TOO_SMALL	8017	Der übergebene Speicher ist für den Aktionsinterpreter zu klein.
AP_CISS_ERR_INVALID_TYPE	8018	Der angegebene Typ ist dem Aktionsinterpreter unbekannt.
AP_CISS_ERR_SYMBOL_NOT_FOUND	8019	Das angegebene Symbol wurde nicht gefunden.

Notify Codes

AP_NOTIFY_ERROR	0	Bei der Durchführung der Funktion ist ein Fehler aufgetreten.
AP_NOTIFY_DATA	1	Bei der Durchführung der Funktion ist kein Fehler aufgetreten.
AP_NOTIFY_CODE_TRANSACT	1	Bestätigung eines AP_TransAct Aufrufs
AP_NOTIFY_CODE_START	2	Bestätigung eines AP_Start Aufrufs

2.4 Funktionen des Scripting (RT Professional)

AP_NOTIFY_CODE_RESULT	3	Ergebnis einer Aktion
AP_NOTIFY_CODE_DISCONNECT	6	
AP_NOTIFY_CODE_ENDACT	7	Bestätigung eines AP_EndAct Aufrufs
AP_NOTIFY_CODE_ACTIVE	8	Bestätigung eines AP_Active Aufrufs
AP_NOTIFY_CODE_INACTIVE	9	Bestätigung eines AP_Inactive Aufrufs
AP_NOTIFY_ERROR_SERVER_QUITT	10	Benachrichtigung, dass sich die Aktionssteuerung beendet hat.
AP_NOTIFY_CODE_TRANSRESULT	11	Notify für einen fehlerhaften Result Aufruf eines APTransAct
AP_NOTIFY_CODE_STARTRESULT	12	Notify für einen fehlerhaften Result Aufruf eines APStart
AP_NOTIFY_CODE_RESULT_RT	13	Notify für zyklische Ergebnisse eines optimierten Aufrufs

2.4.1.4 Konstanten (RT Professional)

Funktions / Aktions - Kennung

GSC_AP_SFCT	0x00000011	Standard - Funktion
GSC_AP_PFCT	0x00000012	Projekt - Funktion
GSC_AP_GSC	0x00000014	GSC - Aktion

Grenzwerte

AP_MAX_TRIG_NAME	21	Maximale Länge des Namens eines Trigger
------------------	----	---

Headertypen

CMHF_APDEFAP	0x00000001	Header einer Projekt-Definition
CMHF_AP_PBIB	0x00000002	Header einer Projekt-Funktion
CMHF_AP_GLOB	0x00000004	Header einer Standard-Funktion
CMHF_AP_ICF	0x00000008	Header einer Internal-Funktion
CMHF_AP_ALL	CMHF_APDEFAP CMHF_AP_PBIB CMHF_AP_GLOB CMHF_AP_ICF	
CMHF_AP_USER	0x00000010	Header einer Anwender-Funktion

Triggertypen

AP_TRIG_UNDEFINED	0	Trigger ist noch nicht initialisiert.
AP_TRIG_VAR	1	Trigger ist eine Variable.
AP_TRIG_TIMER	2	Trigger ist ein Timer.

AP_TRIG_UNKNOWN	3	Trigger ist unbekannt.
AP_TRIG_TRANSACT	4	Trigger wird über TransAct Aufruf spezifiziert.

Zyklustypen

AP_TRIG_CYCLE	1	Zyklisch
AP_TRIG_NCYCLE	2	Azyklisch

Zykluszeiten

AP_TRIG_CYCLE_01	1	entspricht im Control Center der Zykluszeit 250 ms
AP_TRIG_CYCLE_02	2	entspricht im Control Center der Zykluszeit 500 ms
AP_TRIG_CYCLE_03	3	entspricht im Control Center der Zykluszeit 1 s
AP_TRIG_CYCLE_04	4	entspricht im Control Center der Zykluszeit 2 s
AP_TRIG_CYCLE_05	5	entspricht im Control Center der Zykluszeit s
AP_TRIG_CYCLE_06	6	entspricht im Control Center der Zykluszeit 10 s
AP_TRIG_CYCLE_07	7	entspricht im Control Center der Zykluszeit 1 min
AP_TRIG_CYCLE_08	8	entspricht im Control Center der Zykluszeit 5 min
AP_TRIG_CYCLE_09	9	entspricht im Control Center der Zykluszeit 10 min
AP_TRIG_CYCLE_10	10	entspricht im Control Center der Zykluszeit 1 h
AP_TRIG_CYCLE_11	11	entspricht im Control Center der Zykluszeit "Anwenderdefiniert 1"
AP_TRIG_CYCLE_12	12	entspricht im Control Center der Zykluszeit "Anwenderdefiniert 2"
AP_TRIG_CYCLE_13	13	entspricht im Control Center der Zykluszeit "Anwenderdefiniert 3"
AP_TRIG_CYCLE_14	14	entspricht im Control Center der Zykluszeit "Anwenderdefiniert 4"
AP_TRIG_CYCLE_15	15	entspricht im Control Center der Zykluszeit "Anwenderdefiniert 5"

2.4.2 Strukturen (RT Professional)

2.4.2.1 AP_ACT_KEY (RT Professional)

Deklaration

```
typedef struct {
    DWORD        dwKeyType;
    DWORD        dwID;
    CHAR        szActionName[AP_MAX_ACTION_NAME + 1];
    DWORD        dwCycle;
    VARIANT      *pVariant;
    DWORD        dwVariantItem;
    DWORD        dwerror;
    LPVOID       lpvUser;
} AP_ACT_KEY ;
```

Members

dwKeyType

Der Schlüsseltyp legt fest ob die Aktion über einen Namen (szActionName) oder über ID (dwID) adressiert wird.

AP_ID_TYPE	0	Identifizierung der Aktion über ihre ID
AP_NAME_TYPE	1	Identifizierung der Aktion über ihren Namen

Projekt- und Standardfunktionen, die bereits im WinCC-Projekt vorhanden sind, werden über den Name identifiziert. Benutzerdefinierte Funktionen, die erst mit der Funktion APTransAct angemeldet werden, werden über ID identifiziert.

dwID

In Verbindung mit dwKeyType = AP_ID_TYPE wird die Aktion mit dwID gestartet.

Die Id der Aktion wird beim Aufruf der Funktion APTransAct vergeben. Beim Aufruf der Funktion APTransAct ist dwID mit 0 vorzubsetzen.

szActionName

In Verbindung mit dwKeyType = AP_NAME_TYPE wird die Aktion über szActionName gestartet.

Kann beim Starten einer Aktion mit APStart verwendet werden.

dwCycle

Zyklisches Starten der Aktion wenn per Name adressiert. Der Aktualisierungszyklus wird über den Index der Einträge in der Liste der Aktualisierungszyklen definiert.

pVariant

Zeiger auf Feld vom Datentyp VARIANT, welches die Parameter der Aktion beschreibt. Als VARTYPES werden folgende Datentypen unterstützt :

unsigned char	VT_UI1
unsigned integer 2 Bytes	VT_UI2 (WORD)
unsigned integer 4 Bytes	VT_UI4 (DWORD)
short	VT_I2
long	VT_I4
float	VT_R4
double	VT_R8
VARIANT_BOOL	VT_BOOL
SCODE	VT_ERROR
CY	VT_CY
DATE	VT_DATE
BSTR	VT_BSTR
BLOB	VT_BLOB

Die Parameterbeschreibungen der Aktionen und des Rückgabeergebnisses ist nur in der Form des Datentyps VARIANT möglich.

Eine Übergabe als Referenz ist nicht erlaubt.

dwVariantItem

Anzahl der VARIANT-Datentypen, welche die Aktion als Parameter erhält.

dwerror

Fehlernummer, die beim APTransAct auftreten kann.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

Beschreibung

Die Struktur AP_ACT_KEY identifiziert die Aktion eindeutig. Mit dem API-Aufruf APTransAct wird dieser Schlüssel als ID vergeben. Mit diesem Schlüssel wird die Aktion gestartet.

Benötigte Dateien

ap_def.h

API - Funktionen

APActive (Seite 622)	Aktion aktivieren
APInactive (Seite 627)	Aktionen deaktivieren
APStart (Seite 629)	Aktion starten
APTransact (Seite 631)	Aktionen zur Ausführung anmelden

Siehe auch

APActive (Seite 622)

APInactive (Seite 627)

APStart (Seite 629)

APTransact (Seite 631)

2.4.2.2 AP_ACT_RESULT_STRUCT (RT Professional)

Deklaration

```
typedef struct {  
    VARIANT      *ap_result;  
    AP_ACT_KEY   apActKey;  
    CMN_ERROR    error;  
    DWORD        dwreserved;  
} AP_ACT_RESULT_STRUCT ;
```

Members

***ap_result**

Rückgabeergebnis der Aktion in Form eines VARIANT Datentypes.

apActKey

Schlüssel der Aktion zur Identifizierung. Der AP_ACT_KEY entspricht dem Schlüssel, mit dem die Aktion gestartet wurde.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

dwreserved

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

Benötigte Dateien

ap_def.h

API - Funktionen

APFreeResultStruct (Seite 626)	Aktionsergebnisspeicher freigeben
APStart (Seite 629)	Aktion starten

Siehe auch

APFreeResultStruct (Seite 626)

APStart (Seite 629)

2.4.2.3 CREATE_USER_HEADER_FILE (RT Professional)

Deklaration

```
typedef struct {  
    char*      pszStartDir;  
    char*      pszHeaderFileName;  
    BOOL       bShowDlg;  
    char*      pszWindowText;  
} CREATE_USER_HEADER_FILE ;
```

Members

pszStartDir

Zeiger auf das Startverzeichnis, ab welchem nach Anwender-Funktionen gesucht wird mit abschließendem "\"

pszHeaderFileName

Zeiger auf den Namen des Header-Files.

bShowDlg

Dialog mit Fortschrittsanzeige anzeigen

pszWindowText

Zeiger auf die Dialogbeschriftung. Default: BROWSER

Benötigte Dateien

capigsc.h

API - Funktionen

GSCGenCompileUserFunctions (Seite 620)	Anwender-Funktionen übersetzen
--	--------------------------------

Siehe auch

GSCGenCompileUserFunctions (Seite 620)

2.4.2.4 GENERATE_COMPILE (RT Professional)

Deklaration

```
typedef struct {  
    char*      pszProjectName;  
    LPACTION  pAction;  
} GENERATE_COMPILE ;
```

Members

pszProjectName

Zeiger auf den Projektnamen

pAction

Zeiger auf gültigen Aktionsstream mit Source-Code

Benötigte Dateien

capigsc.h

Verwandte Funktionen

GSCGenCompile (Seite 618)	Aktion übersetzen
---------------------------	-------------------

Siehe auch

GSCGenCompile (Seite 618)

2.4.2.5 GET_ACTION_STREAM (RT Professional)

Deklaration

```
typedef struct {  
    char*      pszPathName;  
    DWORD      dwType;  
} GET_ACTION_STREAM, *LPGET_ACTION_STREAM ;
```

Members

pszPathName

vollständiger Dateiname der Funktion bzw. Aktion.

dwType

zulässige Typen:

GSC_AP_SFCT	Standard - Funktion
GSC_AP_PFCT	Projekt - Funktion
GSC_AP_GSC	Aktion

Benötigte Dateien

capigsc.h

API Funktionen

GSCGenGetActionStream (Seite 621)	Aktionsstream ermitteln
-----------------------------------	-------------------------

Siehe auch

GSCGenGetActionStream (Seite 621)

2.4.3 Allgemeine Funktionen (RT Professional)**2.4.3.1 APConnect (RT Professional)****Beschreibung**

Die Funktion meldet eine Applikation bei der Aktionssteuerung an. Wird in fpAppBack eine Callback-Funktion angegeben, so wird die Funktion asynchron ausgeführt, bei fpAppBack == NULL wird die Aktion synchron ausgeführt.

Deklaration

```

BOOL APConnect (
    LPCSTR      lpszAppName,
    AP_RT_PROC  fpAppBack,
    PDWORD     pdwOrderId,
    LPCVOID     lpvUser,
    PCMN_ERROR  pError )

```

Parameter

lpszAppName

Name der Applikation, mit der sich die Applikation beim DMConnect angemeldet hat. Ein DMConnect ist deshalb vorher nötig. APConnect wird nur synchron durchgeführt.

Werden Funktionen der Skript-Programmierung in Aktionen aufgerufen, so ist für lpszAppName der Wert "AktSteu" (exakte Schreibweise beachten) anzugeben, weil die WinCC-Task den DMConnect mit diesem Namen durchgeführt hat.

fpAppBack

Ihre Callback-Funktion, die Benachrichtigungen entgegennimmt. Die Mitteilung, dass sich die Aktionssteuerung beendet hat, wird z.B. über diese Funktion mitgeteilt.

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Bei synchronem Aufruf ist pdwOrderID nicht von Bedeutung, bei asynchronem Auftrag wird die Auftragsnummer in der Callback-Funktion mitgeliefert.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

Wird lpvUser nicht verwendet, muss er mit NULL belegt werden.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Applikation angemeldet

FALSE

Fehler

Fehlermeldungen

AP_NO_ERROR	Kein Fehler aufgetreten
AP_NO_CONNECTION	Applikation kann keine Verbindung mit der Aktionssteuerung aufnehmen.
AP_ERROR_IPC_SEND	Fehler bei der Kommunikation zur Aktionssteuerung. Der Auftrag konnte nicht abgesetzt werden.
AP_TIMEOUT	Zeitüberschreitung, momentan noch nicht realisiert.

AP_ALREADY_CONNECTED	Applikation ist mit der Aktionssteuerung schon verbunden
AP_FAILURE_PARAM	Fehlerhafte Parameterversorgung
AP_TO_MANY_CLIENTS	Die maximale Anzahl von Verbindungen ist erreicht. Keine neuen Verbindungen möglich.
AP_FAILURE_UNKNOWN	Nicht näher beschriebener Fehler
AP_ACT_QUIT	Aktionssteuerung hat sich beendet

Benötigte Dateien

ap_def.h
 apcli_S.lib
 apclient.dll

Verwandte Funktionen

DMConnect (Seite 278)	Verbindung zum Datenmanager aufbauen
APDisconnect (Seite 607)	Verbindung zur Aktionssteuerung abbauen
APTransact (Seite 631)	Aktion zur Ausrührung anmelden
AP_RT_PROC (Seite 611)	Callback-Funktion

Beispiele

Establish connection to script programming (Seite 633) "AP01.cpp"

Siehe auch

DMConnect (Seite 278)
 APTransact (Seite 631)
 AP_RT_PROC (Seite 611)
 APDisconnect (Seite 607)
 Establish connection to script programming (Seite 633)
 Übersicht über die Funktionen (Seite 594)

2.4.3.2 APDisconnect (RT Professional)

Beschreibung

Die Funktion meldet eine Applikation bei der Aktionssteuerung ab.

Deklaration

```
BOOL APDisconnect (
    AP_RT_PROC    fpAppBack,
    PDWORD        pdwOrderId,
    LPCVOID       lpvUser,
    PCMN_ERROR    pError )
```

Parameter

fpAppBack

Ihre Callback-Funktion, die Benachrichtigungen entgegennimmt. Bei Verwendung einer Callback Funktion erfolgt der Auftrag asynchron. Mit fpAppBack == NULL erfolgt er synchron.

Wenn ein Programm eine Notify-Routine anmeldet, muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Bei synchronem Aufruf ist pdwOrderID nicht von Bedeutung, bei asynchronem Auftrag wird die Auftragsnummer in der Callback-Funktion mitgeliefert.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Applikation abgemeldet.

FALSE

Fehler

Bemerkung

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX, ...) benutzt werden. Aufgrund von Microsoft-spezifischen Mechanismen kann das gegebenenfalls zum Hängenbleiben des Aufrufs und damit des Programms führen.

Fehlermeldungen

AP_NO_ERROR	Kein Fehler aufgetreten
AP_NO_CONNECTION	Applikation kann keine Verbindung mit der Aktionssteuerung aufnehmen.
AP_ERROR_IPC_SEND	Fehler bei der Kommunikation zur Aktionssteuerung. Der Auftrag konnte nicht abgesetzt werden.
AP_TIMEOUT	Zeitüberschreitung, momentan noch nicht realisiert.
AP_ALREADY_CONNECTED	Applikation ist mit der Aktionssteuerung schon verbunden
AP_FAILURE_PARAM	Fehlerhafte Parameterversorgung
AP_TO_MANY_CLIENTS	Die maximale Anzahl von Verbindungen ist erreicht. Keine neuen Verbindungen möglich.
AP_FAILURE_UNKNOWN	Nicht näher beschriebener Fehler
AP_ACT_QUIT	Aktionssteuerung hat sich beendet

Benötigte Dateien

ap_def.h
 apcli_S.lib
 apclient.dll

Verwandte Funktionen

APConnect (Seite 605)	Verbindung zur Aktionssteuerung aufbauen
APTransact (Seite 631)	Aktion zur Ausführung anmelden
AP_RT_PROC (Seite 611)	Callback-Funktion

Beispiele

Establish connection to script programming (Seite 633) "AP01.cpp"

Siehe auch

APTransact (Seite 631)
 APConnect (Seite 605)
 AP_RT_PROC (Seite 611)
 Establish connection to script programming (Seite 633)
 Übersicht über die Funktionen (Seite 594)

2.4.3.3 APSetLanguage (RT Professional)

Beschreibung

Über diese Funktion wird die Sprache in der die Fehlertexte ausgegeben werden sollen festgelegt.

Deklaration

```
BOOL APSetLanguage (  
    const DWORD    dwLanguageID )
```

Parameter

dwLanguageID

ID der Sprache entsprechend der Windows Spracheinstellung. Die Fehlertexte werden entsprechend, der neu eingestellten Sprache zurückgeliefert. Wird die Sprache nicht unterstützt, so wird eine Default Sprache verwendet.

Rückgabewert

TRUE

Sprache der Fehlertexte geändert

FALSE

Fehler

Benötigte Dateien

ap_def.h

apcli_S.lib

apclient.dll

Siehe auch

Übersicht über die Funktionen (Seite 594)

2.4.3.4 AP_RT_PROC (RT Professional)

Beschreibung

Wenn Ihre Applikation in Runtime über die Durchführung von API-Funktionen asynchron benachrichtigt werden soll, müssen Sie Callback-Funktionen vom Typ AP_RT_PROC bereitstellen.

Diese Funktion wird von allen Funktionen der Aktionsprogrammierung als Callback-Funktion genutzt.

Deklaration

```

BOOL ( * AP_RT_PROC) (
    DWORD    dwAP_Notify,
    WORD     dwAP_NotifyCode,
    DWORD    dwError,
    LPVOID   lpvData,
    DWORD    dwItems,
    DWORD    dwOrderId,
    LPVOID   lpvUser );

```

Parameter

dwAP_Notify

Beschreibt die Art der Callback-Funktion. Mögliche Werte sind AP_NOTIFY_ERROR und AP_NOTIFY_DATA.

dwAP_NotifyCode

Wenn dwAP_Notify == AP_NOTIFY_ERROR, dann zeigt lpvData auf eine Struktur vom Typ CMN_ERROR mit der Fehlerbeschreibung, und dwAP_NotifyCode ist NULL.

Wenn dwAP_Notify == AP_NOTIFY_DATA, dann enthält dwAP_NotifyCode eine genauere Spezifikation der Callback-Funktion:

AP_NOTIFY_CODE_TRANSACT	Bestätigung eines Aufrufs durch APTransAct
AP_NOTIFY_CODE_START	Bestätigung eines Aufrufs durch APStart
AP_NOTIFY_CODE_RESULT	Ergebnis einer Aktio

dwError

Fehlernummer

IpvData

Zeiger auf bereitgestellte Daten. Die Datenstruktur ist abhängig von dwAP_Notify und dwAP_NotifyCode.

dwItems

Anzahl der Einträge in IpvData.

dwOrderId

Auftragsnummer, die beim Aufruf der API-Funktion vergeben wurde.

IpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

In seltenen Fällen kann es sogar vorkommen, dass die Callback geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Benötigte Dateien

ap_def.h

Verwandte Funktionen

APActive (Seite 622)	Aktion aktivieren
APConnect (Seite 605)	Applikation anmelden
APDisconnect (Seite 607)	Applikation abmelden
APEndAct (Seite 624)	Aktion bei der Aktionssteuerung abmelden

APInactive (Seite 627)	Aktionen deaktivieren
APStart (Seite 629)	Aktion starten
APTransact (Seite 631)	Aktionen zur Ausführung anmelden

Beispiele

Establish connection to script programming (Seite 633) "AP01.cpp"

Siehe auch

APConnect (Seite 605)

APDisconnect (Seite 607)

APActive (Seite 622)

APEndAct (Seite 624)

APInactive (Seite 627)

APStart (Seite 629)

APTransact (Seite 631)

Establish connection to script programming (Seite 633)

Übersicht über die Funktionen (Seite 594)

2.4.4 Funktionen zum Bearbeiten von Source Code (RT Professional)

2.4.4.1 APCompile (RT Professional)

Beschreibung

Ein in lpvScode hinterlegter Quellcode wird übersetzt und als P-Code in lpvPcode hinterlegt. Die Applikation muss den Speicher für den P-Code bereitstellen.

Deklaration

```
BOOL APCompile (
    LPCSTR      szProjectName,
    PDWORD      pdwOrderId,
    LPCSTR      lpvScode,
    const DWORD dwScodeSize,
    LPVOID      *lpvPcode,
    PDWORD      dwPcodeSize,
    const HWND  hwndLog,
    const DWORD dwDebugFlag,
    PDWORD      nErrors,
    PDWORD      nWarnings,
    LPCVOID     lpvUser,
    PCMN_ERROR  pError )
```

Parameter

szProjectName

Zeichenkette, die einen gültigen Projektpfad enthält. Über den Projektpfad wird entschieden, welche Include und Precompiled Header zum Kompilieren verwendet werden.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Die Auftragsnummer muss vom Aufrufer zur Verfügung gestellt werden.

lpvScode

Zeiger auf den zu übersetzenden Quellcode.

dwScodeSize

Größe des Quellcodes in Bytes.

lpvPcode

Adresse eines Zeigers, der bei erfolgreichem Übersetzen den P-Code enthält. Der Speicher wird von der Funktion APCompile angelegt und muss vom Aufrufer wieder mit der Funktion APFreePCode freigegeben werden.

dwPcodeSize

Enthält nach dem Kompilieren die Größe des P-Codes in Bytes.

hwndLog

Fensterhandle, über den die Ausgabe der Fehlermeldungen erfolgen kann. Der Fensterhandle kann auch NULL sein. Bei der Verwendung eines Fensters zur Ausgabe erfolgt eine Message über die Windows-Funktion WM_COPYDATA.

dwDebugFlag

Dieses Flag legt fest, ob der P-Code Debug-Informationen mitliefern soll (Bit1 = 1) oder ob keine Debug-Information geliefert werden sollen (Bit1 = 0). Normalerweise sollte aufgrund der Performance "0" gewählt werden.

Bit2 = 1: Server Einstellungen für Includes verwenden

Bit2 = 0: Lokale Einstellungen für Includes verwenden

nErrors

Anzahl der aufgetretenen Fehler. Bei nErrors > 0 wird kein P-Code zurückgeliefert.

nWarnings

Anzahl der aufgetretenen Warnungen. Auch bei nWarnings > 0 wird der P-Code erzeugt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Source-Code erfolgreich kompiliert

FALSE

Fehler

Benötigte Dateien

ap_def.h

apcli_S.lib

apclient.dll

Verwandte Funktionen

APCompileEx (Seite 616)	Quellcode kompilieren
APFreePCode	Quellcodespeicher freigeben

Siehe auch

APCompileEx (Seite 616)

Übersicht über die Funktionen (Seite 594)

2.4.4.2 APCompileEx (RT Professional)

Beschreibung

Im Unterschied zur Funktion APCompile ist der zu compilierende Source-Code in Teilbereiche unterteilt.

Im Source-Code wird void function {Anweisungen} unterteilt in

void function {	Anfang der Anweisung
Anweisungen	Rumpf
}	Ende der Anweisung

Ein in den Teilbereichen lpvScodeProlog, lpvScodeBase und lpvScodeEpilog hinterlegter Quellcode wird übersetzt und als P-Code in lpvPcode hinterlegt. Die Applikation muss den Speicher für den P-Code bereitstellen.

Deklaration

```

BOOL APCompileEx (
    LPCSTR      szProjectName,
    PDWORD     pdwOrderId,
    LPCSTR      lpvScodeProlog,
    const DWORD dwScodePrologSize,
    LPCSTR      lpvScodeBase,
    const DWORD dwScodeBaseSize,
    LPCSTR      lpvScodeEpilog,
    const DWORD dwScodeEpilogSize,
    LPVOID      *lpvPcode,
    PDWORD     dwPcodeSize,
    const HWND  hwndLog,
    const DWORD dwDebugFlag,
    PDWORD     nErrors,
    PDWORD     nWarnings,
    LPCVOID     lpvUser,
    PCMN_ERROR pError )
    
```

Parameter

szProjectName

Zeichenkette, die einen gültigen Projektpfad enthält. Über den Projektpfad wird entschieden, welche Include und Precompiled Header zum Kompilieren verwendet werden.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Die Auftragsnummer muss vom Aufrufer zur Verfügung gestellt werden.

szProjectName

Zeichenkette, die einen gültigen Projektpfad enthält. Über den Projektpfad wird entschieden, welche Include und Precompiled Header zum Kompilieren verwendet werden.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Die Auftragsnummer muss vom Aufrufer zur Verfügung gestellt werden.

IpvScodeProlog

Zeiger auf den Prolog des zu compilierenden Source-Codes.

dwScodePrologSize

Größe des Prologs in Byte.

IpvScodeBase

Zeiger auf den Rumpf des zu compilierenden Source-Codes.

dwScodeBaseSize

Größe des Rumpfes in Byte.

IpvScodeEpilog

Zeiger auf den Epilog des zu compilierenden Source-Codes.

dwScodeEpilogSize

Größe des Epilogs in Byte.

IpvPcode

Adresse eines Zeigers, der bei erfolgreichem Übersetzen den P-Code enthält. Der Speicher wird von der Funktion APCompile angelegt und muss vom Aufrufer wieder mit der Funktion APFreePCode freigegeben werden.

dwPcodeSize

Enthält nach dem Kompilieren die Größe des P-Codes in Bytes.

hwndLog

Fensterhandle, über den die Ausgabe der Fehlermeldungen erfolgen kann. Der Fensterhandle kann auch NULL sein. Bei der Verwendung eines Fensters zur Ausgabe erfolgt eine Message über die Windows-Funktion WM_COPYDATA.

dwDebugFlag

Dieses Flag legt fest, ob der P-Code Debug-Informationen mitliefern soll (Bit1 = 1) oder ob keine Debug-Information geliefert werden sollen (Bit1 = 0). Normalerweise sollte aufgrund der Performance "0" gewählt werden.

Bit2 = 1: Server Einstellungen für Includes verwenden

Bit2 = 0: Lokale Einstellungen für Includes verwenden

nErrors

Anzahl der aufgetretenen Fehler. Bei nErrors > 0 wird kein P-Code zurückgeliefert.

nWarnings

Anzahl der aufgetretenen Warnungen. Auch bei nWarnings > 0 wird der P-Code erzeugt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Source-Code erfolgreich compiliert

FALSE

Fehler

Benötigte Dateien

ap_def.h

apcli_S.lib

apclient.dll

Verwandte Funktionen

APCompile (Seite 613)	Quellcode compilieren
APFreePCCode	Quellcodespeicher freigeben

Siehe auch

APCompile (Seite 613)

Übersicht über die Funktionen (Seite 594)

2.4.4.3 GSCGenCompile (RT Professional)

Beschreibung

Die Funktion übersetzt eine Aktion.

Deklaration

```
LPACTION GSCGenCompile(  
    LPGENERATE_COMPILE lpGenCompile,  
    HWND hWndParent,  
    unsigned long* plErrors,  
    unsigned long* plWarnings,  
    AllocAppMem lpfnAllocAppMem,  
    LPCMN_ERROR lpdmError)
```

Parameter

lpGenCompile

Zeiger auf die Struktur GENERATE_COMPILE (Seite 604).

hWndParent

Handle auf das Fenster, in dem die Statusausgaben erfolgen sollen.

plErrors

Zeiger auf Variable für Rückgabe der Anzahl der Fehler

plWarnings

Zeiger auf Variable für Rückgabe der Anzahl der Warnungen

lpfnAllocAppMem

Zeiger auf eine Funktion, mit der bei der aufrufenden Applikation Speicher allokiert wird.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

Zeiger auf den übersetzten Aktionsstream. Ein neuer Aktionsstream wird auch dann erzeugt, wenn die Übersetzung fehlerhaft ist. Der Erfolg des Aufrufs muss anhand der Zahl von Fehlern und Warnungen überprüft werden.

Bemerkung

Die übersetzte Aktion steht erst nach dem nächsten Start von Runtime zur Verfügung.

Benötigte Dateien

capigsc.h

gscgr_s.lib

gscgen.dll

Siehe auch

GENERATE_COMPILE (Seite 604)

Übersicht über die Funktionen (Seite 594)

2.4.4.4 GSCGenCompileUserFunctions (RT Professional)

Beschreibung

Die Funktion übersetzt alle speziellen Anwender-Funktionen neu.

Deklaration

```
BOOL GSCGenCompileUserFunctions (  
    LPCREATE_USER_HEADER_FILE    pGenCUHF,  
    HWND                          hWndParent,  
    LPCMN_ERROR                   lpdmError)
```

Parameter

pGenCUHF

Zeiger auf die Struktur CREATE_USER_HEADER_FILE (Seite 603).

hWndParent

Handle auf das Fenster, in dem die Statusausgaben erfolgen sollen.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Anwender-Funktionen übersetzt

FALSE

Fehler

Bemerkung

Aus internen Gründen liefert diese Funktion immer FALSE ohne Fehlermeldung zurück, obwohl die Funktionen generiert werden.

Die übersetzten Anwender-Funktionen stehen erst nach dem nächsten Start von Runtime zur Verfügung.

Benötigte Dateien

capigsc.h
gscgr_s.lib
gscgen.dll

Siehe auch

CREATE_USER_HEADER_FILE (Seite 603)
Übersicht über die Funktionen (Seite 594)

2.4.5 Funktionen zum Bearbeiten von Aktionen (RT Professional)

2.4.5.1 GSCGenGetActionStream (RT Professional)

Beschreibung

Die Funktion ermittelt den Aktionsstream zu einer Projektfunktion, einer Standardfunktion oder einer Aktio.

Deklaration

```
LPACTION GSCGenGetActionStream(  
    LPGET_ACTION_STREAM    pGenGAS,  
    AllocAppMem            lpfnAllocAppMem,  
    LPCMN_ERROR            lpdmError)
```

Parameter

pGenGAS

Zeiger auf eine Struktur vom Typ GET_ACTION_STREAM (Seite 604) , über die die Funktion bzw. Aktion spezifiziert wird.

lpfnAllocAppMem

Zeiger auf eine Funktion, mit der bei der aufrufenden Applikation Speicher allokiert wird.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

Zeiger auf den Aktionsstream.

Benötigte Dateien

capigsc.h
gscgr_s.lib
gscgen.dll

Siehe auch

GET_ACTION_STREAM (Seite 604)
Übersicht über die Funktionen (Seite 594)

2.4.6 Funktionen der Aktionsprogrammierung (RT Professional)

2.4.6.1 APActive (RT Professional)

Beschreibung

Aktionen, die mit APInactive deaktiviert wurden, können mit dieser Funktion wieder aktiviert werden, d.h. die zugehörigen Trigger werden wieder überwacht bzw. können mit APStart wieder gestartet werden.

Deklaration

```
BOOL APActive (
    PAP_ACT_KEY    lpapActKey,
    const DWORD    dwItems,
    AP_RT_PROC     fpAppBack,
    PDWORD         pdwOrderId,
    LPCVOID        lpvUser,
    PCMN_ERROR     pError )
```

Parameter

lpapActKey

Zeiger auf eine Datenstruktur mit folgendem Aufbau:

AP_ACT_KEY 1
AP_ACT_KEY 2
⋮
AP_ACT_KEY n
Actionstream 1
⋮
Actionstream 1
Actionstream n

Die Datenstruktur muss von der aufrufenden Applikation bereitgestellt werden.

Die Aktionen müssen vorher mit APTransAct angemeldet sein. Die ID in AP_ACT_KEY entscheidet, ob die Aktion über die ID oder über den Namen gestartet wird.

dwItems

Anzahl der Aktionen, also der Strukturen AP_ACT_KEY (Seite 599).

fpAppBack

Ihre Callback Funktion für den asynchronen Aufruf. Bei Verwendung von NULL wird der Aufruf synchron ausgeführt.

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Bei synchronem Aufruf ist pdwOrderID nicht von Bedeutung, bei asynchronem Auftrag wird die Auftragsnummer in der Callback-Funktion mitgeliefert.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktion aktiviert.

FALSE

Fehler.

Benötigte Dateien

ap_def.h
apcli_S.lib
apclient.dll

Verwandte Funktionen

APInactive (Seite 627)	Aktionen deaktivieren
APTransact (Seite 631)	Aktion zur Ausführung anmelden
AP_RT_PROC (Seite 611)	Callback-Funktion

Siehe auch

AP_RT_PROC (Seite 611)
AP_ACT_KEY (Seite 599)
APInactive (Seite 627)
APTransact (Seite 631)
Übersicht über die Funktionen (Seite 594)

2.4.6.2 APEndAct (RT Professional)

Beschreibung

Die zur Ausführung angemeldete Aktion wird bei der Aktionssteuerung abgemeldet.

Deklaration

```
BOOL APEndAct (
    AP_RT_PROC      fpAppBack,
    PDWORD          pdwOrderId,
    const PDWORD    pdwOrderEnd,
    LPCVOID         lpvUser,
    PCMN_ERROR      pError )
```

Parameter

fpAppBack

Ihre Callback Funktion für den asynchronen Aufruf. Bei Verwendung von NULL wird der Aufruf synchron ausgeführt.

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Bei synchronem Aufruf ist pdwOrderID nicht von Bedeutung, bei asynchronem Auftrag wird die Auftragsnummer in der Callback-Funktion mitgeliefert.

pdwOrderEnd

Auftragsnummer der Transaktion, die beendet werden soll.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Aktion abgemeldet.

FALSE

Fehler.

Benötigte Dateien

ap_def.h

apcli_S.lib

apclient.dll

Verwandte Funktionen

APTransact (Seite 631)	Aktion zur Ausführung anmelden
AP_RT_PROC (Seite 611)	Callback-Funktion

Siehe auch

AP_RT_PROC (Seite 611)

APTransact (Seite 631)

Übersicht über die Funktionen (Seite 594)

2.4.6.3 APFreeResultStruct (RT Professional)

Beschreibung

Beim synchronen Aufruf der Funktion APStart werden die Aktionsergebnisse in Form von einem Array von Strukturen des Typs AP_ACT_RESULT_STRUCT allokiert. Der von APStart allokierte Speicher muss mit der Funktion APFreeResultStruct wieder freigegeben werden.

Deklaration

```
BOOL APFreeResultStruct (
    PAP_ACT_RESULT_STRUCT *lpapars,
    const DWORD           dwItems )
```

Parameter

lpvPcode

Zeiger auf die erste von dwItems Strukturen vom Typ AP_ACT_RESULT_STRUCT (Seite 602).

dwItems

Anzahl der Strukturen AP_ACT_RESULT_STRUCT.

Rückgabewert

TRUE

Speicher freigegeben.

FALSE

Fehler.

Benötigte Dateien

- ap_def.h
- apcli_S.lib
- apclient.dll

Verwandte Funktionen

APStart (Seite 629)	Ausführung der Aktion gestartet
---------------------	---------------------------------

Siehe auch

APStart (Seite 629)

AP_ACT_RESULT_STRUCT (Seite 602)

Übersicht über die Funktionen (Seite 594)

2.4.6.4 APInactive (RT Professional)**Beschreibung**

Aktionen, die mit APTransAct angemeldet worden sind, können mit APInactive deaktiviert werden. Die zugehörigen Trigger werden nicht mehr überwacht und die Aktionen können nicht mit APStart gestartet werden. Die Aktionen können mit APActive wieder aktiviert werden.

Deklaration

```

BOOL APInactive (
    PAP_ACT_KEY    lpActKey,
    const DWORD    dwItems,
    AP_RT_PROC     fpAppBack,
    PDWORD         pdwOrderId,
    LPCVOID        lpvUser,
    PCMN_ERROR     pError )

```

Parameter**lpapActKey**

Zeiger auf eine Datenstruktur mit folgendem Aufbau:

AP_ACT_KEY 1
AP_ACT_KEY 2
:
AP_ACT_KEY n
Actionstream 1
:
Actionstream 1
Actionstream n

Die Datenstruktur muss von der aufrufenden Applikation bereitgestellt werden.

Die Aktionen müssen vorher mit APTransAct angemeldet sein. Die ID in AP_ACT_KEY entscheidet, ob die Aktion über die ID oder über den Namen gestartet wird.

dwItems

Anzahl der Aktionen, also der Strukturen AP_ACT_KEY (Seite 599).

fpAppBack

Ihre Callback Funktion für den asynchronen Aufruf. Bei Verwendung von NULL wird der Aufruf synchron ausgeführt.

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Bei synchronem Aufruf ist pdwOrderID nicht von Bedeutung, bei asynchronem Auftrag wird die Auftragsnummer in der Callback-Funktion mitgeliefert.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktion deaktiviert.

FALSE

Fehler.

Benötigte Dateien

ap_def.h

apcli_S.lib

apclient.dll

Verwandte Funktionen

APActive (Seite 622)	Aktionen aktivieren
APTransact (Seite 631)	Aktion zur Ausführung anmelden
AP_RT_PROC (Seite 611)	Callback-Funktion

Siehe auch

AP_RT_PROC (Seite 611)
 APActive (Seite 622)
 AP_ACT_KEY (Seite 599)
 APTransact (Seite 631)
 Übersicht über die Funktionen (Seite 594)

2.4.6.5 APStart (RT Professional)**Beschreibung**

Die Aktionen werden zur Ausführung gestartet. Die gleichzeitige Ausführung mehrerer Aktionen ist möglich. Die Aktionen sind in der Struktur AP_ACT_KEY beschrieben.

Deklaration

```

BOOL APStart (
    AP_ACT_KEY          lpapActKey,
    const DWORD         dwItems,
    AP_RT_PROC          fpAppBack,
    AP_RT_PROC          fpAppResult,
    PDWORD              pdwOrderId,
    PAP_ACT_RESULT_STRUCT *lpapars,
    LPCVOID             lpUser,
    PCMN_ERROR          pError )
  
```

Parameter**lpapActKey**

Zeiger auf eine Datenstruktur mit folgendem Aufbau:

AP_ACT_KEY 1
AP_ACT_KEY 2
⋮
AP_ACT_KEY n
Actionstream 1
⋮
Actionstream 1
Actionstream n

Die Datenstruktur muss von der aufrufenden Applikation bereitgestellt werden.

Die Aktionen müssen vorher mit APTransAct angemeldet sein. Die ID in AP_ACT_KEY entscheidet, ob die Aktion über die ID oder über den Namen gestartet wird.

dwItems

Anzahl der Aktionen, also der Strukturen AP_ACT_KEY (Seite 599).

fpAppBack

Ihre Callback Funktion für den asynchronen Aufruf. Bei Verwendung von NULL wird der Aufruf synchron ausgeführt.

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

fAppResult

Ihre Callback-Funktion für zyklisch ablaufende Aktionsergebnisse. In der Callback-Funktion werden die Strukturen vom Typ AP_ACT_RESULT_STRUCT (Seite 602) geliefert. Bei synchronem Aufruf ist lpapars mit dem Ergebnis belegt.

pdwOrderId

Auftragsnummer, die beim Aufruf der Funktion APTransAct vergeben wird. Bei synchronem Aufruf ist pdwOrderID nicht von Bedeutung, bei asynchronem Auftrag wird die Auftragsnummer in der Callback-Funktion mitgeliefert.

lpapars

Zeiger auf das Aktionsergebnis bei synchronem Aufruf. Entsprechend der Anzahl der gestarteten Aktionen wird ein Array von Strukturen vom Typ AP_ACT_RESULT_STRUCT (Seite 602) zurückgeliefert. Der Speicher wird von der Funktion allokiert und muss mit APFreeResultStruct wieder freigegeben werden.

Bei asynchroner Verwendung ist lpapars = NULL

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

pError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktion gestartet.

FALSE

Fehler.

Bemerkung

Es können mit APStart nur Aktionen gestartet werden, die zuvor nicht mit APInactive deaktiviert wurden.

Benötigte Dateien

ap_def.h

apcli_S.lib

apclient.dll

Verwandte Funktionen

APFreeResultStruct (Seite 626)	Speicher freigeben
AP_RT_PROC (Seite 611)	Callback-Funktion

Siehe auch

AP_RT_PROC (Seite 611)

APFreeResultStruct (Seite 626)

AP_ACT_KEY (Seite 599)

AP_ACT_RESULT_STRUCT (Seite 602)

Übersicht über die Funktionen (Seite 594)

2.4.6.6 APTransact (RT Professional)

Verwendung

Deklaration

Parameter

xxx

xxx

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

.

FALSE

Fehler.

Bemerkung

Benötigte Dateien

ap_def.h

apcli_S.lib

apclient.dll

Verwandte Funktionen

--	--

Beispiele

Auto-Hotspot "DM01.cpp"

Siehe auch

APConnect (Seite 605)

APDisconnect (Seite 607)

AP_RT_PROC (Seite 611)

APActive (Seite 622)

APEndAct (Seite 624)

APInactive (Seite 627)

AP_ACT_KEY (Seite 599)

Übersicht über die Funktionen (Seite 594)

2.4.7 Beispiele (RT Professional)

2.4.7.1 Establish connection to script programming (RT Professional)

Übersicht

```
//
=====
=
// Filename:..... ap01.c
//
=====
=
// : Modul with examples of AP_API
//
*****
*
// Copyright (C) 1995/96 SIEMENS AG, AUT 913 All rights reserved
//
*****
*
#include "stdafx.h"
#include "ap01.h" // if console application

//{{ODK_EXAMPLE}Establish connection to script programming (AP)}
//{{FUNCTION}APConnect (AP)}
//{{FUNCTION}APDisconnect (AP)}
//{{FUNCTION}AP_RT_PROC (AP)}
//{{FUNCTION}(END)}
// Establish connection to script programming (AP)
//
=====
=
// Function: AprConnect(void) ODK AP CS
//
=====
=
// short : Establish connection to script programming
// :
//
=====
=
BOOL MyAPRTCallback(DWORD dwAP_Notify, DWORD dwAP_NotifyCode, DWORD
dwError,
                    LPVOID lpvData, DWORD dwItems, DWORD dwOrderID, LPVOID
lpvUser)
{
    lpvUser;
    dwOrderID;
    lpvData;
    TCHAR szText[255];
}
```

```

        _sntprintf_s( szText, _countof(szText), _TRUNCATE,
        _T("AprNotCon:: AP= %d ;
                dwAP_Notify, dwAP_NotifyCode, dwError, dwItems);
        ODKTrace(szText);
        return(TRUE );
    }

void MyApConnect(void)
{
    TCHAR szText[255];
    CMN_ERROR Error;
    BOOL ret= FALSE;
    DWORD dwOrderID = 0;
    TCHAR szApp[255];
    VOID* pUser = NULL;
    _tcsncpy_s(szApp, _countof(szApp), _T("MyODKApp_23"),
    _TRUNCATE); // must be the same AppName as by DMConnect
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = APConnect(szApp, MyAPRTCallback, &dwOrderID, pUser, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error
in APConnect: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE,
        _T("APConnect"));
    }
    ODKTrace(szText);
    //printf("%s\r\n"szText);
    void MyAPDisconnect()
    VOID* pUser = NULL;
    CMN_ERROR Error;
    TCHAR szText[255];
    BOOL ret = FALSE;
    DWORD dwOrderID = 0;
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = APDisconnect(NULL, &dwOrderID, pUser, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error
in APDisconnect: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE,
        _T("APDisconnect"));
    }
    ODKTrace(szText);
}

```

```

}
//
-----
//{{ODK_EXAMPLE}}(END)}

```

Siehe auch

APConnect (Seite 605)
 APDisconnect (Seite 607)
 AP_RT_PROC (Seite 611)

2.5 Funktionen der Benutzerverwaltung (RT Professional)**2.5.1 Grundlagen (RT Professional)****2.5.1.1 Übersicht über die Funktionen (RT Professional)****Übersicht**

PWGEN_ENUM_GROUPS_CALLBACK (Seite 657)	Benutzergruppen auflisten (Callback)
PWGEN_ENUM_LEVELS_CALLBACK (Seite 663)	Berechtigungsstufen auflisten (Callback)
PWGEN_ENUM_USERS_CALLBACK (Seite 652)	Benutzer auflisten (Callback)
PWGENAddGroup (Seite 654)	Benutzergruppe anlegen
PWGENAddPermLevel (Seite 658)	Berechtigungsstufe anlegen
PWGENAddUser (Seite 643)	Benutzer anlegen
PWGENAddUserEx (Seite 645)	Benutzer anlegen
PWGENChangePassword (Seite 647)	Passwort ändern
PWGENCheckPermission (Seite 659)	Benutzerberechtigungen eines Benutzers prüfen
PWGENCheckUser (Seite 648)	Benutzer überprüfen
PWGENConnect (Seite 640)	Verbindung zur Datenbank aufnehmen
PWGENDeletePermLevel (Seite 661)	Berechtigungsstufe löschen
PWGENDeleteUser (Seite 649)	Benutzer löschen / Benutzergruppe löschen
PWGENDisconnect (Seite 642)	Verbindung zur Datenbank abbauen
PWGENEnumGroups (Seite 655)	Benutzergruppen auflisten
PWGENEnumPermLevels (Seite 662)	Berechtigungsstufen auflisten
PWGENEnumUsers (Seite 651)	Benutzer auflisten
PWGENReadUserPerm (Seite 665)	Benutzerberechtigungen ermitteln
PWRTCheckPermission (Seite 666)	Bereichsberechtigung überprüfen
PWRTCheckPermissionOnPicture (Seite 667)	Berechtigungsstufe überprüfen

2.5 Funktionen der Benutzerverwaltung (RT Professional)

PWRTGetCurrentUser (Seite 671)	Loginname ermitteln
PWRTGetLoginPriority (Seite 672)	GetLoginPriority
PWRTIsLoggedInByCard (Seite 673)	Login über Chipkarte
PWRTLogin (Seite 674)	Login über Dialog
PWRTLogout (Seite 676)	Logout
PWRTLogoutEx (Seite 677)	Logout
PWRTPermissionLevelDialog (Seite 668)	Berechtigungsstufe auswählen
PWRTPermissionLevelDialogEx (Seite 669)	Berechtigungsstufe auswählen
PWRTPermissionToString (Seite 670)	Berechtigungsstufe textuelle Beschreibung ermitteln
PWRTSilentLogin (Seite 678)	Login
PWRTSilentLoginEx (Seite 679)	Login

2.5.1.2 Übersicht über die Strukturen (RT Professional)

Übersicht

PWGEN_GROUPINFO (Seite 638)	Benutzergruppeninformationen
PWGEN_LEVELINFO (Seite 639)	Berechtigungsstufeninformationen
PWGEN_USERINFO (Seite 639)	Benutzerinformationen

2.5.1.3 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

Fehlermeldungen für RT:

PWRT_CAP_SHMERROR	1	Shared Memory Fehler
PWRT_CAP_NOLOGIN	2	Es ist kein Benutzer eingeloggt.
PWRT_CAP_NOPERM	3	Der Benutzer besitzt keine Berechtigung.
PWRT_CAP_NOAREA	4	Der Bereich konnte nicht gefunden werden.
PWRT_GPA_SHMERROR	1	Shared Memory Fehler
PWRT_GPA_NOLOGIN	2	Es ist kein Benutzer eingeloggt.

Fehlermeldungen für CS ab V5.0:

PWGEN_API_ERR_SUPPLY	0x10000001L	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_MEMORY	0x10000001L	Nicht genügend Speicher vorhanden.

2.5 Funktionen der Benutzerverwaltung (RT Professional)

PWGEN_API_NOT_SUPPORTED	0x10000003L	Diese Funktion wird nicht unterstützt.
PWGEN_API_INVALID_PARAM	0x10000004L	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_NO_INTERFACE	0x10000005L	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_I_ENUM	0x10000006L	Fehler beim Enumerieren
PWGEN_API_I_READ	0x10000007L	Fehler beim Lesen
PWGEN_API_I_CREATE	0x10000008L	Fehler beim Anlegen neuer Daten
PWGEN_API_I_MODIFY	0x10000009L	Fehler beim Ändern von Daten
PWGEN_API_I_DELETE	0x1000000AL	Fehler beim Löschen von Daten
PWGEN_API_TERMINATED	0x1000000FL	Die Bearbeitung wurde vorzeitig abgebrochen.
PWGEN_API_NO_CONNECTION	0x10000010L	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_IS_CONNECTED	0x10000011L	PWGENConnect wurde bereits durchgeführt.
PWGEN_API_NO_USER	0x10000020L	User nicht gefunden
PWGEN_API_MAX_USER	0x10000021L	Die maximale Anzahl User ist erreicht (~ 128).
PWGEN_API_EXIST_USER	0x10000022L	Der gewählte User existiert bereits.
PWGEN_API_NO_GROUP	0x10000030L	Benutzergruppe wurde nicht gefunden.
PWGEN_API_MAX_GROUP	0x10000031L	Die maximale Anzahl Benutzergruppen ist erreicht (~ 10).
PWGEN_API_EXIST_GROUP	0x10000032L	Die gewählte Benutzergruppe existiert bereits.
PWGEN_API_NO_LEVEL	0x10000040L	Berechtigungsstufe nicht gefunden.
PWGEN_API_MAX_LEVEL	0x10000041L	Die maximale Anzahl Berechtigungsstufen ist erreicht (~ 1000).
PWGEN_API_EXIST_LEVEL	0x10000042L	Die gewählte Berechtigungsstufe existiert bereits.
PWGEN_API_NO_PERM	0x10000050L	Keine Berechtigung vergeben.
PWGEN_API_MAX_PERM	0x10000051L	
PWGEN_API_EXIST_PERM	0x10000052L	Die Berechtigung existiert bereits.
PWGEN_API_NO_AREA	0x10000060L	Kein Bereich wurde vergeben.
PWGEN_API_MAX_AREA	0x10000061L	Die maximale Anzahl von Bereichen ist erreicht (~ 32).
PWGEN_API_EXIST_AREA	0x10000062L	Der Bereich existiert bereits.
PWGEN_API_NO_PASSWORD	0x10000070L	Kein oder kein gültiges Passwort
PWGEN_API_MAX_PASSWORD	0x10000071L	
PWGEN_API_EXIST_PASSWORD	0x10000072L	

2.5.1.4 Konstanten (RT Professional)

String- und Arraygrößen

MAX_LOGIN	25
MAX_PASS	25
MAX_LEVEL	70

2.5.2 Strukturen (RT Professional)

2.5.2.1 PWGEN_GROUPINFO (RT Professional)

Deklaration

```
typedef struct {  
    TCHAR    name[MAX_LOGIN];  
    int      expiration_time;  
}  
PWGEN_GROUPINFO;
```

Members

name

Name der Benutzergruppe

expiration_time

automatische Logoutzeit in Minuten

Benötigte Dateien

usegenap.h

API - Funktionen

PWGEN_ENUM_GROUPS_CALLBACK (Seite 657)	Benutzergruppen auflisten
--	---------------------------

Siehe auch

PWGEN_ENUM_GROUPS_CALLBACK (Seite 657)

2.5.2.2 PWGEN_LEVELINFO (RT Professional)

Deklaration

```
typedef struct {
    int         levelNumber;
    DWORD      dwTextID;
}
PWGEN_LEVELINFO;
```

Members

levelNumber

Nummer der Berechtigung

dwTextID

ID, mit der die Beschreibung der Berechtigung identifiziert wird.

Benötigte Dateien

usegenap.h

API - Funktionen

PWGEN_ENUM_LEVELS_CALLBACK (Seite 663)	Berechtigungsstufen auflisten
--	-------------------------------

Siehe auch

PWGEN_ENUM_LEVELS_CALLBACK (Seite 663)

2.5.2.3 PWGEN_USERINFO (RT Professional)

Deklaration

```
typedef struct {
    TCHAR      login[MAX_LOGIN];
    TCHAR      group[MAX_LOGIN];
    int        expiration_time;
}
PWGEN_USERINFO;
```

Members

login

Loginname des Benutzers

group

Name der Gruppe, der der Benutzer angehört

expiration_time

automatische Logoutzeit in Minuten

Benötigte Files

usegenap.h

API Funktionen

PWGEN_ENUM_USERS_CALLBACK (Seite 652)	Benutzer auflisten
---------------------------------------	--------------------

Siehe auch

PWGEN_ENUM_USERS_CALLBACK (Seite 652)

2.5.3 Allgemeine Funktionen (RT Professional)

2.5.3.1 PWGENConnect (RT Professional)

Beschreibung

Die Funktion nimmt Verbindung zur Datenbank des aktuell geöffneten Projekts auf.

Deklaration

```
BOOL PWGENConnect (
    LPCTSTR      DSNName,
    LPCMN_ERROR  err)
```

Parameter

DSNName

DataSource-Name des Projekts. Der Parameter DSNName wird nicht ausgewertet. Es kann ein Leerstring übergeben werden, aber nicht NULL. Es wird immer die Datenbank des aktuell geöffneten Projektes verwendet.

err

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung aufgenommen

FALSE

Fehler

Bemerkung

Jede Applikation kann nur ein Connect durchführen. Bei einem wiederholten Connect wird die Fehlermeldung PWGEN_API_IS_CONNECTED zurückgegeben.

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_IS_CONNECTED	PWGENConnect wurde bereits durchgeführt.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGENDisconnect (Seite 642)	Verbindung zur Datenbank abbauen
-----------------------------	----------------------------------

Siehe auch

PWGENDisconnect (Seite 642)

Übersicht über die Funktionen (Seite 635)

2.5.3.2 PWGENDisconnect (RT Professional)

Beschreibung

Die Verbindung zur Datenbank des aktuell geöffneten Projekts wird wieder abgebaut.

Deklaration

```
BOOL PWGENDisconnect (  
    LPCMN_ERROR    err)
```

Parameter

err

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung abgebaut

FALSE

Fehler

Bemerkung

Vor Beenden der Applikation muss ein PWGENDisconnect durchgeführt werden, da sonst die intern verwendeten UserAdminASO-Schnittstellen nicht mehr freigegeben werden.

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX, ...) benutzt werden. Aufgrund von Microsoft-spezifischen Mechanismen kann das gegebenenfalls zum Hängenbleiben des Aufrufs und damit des Programms führen.

Fehlermeldungen

PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGENConnect (Seite 640)	Verbindung zur Datenbank aufnehmen.
--------------------------	-------------------------------------

Siehe auch

PWGENConnect (Seite 640)

Übersicht über die Funktionen (Seite 635)

2.5.4 Funktionen zum Bearbeiten von Benutzern (RT Professional)

2.5.4.1 PWGENAddUser (RT Professional)

Beschreibung

Die Funktion legt einen neuen Benutzer einschließlich einer leeren Benutzerberechtigungsmatrix an. Mit der Funktion PWGENAddUserEx können Sie die Berechtigungen der Gruppe auf den anzulegenden Benutzer übertragen.

Deklaration

```

BOOL PWGENAddUser (
    LPCTSTR    username,
    LPCTSTR    password,
    LPCTSTR    group,
    int        expiration_time,
    LPCMN_ERROR error )

```

Parameter

username

Loginname des neuen Benutzers

password

Kennwort des Benutzers

group

Gruppe, zu der der Benutzer hinzugefügt werden soll

expiration_time

Automatische Logoutzeit in Minuten. Liegt die gewählte expiration_time außerhalb des zulässigen Bereichs, so wird sie mit 0 belegt.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benutzer angelegt

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_EXIST_USER	Der gewählte User existiert bereits.
PWGEN_API_MAX_USER	Die maximale Anzahl User ist erreicht (~ 128).
PWGEN_API_NO_GROUP	Benutzergruppe nicht gefunden.
PWGEN_API_I_CREATE	Fehler beim Anlegen neuer Daten.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGENAddUserEx (Seite 645)	Benutzer anlegen
----------------------------	------------------

Siehe auch

PWGENAddUserEx (Seite 645)

Übersicht über die Funktionen (Seite 635)

2.5.4.2 PWGENAddUserEx (RT Professional)

Beschreibung

Die Funktion legt einen neuen Benutzer an. In Abhängigkeit des Parameters `copy_group_protection` können die Berechtigungen der Gruppe auf den anzulegenden Benutzer übertragen werden.

Deklaration

```
BOOL PWGENAddUserEx (  
    LPCTSTR      username,  
    LPCTSTR      password,  
    LPCTSTR      group,  
    int          expiration_time,  
    BOOL         copy_group_permissions,  
    LPCMN_ERROR  error )
```

Parameter

username

Loginname des neuen Benutzers

password

Kennwort des Benutzers

group

Gruppe, zu der der Benutzer hinzugefügt werden soll

expiration_time

Automatische Logoutzeit in Minuten. Liegt die gewählte `expiration_time` außerhalb des zulässigen Bereichs, so wird sie mit 0 belegt.

copy_group_permissions

Ist `copy_group_permission = TRUE`, so werden die Benutzerberechtigungen der Gruppe auf den neu anzulegenden Benutzer übertragen.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benutzer angelegt

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_EXIST_USER	Der gewählte User existiert bereits.
PWGEN_API_MAX_USER	Die maximale Anzahl User ist erreicht (~ 128).
PWGEN_API_NO_GROUP	Benutzergruppe nicht gefunden.
PWGEN_API_I_CREATE	Fehler beim Anlegen neuer Daten.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGENAddUser (Seite 643)	Benutzer anlegen
--------------------------	------------------

Siehe auch

PWGENAddUser (Seite 643)

Übersicht über die Funktionen (Seite 635)

2.5.4.3 PWGENChangePassword (RT Professional)

Beschreibung

Mit der Funktion kann das Passwort des über username spezifizierten Benutzers geändert werden.

Deklaration

```

BOOL PWGENChangePassword (
    LPCTSTR      username,
    LPCTSTR      oldpassword,
    LPCTSTR      newpassword,
    LPCMN_ERROR  error )

```

Parameter

username

Loginname des Benutzers

oldpassword

altes Kennwort des Benutzers

newpassword

neues Kennwort des Benutzers

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Passwort geändert

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_NO_PASSWORD	Kein oder kein gültiges Passwort.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.

PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_USER	User nicht gefunden.
PWGEN_API_I_READ	Fehler beim Lesen.
PWGEN_API_I_MODIFY	Fehler beim Ändern von Daten.

Benötigte Dateien

usegenap.h
usegen.lib
usegen.dll

Siehe auch

Übersicht über die Funktionen (Seite 635)

2.5.4.4 PWGENCheckUser (RT Professional)

Beschreibung

Es wird geprüft, ob der angegebene Benutzer im aktuellen WinCC-Projekt existiert und das übergebene Passwort korrekt ist.

Deklaration

```
BOOL PWGENCheckUser (  
    LPCTSTR      username,  
    LPCTSTR      password,  
    LPCMN_ERROR  error )
```

Parameter

username

Name des Benutzers.

password

Das Kennwort, das zum angegebenen Benutzer gehört.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benutzer existiert und das Passwort ist korrekt.

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_USER	User nicht gefunden.
PWGEN_API_NO_PASSWORD	Kein oder kein gültiges Passwort.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Siehe auch

Übersicht über die Funktionen (Seite 635)

2.5.4.5 PWGENDeleteUser (RT Professional)

Beschreibung

Der angegebene Benutzer bzw. die Benutzergruppe wird einschließlich der Berechtigungsdaten gelöscht.

Deklaration

```

BOOL PWGENDeleteUser (
    LPCTSTR      username,
    BOOL         is_user,
    LPCMN_ERROR  error )

```

Parameter

username

Name des zu löschenden Benutzers.

is_user

Der Parameter is_user dient der Unterscheidung, ob ein Benutzer oder eine Benutzergruppe gelöscht werden soll.

TRUE	Es soll ein Benutzer gelöscht werden.
FALSE	Es soll eine Benutzergruppe gelöscht werden.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benutzer bzw. Benutzergruppe gelöscht

FALSE

Fehler

Bemerkung

Wenn versucht wird den Administrator zu löschen, so liefert die Funktion den Fehler PWGEN_API_EXIST_USER.

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_I_DELETE	Fehler beim Löschen von Daten.
PWGEN_API_NO_USER	User nicht gefunden.
PWGEN_API_NO_GROUP	Benutzergruppe nicht gefunden.
PWGEN_API_EXIST_USER	Der gewählte User existiert bereits.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Siehe auch

Übersicht über die Funktionen (Seite 635)

2.5.4.6 PWGENEnumUsers (RT Professional)

Beschreibung

Die Funktion liest die projektierten Benutzer ein, ruft für jeden Benutzer die Callback-Funktion auf und liefert die Anzahl der Benutzer in dwCount.

Deklaration

```
BOOL PWGENEnumUsers (
    LPDWORD                dwCount,
    PWGEN_ENUM_USERS_CALLBACK cfn,
    PVOID                  userdata,
    LPCMN_ERROR            error )
```

Parameter

dwCount

Zeiger auf die Speicherstelle, an der die Anzahl der Benutzer abgelegt werden soll.

cfn

Ihre Callback-Funktion, die die Informationen entgegennimmt. Wenn cfn == NULL, dann werden die Benutzer nur gezählt.

userdata

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benutzer aufgelistet

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGEN_ENUM_USERS_CALLBACK (Seite 652)	Benutzer auflisten (Callback)
---------------------------------------	-------------------------------

Siehe auch

PWGEN_ENUM_USERS_CALLBACK (Seite 652)

Übersicht über die Funktionen (Seite 635)

2.5.4.7 PWGEN_ENUM_USERS_CALLBACK (RT Professional)

Beschreibung

Um die vom System aufgelisteten Benutzerinformationen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ PWGEN_ENUM_USERS_CALLBACK bereitstellen.

Deklaration

```
BOOL ( * PWGEN_ENUM_USERS_CALLBACK) (
    LPWGEN_USERINFO      lpUserInfo,
    PVOID                lpUser);
```

Parameter

IpUserInfo

Zeiger auf eine Struktur vom Typ PWGEN_USERINFO (Seite 639) mit den Daten eines Benutzers.

IpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Benutzer aufgelistet

FALSE

Fehler

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

usegenap.h

Verwandte Funktionen

PWGENEnumUsers (Seite 651)	Benutzer auflisten
----------------------------	--------------------

Siehe auch

PWGENEnumUsers (Seite 651)

PWGEN_USERINFO (Seite 639)

Übersicht über die Funktionen (Seite 635)

2.5.5 Funktionen zum Bearbeiten von Benutzergruppen (RT Professional)

2.5.5.1 PWGENAddGroup (RT Professional)

Beschreibung

Die Funktion legt eine neue Benutzergruppe einschließlich einer leeren Benutzerberechtigungsmatrix an.

Deklaration

```
BOOL PWGENAddGroup (  
    LPCTSTR      username,  
    int          expiration_time,  
    LPCMN_ERROR  error )
```

Parameter

username

Name der anzulegenden Gruppe

expiration_time

Automatische Logoutzeit. Liegt die gewählte expiration_time außerhalb des zulässigen Bereichs so wird sie mit 0 belegt.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benutzergruppe angelegt

FALSE

Benutzergruppe existiert bereits

Bemerkung

Um eine bestehende Benutzergruppe zu löschen, verwenden Sie die Funktion PWGENDeleteUser.

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_EXIST_GROUP	Die gewählte Benutzergruppe existiert bereits.
PWGEN_API_MAX_GROUP	Die maximale Anzahl Benutzergruppen ist erreicht (~ 10).
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.
PWGEN_API_I_CREATE	Fehler beim Anlegen neuer Daten.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGENDeleteUser (Seite 649)	Benutzer anlegen
-----------------------------	------------------

Siehe auch

PWGENDeleteUser (Seite 649)

Übersicht über die Funktionen (Seite 635)

2.5.5.2 PWGENEnumGroups (RT Professional)

Beschreibung

Die Funktion liest die projizierten Benutzergruppen ein, ruft für jede Gruppe die Callback-Funktion auf und liefert die Anzahl der Gruppen in dwCount.

Deklaration

```

BOOL PWGENEnumGroups (
    LPDWORD                dwCount,
    PWGEN_ENUM_GROUPS_CALLBACK cfn,
    PVOID                  userdata,
    LPCMN_ERROR            error )

```

Parameter

dwCount

Zeiger auf die Speicherstelle, an der die Anzahl der Benutzergruppen abgelegt werden soll.

cfn

Ihre Callback-Funktion, die die Informationen entgegennimmt. Wenn cfn == NULL ist, dann werden die Benutzergruppen nur gezählt.

userdata

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Benutzergruppen aufgelistet

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGEN_ENUM_GROUPS_CALLBACK (Seite 657)	Benutzergruppen auflisten (Callback)
--	--------------------------------------

Siehe auch

PWGEN_ENUM_GROUPS_CALLBACK (Seite 657)

Übersicht über die Funktionen (Seite 635)

2.5.5.3 PWGEN_ENUM_GROUPS_CALLBACK (RT Professional)**Beschreibung**

Um die vom System aufgelisteten Informationen von Benutzergruppen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ PWGEN_ENUM_GROUPS_CALLBACK bereitstellen.

Deklaration

```
BOOL ( * PWGEN_ENUM_GROUPS_CALLBACK) (  
    LPWGEN_GROUPINFO    lpGroupInfo,  
    PVOID                lpUser);
```

Parameter**lpUserInfo**

Zeiger auf eine Struktur vom Typ PWGEN_GROUPINFO (Seite 638) mit den Daten einer Benutzergruppe.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert**TRUE**

Benutzergruppen aufgelistet

FALSE

Fehler

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

usegenap.h

Verwandte Funktionen

PWGENEnumGroups (Seite 655)	Benutzergruppen auflisten
-----------------------------	---------------------------

Siehe auch

PWGENEnumGroups (Seite 655)

PWGEN_GROUPINFO (Seite 638)

Übersicht über die Funktionen (Seite 635)

2.5.6 Funktionen zum Bearbeiten von Berechtigungen (RT Professional)

2.5.6.1 PWGENAddPermLevel (RT Professional)

Beschreibung

Legt für alle Benutzer eine neue Berechtigung an.

Deklaration

```
BOOL PWGENAddPermLevel (  
    DWORD          txtID,  
    int            number,  
    LPCMN_ERROR    error )
```

Parameter

txtID

Text-ID des Texts für die Berechtigung.

number

Nummer der anzulegenden Berechtigung

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Berechtigungsstufe angelegt

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.
PWGEN_API_MAX_LEVEL	Die maximale Anzahl Berechtigungsstufen (1000) ist erreicht .
PWGEN_API_EXIST_LEVEL	Die gewählte Berechtigungsstufe existiert bereits.
PWGEN_API_I_CREATE	Fehler beim Anlegen neuer Daten.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Siehe auch

Übersicht über die Funktionen (Seite 635)

2.5.6.2 PWGENCheckPermission (RT Professional)**Beschreibung**

Mit der Funktion wird geprüft, ob der Benutzer zur angegebenen Berechtigungsstufe die Bedienberechtigungen besitzt.

Deklaration

```

BOOL PWGENCheckPermission(
    LPCTSTR          username,
    DWORD            permlevel,
    LPCMN_ERROR      error)

```

Parameter

username

Name des Benutzers

permlevel

Nummer der zu prüfenden Berechtigung

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Bedienberechtigung zur Berechtigungsstufe des Benutzers liegt vor.

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.
PWGEN_API_NO_LEVEL	Berechtigungsstufe nicht gefunden.
PWGEN_API_I_READ	Fehler beim Lesen.
PWGEN_API_NO_PERM	Keine Berechtigung vergeben.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Siehe auch

Übersicht über die Funktionen (Seite 635)

2.5.6.3 PWGENDeletePermLevel (RT Professional)

Beschreibung

Löscht aus den Benutzerberechtigungsmatrizen die über levelNumber spezifizierte Berechtigungsstufe.

Deklaration

```
BOOL PWGENDeletePermLevel (
    int          levelNumber,
    LPCMN_ERROR  error )
```

Parameter

levelNumber

Nummer der zu löschenden Berechtigung

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Berechtigungsstufe gelöscht

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.
PWGEN_API_NO_LEVEL	Berechtigungsstufe nicht gefunden.
PWGEN_API_I_DELETE	Fehler beim Löschen von Daten.

Benötigte Dateien

usegenap.h
usegen.lib
usegen.dll

Siehe auch

Übersicht über die Funktionen (Seite 635)

2.5.6.4 PWGENEnumPermLevels (RT Professional)

Beschreibung

Die Funktion liest die projektierten Berechtigungsstufen ein, ruft für jede Stufe die Callback-Funktion auf und liefert die Anzahl der Berechtigungsstufen in dwCount

Deklaration

```
BOOL PWGENEnumPermLevels (
    LPDWORD                dwCount,
    PWGEN_ENUM_LEVELS_CALLBACK cfn,
    PVOID                  userdata,
    LPCMN_ERROR            error )
```

Parameter

dwCount

Zeiger auf die Speicherstelle, an der die Anzahl der Berechtigungsstufen abgelegt wird.

cfn

Ihre Callback-Funktion, die die Informationen entgegennimmt. Wenn cfn == NULL ist, dann werden die Berechtigungen nur gezählt.

userdata

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

error

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Berechtigungsstufen aufgelistet

FALSE

Fehler

Fehlermeldungen

PWGEN_API_INVALID_PARAM	Ein Funktionsparameter stimmt nicht oder fehlt.
PWGEN_API_ERR_SUPPLY	Ein nicht näher spezifizierter Fehler ist aufgetreten.
PWGEN_API_NO_INTERFACE	Keine Verbindung zum UserAdminASO-Interface.
PWGEN_API_NO_CONNECTION	Es ist noch kein PWGENConnect erfolgt.
PWGEN_API_I_ENUM	Fehler beim Enumerieren.
PWGEN_API_NO_MEMORY	Nicht genügend Speicher vorhanden.

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Verwandte Funktionen

PWGEN_ENUM_LEVELS_CALLBACK (Seite 663)	Berechtigungen auflisten (Callback)
--	-------------------------------------

Siehe auch

PWGEN_ENUM_LEVELS_CALLBACK (Seite 663)

Übersicht über die Funktionen (Seite 635)

2.5.6.5 PWGEN_ENUM_LEVELS_CALLBACK (RT Professional)**Beschreibung**

Um die vom System aufgelisteten Berechtigungsstufen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ PWGEN_ENUM_LEVELS_CALLBACK bereitstellen.

Deklaration

```
BOOL ( * PWGEN_ENUM_LEVELS_CALLBACK) (  
    LPWGEN_LEVELINFO    lpLevelInfo,  
    PVOID                lpUser);
```

Parameter

lpLevelInfo

Zeiger auf eine Struktur vom Typ PWGEN_LEVELINFO (Seite 639) mit den Daten der Berechtigungsstufe.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Berechtigungsstufen aufgelistet

FALSE

Fehler

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

usegenap.h

Verwandte Funktionen

PWGENEnumPermLevels (Seite 662)	Berechtigungsstufen auflisten
---------------------------------	-------------------------------

Siehe auch

PWGENEnumPermLevels (Seite 662)

PWGEN_LEVELINFO (Seite 639)

Übersicht über die Funktionen (Seite 635)

2.5.6.6 PWGENReadUserPerm (RT Professional)**Beschreibung**

Mit der Funktion werden die Bereiche ermittelt, für die ein Benutzer auf einer bestimmten Berechtigungsstufe Bedienberechtigungen besitzt.

Deklaration

```

BOOL PWGENReadUserPerm(
    LPCTSTR    username,
    BOOL       is_user,
    int        levelnumber,
    LPBYTE     freigabe,
    LPDWORD    areaperms )

```

Parameter**username**

Name des Benutzers bzw. der Benutzergruppe

is_user

is_user kennzeichnet ob es sich bei username um einen Benutzer oder eine Benutzergruppe handelt.

TRUE	Benutzer
FALSE	Benutzergruppe

levelnumber

Nummer der zu lesenden Berechtigung

freigabe

Ist freigabe = 1, so besitzt der Benutzer auf dieser Berechtigungsstufe die Bedienberechtigung für alle Bereiche.

areaperms

Über die 32 Bit von areaperm werden die Bereiche charakterisiert. Besitzt ein Benutzer die Berechtigung für einen Bereich, so wird das entsprechende Bit gesetzt. Dabei entspricht das niederwertigste Bit dem ersten Bereich.

Rückgabewert

TRUE

Bedienberechtigungen ermittelt

FALSE

Fehler

Benötigte Dateien

usegenap.h

usegen.lib

usegen.dll

Siehe auch

Übersicht über die Funktionen (Seite 635)

2.5.6.7 PWRTCheckPermission (RT Professional)

Beschreibung

Überprüft, ob der angemeldete Benutzer die Bedienberechtigung für die übergebene Berechtigung hat.

Deklaration

```
BOOL PWRTCheckPermission (  
    DWORD    permlevel,  
    DWORD    suppress_messagebox )
```

Parameter

permlevel

Nummer der zu überprüfenden Berechtigungsstufe.

suppress_messagebox

Bei suppress_messagebox != 0 wird kein Dialog angezeigt.

Rückgabewert

TRUE

Berechtigung erteilt

FALSE

Berechtigung nicht erteilt

Benötigte Dateien

pwrt_api.h
pass_s.lib
useadmin.dll

Beispiele

PWRT check permission (Seite 681) "PWRTBunch.cpp"

Siehe auch

PWRT check permission (Seite 681)
Übersicht über die Funktionen (Seite 635)

2.5.6.8 PWRTCheckPermissionOnPicture (RT Professional)

Beschreibung

Überprüft, ob der angemeldete Benutzer die Bedienberechtigung für die übergebene Berechtigung hat.

Deklaration

```
BOOL PWRTCheckPermissionOnPicture (  
    DWORD          permlevel,  
    LPCTSTR        picture_name,  
    DWORD          suppress_messagebox,  
    LPCMN_ERROR    lperr )
```

Parameter

permlevel

Nummer der zu überprüfenden Berechtigung.

picture_name

Name des Bildes, in dem das zu testende Objekt liegt.

suppress_messagebox

Bei `suppress_messagebox != 0` wird kein Dialog angezeigt.

lperr

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Berechtigung erteilt

FALSE

Berechtigung nicht erteilt

Benötigte Dateien

pwrt_api.h

pass_s.lib

useadmin.dll

Beispiele

Checks admission of a certain level for a picture (Seite 682) "PWRTBunch.cpp"

Siehe auch

Checks admission of a certain level for a picture (Seite 682)

Übersicht über die Funktionen (Seite 635)

2.5.6.9 PWRTPermissionLevelDialog (RT Professional)

Beschreibung

Sie sollten den Dialog zur Auswahl der Berechtigung über die Funktion PWRTPermissionLevelDialogEx öffnen, da dann beim Bewegen des Dialogfensters ein Redraw durchgeführt wird.

Deklaration

```
LONG PWRTPermissionLevelDialog (  
    )
```

Parameter

Keine

Verwandte Funktionen

PWRTPermissionLevelDialogEx (Seite 669)	Berechtigung über Dialog auswählen
---	------------------------------------

Beispiele

Permission level query through a dialog with specifying a possible error (Seite 684)
"PWRTBunch.cpp"

Siehe auch

Permission level query through a dialog with specifying a possible error (Seite 684)
PWRTPermissionLevelDialogEx (Seite 669)
PWRTPermissionToString (Seite 670)
Übersicht über die Funktionen (Seite 635)

2.5.6.10 PWRTPermissionLevelDialogEx (RT Professional)

Beschreibung

Das Schützen der Bedienbarkeit von Objekten erfolgt in WinCC durch einen Dialog zur Auswahl einer Berechtigung. PWRTPermissionLevelDialogEx blendet einen entsprechenden Dialog auf, der die vorhandenen Berechtigungen zur Auswahl anbietet.

Deklaration

```
LONG PWRTPermissionLevelDialogEx (
    HWND          hParentWnd,
    CMN_ERROR     *lpErr)
```

Parameter

hParentWnd

Handle auf das Vaterfenster des Dialogs.

lpErr

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

0-999: Berechtigungsstufe

-1: Dialog wurde mit Cancel beendet.

Benötigte Dateien

pwrt_api.h
pass_s.lib
useadmin.dll

Beispiel

Permission level query through a dialog with specifying a possible error (Seite 684)
"PWRTBunch.cpp"

Siehe auch

PWRTPermissionLevelDialog (Seite 668)
Permission level query through a dialog with specifying a possible error (Seite 684)
Übersicht über die Funktionen (Seite 635)

2.5.6.11 PWRTPermissionToString (RT Professional)

Beschreibung

Ermittelt die zu einer Berechtigungsstufe gehörende Beschreibung.

Deklaration

```
BOOL PWRTPermissionToString (  
    LONG      perm,  
    LPTSTR    string,  
    int       bufsize)
```

Parameter

perm

Berechtigungsstufe, deren Beschreibung ermittelt werden soll.

string

Zeiger auf den Puffer für die Aufnahme der Beschreibung

bufsize

Größe des Puffers

Rückgabewert

TRUE

Beschreibung erfolgreich übertragen.

FALSE

Beschreibung konnte nicht ermittelt werden.

Benötigte Dateien

pwrt_api.h

pass_s.lib

useadmin.dll

Verwandte Funktionen

PWRTPermissionLevelDialog (Seite 668)	Berechtigungsstufe auswählen
---------------------------------------	------------------------------

Beispiele

Gets a string associated with the permission number (Seite 683). "PWRTBunch.cpp"

Siehe auch

PWRTPermissionLevelDialog (Seite 668)

Gets a string associated with the permission number (Seite 683)

Übersicht über die Funktionen (Seite 635)

2.5.7 Funktionen zum Anmelden, Abmelden (RT Professional)

2.5.7.1 PWRTGetCurrentUser (RT Professional)

Beschreibung

Ermittelt den Benutzernamen des aktuellen angemeldeten Benutzers.

Deklaration

```
BOOL PWRTGetCurrentUser (
    LPTSTR    dest,
    int       bufsize )
```

Parameter

dest

Datenpuffer zum Empfang des Benutzernamens

bufsize

Größe des Datenpuffers in Byte

Rückgabewert

TRUE

Benutzername erfolgreich übertragen

FALSE

Es ist kein Benutzer angemeldet

Benötigte Dateien

pwrt_api.h

pass_s.lib

useadmin.dll

Beispiele

Return the name of the current user (Seite 685) "PWRTBunch.cpp"

Siehe auch

Return the name of the current user (Seite 685)

Übersicht über die Funktionen (Seite 635)

2.5.7.2 PWRTGetLoginPriority (RT Professional)

Beschreibung

Loginprüfung ohne Dialog.

Deklaration

```
LONG PWRTGetLoginPriority (  
    )
```

Parameter

Keine

Rückgabewert

Prioritätslevel:

LOGIN_STANDARD	(Wert: 0)
LOGIN_CARD	(Wert: 1)
LOGIN_KEYSWITCH	(Wert: 2)

Benötigte Dateien

pwr_api.h
pass_s.lib
useadmin.dll

Beispiele

Queries the current login priority (Seite 686) "PWRTBunch.cpp"

Siehe auch

Queries the current login priority (Seite 686)
Übersicht über die Funktionen (Seite 635)

2.5.7.3 PWRTIsLoggedInByCard (RT Professional)**Beschreibung**

Ermittelt, ob der Benutzer mittels Chipkarte angemeldet ist.

Deklaration

```
BOOL PWRTIsLoggedInByCard (  
    )
```

Parameter

keine

Rückgabewert

TRUE

Benutzer mit Chipkarte eingeloggt

FALSE

Benutzer nicht mit Chipkarte oder gar nicht angemeldet

Benötigte Dateien

pwr_api.h

pass_s.lib

useadmin.dll

Beispiele

Checks for the user has been logged on by card (Seite 686) "PWRTBunch.cpp"

Siehe auch

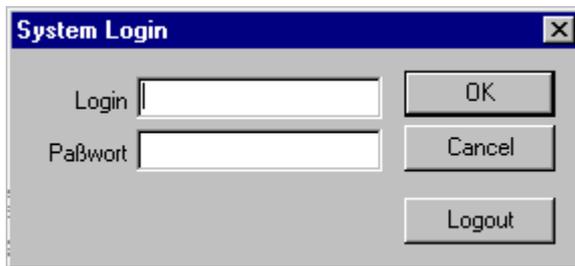
Checks for the user has been logged on by card (Seite 686)

Übersicht über die Funktionen (Seite 635)

2.5.7.4 PWRTLogin (RT Professional)

Beschreibung

Blendet den Login Dialog auf und lädt bei erfolgreichem Login die Benutzerdaten in das Shared Memory.



Deklaration

```
BOOL PWRTLogin (  
    TCHAR    monitor )
```

Parameter

monitor

Bildschirm, auf dem der Dialog aufgeblendet wird. Der Wert wird nicht numerisch, sondern im TCHAR-Format, für Bildschirm 1 also '1', angegeben.

Rückgabewert

TRUE

Dialog geöffnet. Die Funktion wartet nicht auf das Beenden des Dialogs.

FALSE

Dialog nicht geöffnet

Bemerkung

Neben der manuellen Eingabe der Login-Daten besteht auch die Möglichkeit sich mit einer Chipkarte einzuloggen. Der Kartenleser ist direkt an der OS angeschlossen. Ist bei der Ausführung der Funktion eine gültige Karte angeschlossen, wird der Login-Dialog automatisch ausgeblendet.

Benötigte Dateien

pwrt_api.h

pass_s.lib

useadmin.dll

Verwandte Funktionen

PWRTSilentLogin (Seite 678)	Login
-----------------------------	-------

Beispiele

PWRT login - dialog provided by WinCC itself (Seite 687) "PWRTBunch.cpp"

Siehe auch

PWRTSilentLogin (Seite 678)

PWRT login - dialog provided by WinCC itself (Seite 687)

PWRTSilentLoginEx (Seite 679)

Übersicht über die Funktionen (Seite 635)

2.5.7.5 PWRTLogout (RT Professional)

Beschreibung

Die Funktion bewirkt, dass im Shared Memory ein Flag auf abgemeldet gesetzt wird.

Deklaration

```
BOOL PWRTLogout (  
    )
```

Parameter

Keine

Rückgabewert

TRUE

Logout erfolgreich an PassDBRT weitergegeben.

FALSE

Logout abgewiesen (z.B. PassDBRT nicht verfügbar, ServiceMode, ...).

Benötigte Dateien

pwrt_api.h

pass_s.lib

useadmin.dll

Verwandte Funktionen

PWRTLogoutEx (Seite 677)	Benutzer abmelden mit Prioritätslevel
--------------------------	---------------------------------------

Beispiele

PWRT logout (Seite 688) "PWRTBunch.cpp"

Siehe auch

PWRT logout (Seite 688)

PWRTLogoutEx (Seite 677)

Übersicht über die Funktionen (Seite 635)

2.5.7.6 PWRTLogoutEx (RT Professional)

Beschreibung

Die Funktion bewirkt, dass im Shared Memory ein Flag auf abgemeldet gesetzt wird, sofern der angegebene Prioritätslevel nicht zu niedrig ist.

Deklaration

```
BOOL PWRTLogoutEx (  
    int    nLevel  
)
```

Parameter

nLevel

Prioritätslevel des Benutzers.

Rückgabewert

TRUE

Logout erfolgreich an PassDBRT weitergegeben.

FALSE

Logout abgewiesen (z.B. PassDBRT nicht verfügbar, ServiceMode, ...).

Benötigte Dateien

pwrt_api.h

pass_s.lib

useadmin.dll

Verwandte Funktionen

PWRTLogout (Seite 676)	Benutzer abmelden
------------------------	-------------------

Beispiele

Silent logout with priority level (Seite 689) "PWRTBunch.cpp"

Siehe auch

- Silent logout with priority level (Seite 689)
- PWRTLogout (Seite 676)
- Übersicht über die Funktionen (Seite 635)

2.5.7.7 PWRTSilentLogin (RT Professional)

Beschreibung

Im Unterschied zu PWRTLogin erfolgt das Anmelden nicht über einen Dialog. Die Daten des Benutzers, Benutzername und Kennwort, werden direkt an die Funktion übergeben.

Deklaration

```
BOOL PWRTSilentLogin (  
    LPCTSTR    login,  
    LPCTSTR    password )
```

Parameter

login

Name des Benutzers.

password

Kennwort des Benutzers

Rückgabewert

TRUE

Login erfolgreich

FALSE

Login abgewiesen

Bemerkung

Ist bei der Ausführung der Funktion eine gültige Chipkarte angeschlossen, so wird der Login abgewiesen.

Benötigte Dateien

pwrt_api.h

pass_s.lib
useadmin.dll

Verwandte Funktionen

PWRTLogin (Seite 674)	Login über Dialog
PWRTSilentLoginEx (Seite 679)	Login mit Prioritätslevel

Beispiele

Login without using a dialog (Seite 690) "PWRTBunch.cpp"

Siehe auch

PWRTLogin (Seite 674)
PWRTSilentLoginEx (Seite 679)
Login without using a dialog (Seite 690)
Übersicht über die Funktionen (Seite 635)

2.5.7.8 PWRTSilentLoginEx (RT Professional)

Beschreibung

Im Unterschied zu PWRTLogin erfolgt das Anmelden nicht über einen Dialog. Die Daten des Benutzers (Benutzername, Kennwort und Prioritätslevel) werden direkt an die Funktion übergeben.

Deklaration

```
BOOL PWRTSilentLoginEx (  
    LPCTSTR    login,  
    LPCTSTR    password,  
    int        nLevel)
```

Parameter

login

Name des Benutzers.

password

Kennwort des Benutzers

nLevel

Prioritätslevel des Benutzers

Rückgabewert

TRUE

Login erfolgreich

FALSE

Login abgewiesen

Bemerkung

Ist bei der Ausführung der Funktion eine gültige Chipkarte angeschlossen, so wird der Login abgewiesen.

Benötigte Dateien

pwrt_api.h

pass_s.lib

useadmin.dll

Verwandte Funktionen

PWRTLogin (Seite 674)	Login über Dialog
PWRTSilentLogin (Seite 678)	Login ohne Dialog

Beispiele

Silent login with priority level (Seite 691) "PWRTBunch.cpp"

Siehe auch

PWRTSilentLogin (Seite 678)

Silent login with priority level (Seite 691)

PWRTLogin (Seite 674)

Übersicht über die Funktionen (Seite 635)

2.5.8 Beispiele (RT Professional)

2.5.8.1 PWRT check permission (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}PWRT check permission. (USE)}
//{{FUNCTION}PWRTCheckPermission (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::Pwrtcheckpermission()
{
    BOOL bRet;
    CString csOut;
    CString csPermLevName;
    CGetText l_PermissionLevel("Insert the permission level number:");
    if(l_PermissionLevel.DoModal()==IDOK)
    {
        ////////////////////////////////////////////////////////////////////
        ////////////////////////////////////////////////////////////////////
        bRet = PWRTCheckPermission( l_PermissionLevel.m_lNumber, 0L ); // 0 = show the
message box
        ////////////////////////////////////////////////////////////////////
        ////////////////////////////////////////////////////////////////////
        bRet &= PWRTPermissionToString(l_PermissionLevel.m_lNumber,
csPermLevName.GetBuffer(1024), 1024);
        ////////////////////////////////////////////////////////////////////
        ////////////////////////////////////////////////////////////////////
        if(!bRet)
        {
            m_pView->Print("ERROR: ", FSIZE_FUNCMARK);
            m_pView->Print("PWRTCheckPermission.\n", FSIZE_PARAMMARK, FALSE, TRUE);
            csOut.Format("Level = %ld (\\"%s\\") - Access denied.\n",
l_PermissionLevel.m_lNumber,
csPermLevName);
            m_pView->Print(csOut, FSIZE_SUBMARK);
            m_pView->Print("\n");
            return;
        }
        else
        {
            csOut.Format("Level = %ld (\\"%s\\")\n",
l_PermissionLevel.m_lNumber,
csPermLevName);
            m_pView->Print("PWRTCheckPermission\n", FSIZE_FUNCMARK, TRUE);
            m_pView->Print("Access approved:\n", FSIZE_PARAMMARK, FALSE, TRUE);
            m_pView->Print(csOut, FSIZE_SUBMARK);
            m_pView->Print("\n");
        }
    }
}

```

```
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

PWRTCheckPermission (Seite 666)

2.5.8.2 Checks admission of a certain level for a picture (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}Checks admission of a certain level for a picture. (USE)}  
//{{FUNCTION}PWRTCheckPermissionOnPicture (USE)}  
//{{FUNCTION} (END)}  
void CPWRTBunch::Pwrtcheckpermissiononpicture()  
{  
    BOOL bRet;  
    CMN_ERROR err;  
    CString csOut;  
    CGetText l_PermissionLevel("Enter the permission level:");  
    CGetText l_PictureName("Enter the picture name:", FALSE);  
    if(l_PermissionLevel.DoModal()==IDCANCEL)  
    {  
        return;  
    }  
    if(l_PictureName.DoModal()==IDCANCEL)  
    {  
        return;  
    }  
    //////////////////////////////////////  
    // bRet=PWRTCheckPermissionOnPicture(l_PermissionLevel.m_lNumber, l_PictureName.m_csText,  
    0, &err);  
    //////////////////////////////////////  
    // if(bRet)  
    {  
        csOut.Format("Picture: \"%s\"\nLevel = %ld\n",  
        l_PictureName.m_csText,  
        l_PermissionLevel.m_lNumber);  
        m_pView->Print("PWRTCheckPermissionOnPicture\n", FSIZE_FUNCMARK, TRUE);  
        m_pView->Print("Access approved:\n", FSIZE_PARAMMARK, FALSE, TRUE);  
        m_pView->Print(csOut, FSIZE_SUBMARK);  
        m_pView->Print("\n");  
    }  
    else  
    {  
        m_pView->PrintError(&err, "PWRTCheckPermissionOnPicture");  
        m_pView->PrintError("Access denied", "PWRTCheckPermissionOnPicture");  
    }  
}  
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

PWRTCheckPermissionOnPicture (Seite 667)

2.5.8.3 Gets a string associated with the permission number (RT Professional)**Übersicht**

```

//{{ODK_EXAMPLE}Gets a string associated with the permission number.
(USE)}
//{{FUNCTION}PWRTPermissionToString (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::Pwrtpermissiontostring()
{
    BOOL bRet;
    CString csOut;
    CString csPermLevName;
    CGetText l_PermissionLevel("Insert the permission level
number:");
    if(l_PermissionLevel.DoModal()==IDOK)
    {
        ///////////////////////////////////////////////////////////////////
        ///////////////////////////////////////////////////////////////////
        bRet=PWRTPermissionToString(l_PermissionLevel.m_lNumber,
csPermLevName.GetBuffer(1024), 1024);
        ///////////////////////////////////////////////////////////////////
        ///////////////////////////////////////////////////////////////////
        if(!bRet)
        {
            m_pView->Print("ERROR: ", FSIZE_FUNCMARK);
            m_pView->Print("PWRTPermissionToString.\n",
FSIZE_PARAMMARK, FALSE, TRUE);
            m_pView->Print("\n");
            return;
        }
        csOut.Format("PWRTPermissionToString( %ld, buffer )\n",
l_PermissionLevel.m_lNumber);
        m_pView->Print(csOut, FSIZE_FUNCMARK, TRUE);
        m_pView->Print("Permission level name:\n", FSIZE_PARAMMARK,
FALSE, TRUE);
        csOut.Format("buffer = \"%s\"\n", csPermLevName);
        m_pView->Print(csOut, FSIZE_SUBMARK);
        m_pView->Print("\n");
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

PWRTPermissionToString (Seite 670)

2.5.8.4 Permission level query through a dialog with specifying a possible error (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Permission level query through a dialog with specifying a possible error.
(USE)}
//{{FUNCTION}PWRTPermissionLevelDialogEx (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::PWRTxPermissionLevelDialogEx()
{
    long lLevel;
    CMN_ERROR err;
    CString csOut;
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    //
    lLevel=PWRTPermissionLevelDialogEx(*m_pView, &err);
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    //
    if(m_pView->PrintError(&err, "PWRTPermissionLevelDialogEx"))
    {
        m_pView->Print("lLevel = PWRTPermissionLevelDialogEx(...)\n", FSIZE_FUNCMARK, TRUE);
        m_pView->Print("Permission level number chosen:\n", FSIZE_PARAMMARK, FALSE, TRUE);
        csOut.Format("lLevel = %ld\n", lLevel);
        m_pView->Print(csOut, FSIZE_SUBMARK);
        m_pView->Print("\n");
    }
}
//{{ODK_EXAMPLE}(END)}
    
```

Siehe auch

- PWRTPermissionLevelDialog (Seite 668)
- PWRTPermissionLevelDialogEx (Seite 669)

2.5.8.5 Return the name of the current user (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Return the name of the current user. (USE)}
//{{FUNCTION}PWRTGetCurrentUser (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::Pwrtgetcurrentuser()
{
    BOOL bRet;
    CString csUsername;
    CString csOut;
    ////////////////////////////////////////////////////////////////////
    //
    bRet=PWRTGetCurrentUser(csUsername.GetBuffer(1024), 1023);
    ////////////////////////////////////////////////////////////////////
    //
    if(!bRet)
    {
        m_pView->Print("ERROR: ", FSIZE_FUNCMARK, TRUE);
        m_pView->Print("PWRTGetCurrentUser.\n", FSIZE_PARAMMARK, FALSE, TRUE);
        m_pView->Print("Failed to get current user's name.\n", FSIZE_SUBMARK);
        m_pView->Print("\n");
        return;
    }
    m_pView->Print("PWRTGetCurrentUser\n", FSIZE_FUNCMARK, TRUE);
    m_pView->Print("Current user's name:\n", FSIZE_PARAMMARK, FALSE, TRUE);
    csOut.Format("Name = \"%s\"\n", csUsername);
    m_pView->Print(csOut, FSIZE_SUBMARK);
    m_pView->Print("\n");
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

PWRTGetCurrentUser (Seite 671)

2.5.8.6 Queries the current login priority (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}Queries the current login priority. (USE)}
//{{FUNCTION}PWRTGetLoginPriority (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::Pwrtgetloginpriority()
{
    long lPriority;
    CString csOut;
    CMap<long, LONG, CString, CString> l_map(3);
    l_map[LOGIN_STANDARD]=CString("LOGIN_STANDARD");
    l_map[LOGIN_CARD]=CString("LOGIN_CARD");
    l_map[LOGIN_KEYSWITCH]=CString("LOGIN_KEYSWITCH");
    ////////////////////////////////////////////////////////////////////
//
    lPriority=PWRTGetLoginPriority();
    ////////////////////////////////////////////////////////////////////
//
    if(lPriority!=-1)
    {
        m_pView->Print("PWRTGetLoginPriority\n", FSIZE_FUNCMARK, TRUE);
        m_pView->Print("Priority returned:\n", FSIZE_PARAMMARK, FALSE, TRUE);
        csOut.Format("Priority level = %s\n", l_map[lPriority]);
        m_pView->Print(csOut, FSIZE_SUBMARK);
        m_pView->Print("\n");
    }
}
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

PWRTGetLoginPriority (Seite 672)

2.5.8.7 Checks for the user has been logged on by card (RT Professional)

Übersicht

```
//{{ODK_EXAMPLE}Checks for the user has been logged on by card. (USE)}
//{{FUNCTION}PWRTIsLoggedInByCard (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::Pwrtisloggedinbycard()
{
    BOOL bRet;
    CString csOut;
    ////////////////////////////////////////////////////////////////////
    ////////////////////////////////////////////////////////////////////
    bRet=PWRTIsLoggedInByCard();
}
```

```

////////////////////////////////////
////////////////////////////////////
        if(bRet)
        {
            m_pView->Print("PWRTIsLoggedInByCard\n", FSIZE_FUNCMARK,
TRUE);
            m_pView->Print("An user show-up logged in by card.\n",
FSIZE_PARAMMARK, FALSE, TRUE);
        }
        else
        {
            m_pView->PrintError(_T("Either no user logged-on-by-card or
no user logged on at all."),
            _T("PWRTIsLoggedInByCard()"));
        }
    }
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

PWRTIsLoggedInByCard (Seite 673)

2.5.8.8 PWRT login - dialog provided by WinCC itself (RT Professional)**Beispiel**

```

//{{ODK_EXAMPLE} PWRT login - dialog provided by WinCC intself. (USE)}
//{{FUNCTION}PWRTLogin (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::PWRTxLogin()
{
    BOOL bRet;
    //////////////////////////////////////
    //
    bRet=PWRTLogin(_T('1')); // it always appears on the first screen
    //////////////////////////////////////
    //
    if(!bRet)
    {
        m_pView->Print("ERROR: ", FSIZE_FUNCMARK, TRUE);
        m_pView->Print("PWRTLogin.\n", FSIZE_PARAMMARK, FALSE, TRUE);
        m_pView->Print("General function failure.\n", FSIZE_SUBMARK);
        m_pView->Print("\n");
        return;
    }
    m_pView->Print("PWRTLogin\n", FSIZE_FUNCMARK, TRUE);
    m_pView->Print("Logging on...\n", FSIZE_PARAMMARK, FALSE, TRUE);
    m_pView->Print("\n");
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

PWRTLogin (Seite 674)

2.5.8.9 PWRT logout (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}PWRT logout. (USE)}  
//{{FUNCTION}PWRTLogout (USE)}  
//{{FUNCTION}(END)}  
void CPWRTBunch::Pwrtlogout()  
{  
    BOOL bRet;  
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
    //  
    bRet=PWRTLogout(); // it appears on the first screen  
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
    //  
    if(!bRet)  
    {  
        m_pView->Print("ERROR: ", FSIZE_FUNCMARK, TRUE);  
        m_pView->Print("PWRTLogout.\n", FSIZE_PARAMMARK, FALSE, TRUE);  
        m_pView->Print("General function failure.\n", FSIZE_SUBMARK);  
        m_pView->Print("\n");  
        return;  
    }  
    m_pView->Print("PWRTLogout\n", FSIZE_FUNCMARK, TRUE);  
    m_pView->Print("Logging out...\n", FSIZE_PARAMMARK, FALSE, TRUE);  
    m_pView->Print("\n");  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

PWRTLogout (Seite 676)

2.5.8.10 Silent logout with priority level (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}Silent logout with priority level. (USE)}
//{{FUNCTION}PWRTLogoutEx (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::PWRTxLogoutEx()
{
    BOOL bRet;
    CString csOut;
    CGetText l_PriorityLevel("The priority level:");
    if(l_PriorityLevel.DoModal()==IDCANCEL)
    {
        return;
    }
    ///////////////////////////////////////////////////////////////////
    //
    bRet=PWRTLogoutEx( l_PriorityLevel.m_lNumber);
    ///////////////////////////////////////////////////////////////////
    //
    if(!bRet)
    {
        m_pView->PrintError( _T("Cannot logout."),
            _T("PWRTLogoutEx"));
    }
    else
    {
        m_pView->Print("PWRTLogoutEx\n", FSIZE_FUNCMARK, TRUE);
        m_pView->Print("Logging out...\n", FSIZE_PARAMMARK, FALSE, TRUE);
        csOut.Format("Priority level = %ld\n", l_PriorityLevel.m_lNumber);
        m_pView->Print(csOut, FSIZE_SUBMARK);
        m_pView->Print("\n");
    }
}
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

PWRTLogoutEx (Seite 677)

2.5.8.11 Login without using a dialog (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}Logs on without using a dialog. (USE)}  
//{{FUNCTION}PWRTSilentLogin (USE)}  
//{{FUNCTION}(END)}  
void CPWRTBunch::Pwrtsilentlogin()  
{  
    BOOL bRet;  
    CPassdlg l_Password("Enter name and password:");  
    if(l_Password.DoModal() == IDCANCEL)  
    {  
        return;  
    }  
    //////////////////////////////////////  
    //  
    bRet=PWRTSilentLogin(l_Password.m_csUsername,  
    l_Password.m_csPassword);  
    //////////////////////////////////////  
    //  
    if(!bRet)  
    {  
        m_pView->PrintError( _T("Access denied or no Runtime project open."),  
        _T("PWRTSilentLogin"));  
    }  
    else  
    {  
        m_pView->Print("PWRTSilentLogin\n", FSIZE_FUNCMARK, TRUE);  
        m_pView->Print("Access approved:\n", FSIZE_PARAMMARK, FALSE, TRUE);  
    }  
}  
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

PWRTSilentLogin (Seite 678)

2.5.8.12 Silent login with priority level (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Silent login with priority level. (USE)}
//{{FUNCTION}PWRTSilentLoginEx (USE)}
//{{FUNCTION}(END)}
void CPWRTBunch::Pwrtsilentloginex()
{
    BOOL bRet;
    CString csOut;
    CPassdlg l_Password("Enter name and password:");
    CGetText l_PriorityLevel("Enter priority level:");
    if(l_Password.DoModal()==IDCANCEL)
    {
        return;
    }
    if(l_PriorityLevel.DoModal()==IDCANCEL)
    {
        return;
    }
    ///////////////////////////////////////////////////////////////////
    //
    bRet=PWRTSilentLoginEx(l_Password.m_csUsername, l_Password.m_csPassword,
l_PriorityLevel.m_lNumber);
    ///////////////////////////////////////////////////////////////////
    //
    if(!bRet)
    {
        m_pView->PrintError( _T("Access denied or no Runtime project open."),
        _T("PWRTSilentLogin"));
    }
    else
    {
        m_pView->Print("PWRTSilentLogin\n", FSIZE_FUNCMARK, TRUE);
        m_pView->Print("Access approved:\n", FSIZE_PARAMMARK, FALSE, TRUE);
        csOut.Format("Priority level = %ld\n", l_PriorityLevel.m_lNumber);
        m_pView->Print(csOut, FSIZE_SUBMARK);
        m_pView->Print("\n");
    }
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

PWRTSilentLoginEx (Seite 679)

2.6 Funktionen des Textsystems (RT Professional)

2.6.1 Grundlagen (RT Professional)

2.6.1.1 Übersicht über die Funktionen (RT Professional)

Deklaration

TXT_ENUM_INFOTEXTS_PROC (Seite 704)	Infotexte auflisten (Callback)
TXT_ENUM_LANGUAGES_PROC (Seite 712)	projektierte Sprachen auflisten (Callback)
TXTCloseProject (Seite 695)	Textbibliothek schließen
TXTEnumInfoText (Seite 702)	Infotexte auflisten
TXTEnumLanguages (Seite 710)	projektierte Sprachen auflisten
TXTGetFont (Seite 713)	Zeichensatz einer Sprache ermitteln
TXTGetMaxTextID (Seite 697)	Text-ID (maximale) ermitteln
TXTOpenProject (Seite 698)	Textbibliothek öffnen
TXTRTConnect (Seite 700)	Verbindung zu Textserver festhalten
TXTRTDisconnect (Seite 701)	Verbindung zu Textserver freigeben
TXTRTGetInfoText (Seite 707)	Infotexte lesen
TXTRTGetInfoTextMC (Seite 708)	Infotexte lesen vom Textserver
TXTRTGetLanguageID (Seite 716)	Aktiviere Sprache ermitteln
TXTRTSetLanguage (Seite 718)	Sprachumschaltung der Fehlermeldungstexte
TXTShowLanguagesDialog (Seite 715)	Sprachen (installierte) anzeigen
TXTUpdateRuntime (Seite 705)	Infotexte im Runtime aktualisieren

2.6.1.2 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

TXT_SYS_ERROR	(Wert: 0x10000000)	Falls in dwError1 der Fehler-Struktur CMN_ERROR dieses Bit gesetzt ist, dann enthält dwError2 den Systemfehlercode.
TXT_OK	(Wert: 0x00000000)	Es ist kein Fehler aufgetreten.
TXT_NO_CONNECT	(Wert: 0x00000001)	Die Datenbank ist noch nicht geöffnet.
TXT_CONNECT	(Wert: 0x00000002)	Fehler beim Anbinden an die Datenbank.
TXT_UPDATE	(Wert: 0x00000003)	Fehler beim Schreiben in die Datenbank.

TXT_CREATE_KEY	(Wert: 0x00000004)	Die TextID konnte nicht erzeugt werden.
TXT_KEY_NOT_FOUND	(Wert: 0x00000005)	Die angegebene TextID konnte nicht gefunden werden.
TXT_LANGUA- GE_NOT_FOUND	(Wert: 0x00000006)	Die angegebene Sprache konnte nicht gefunden werden.
TXT_NO_NEW_RECORD	(Wert: 0x00000007)	Es konnte kein neuer Record angelegt werden (AddRecord erzeugte einen Fehler)
TXT_FONT_NOT_FOUND	(Wert: 0x00000008)	Font nicht gefunden
TXT_TABLE_NOT_FOUND	(Wert: 0x00000009)	Eine Datenbanktabelle wurde nicht gefunden. Das Feld dwError2 enthält nähere Informationen:
TXTVERSIONINFO	(Wert: 0x00000001)	Versionsinfo Tabelle wurde nicht gefunden.
TXTLANGUAGES	(Wert: 0x00000002)	Sprachen-Tabelle wurde nicht gefunden.
TXTTABLE	(Wert: 0x00000003)	Text-Tabelle wurde nicht gefunden.
TXT_WRONG_PROJECT	(Wert: 0x00000010)	falscher Projektname
TXT_OPEN	(Wert: 0x00000011)	Datenbank konnte nicht geladen werden.
TXT_PARAM	(Wert: 0x00000012)	Ungültiger Parameter
TXT_DISCONNECT	(Wert: 0x00000013)	Fehler beim Abmelden der Datenbank
TXT_CALLBACK	(Wert: 0x00000014)	Callback lieferte FALSE zurück
TXT_ALREADY_FOUND	(Wert: 0x00000015)	Text bereits vorhanden
TXT_CREATE	(Wert: 0x00000016)	Datenbanktabellen konnten nicht angelegt werden.
TXT_UPDATE_RUNTIME	(Wert: 0x00000017)	Runtime nicht gefunden oder SendMessage lieferte NULL zurück.
TXT_ALREADY_EXIST	(Wert: 0x00000018)	Sprache bereits vorhanden
TXT_NO_LANGUAGE	(Wert: 0x00000019)	ungültige Sprache angegeben
TXT_TEXT_LIBRARY_RUNNING	(Wert: 0x00000020)	TEXTBIB.EXE läuft bereits; öffnen nicht erlaubt
TXT_TABLES_ALREADY_EXIST	(Wert: 0x00000021)	Datenbanktabellen bereits vorhanden. Erzeugen nicht mehr erlaubt.

Text Library RT

TXT_RT_OK	(Wert: 0x00000000)	Kein Fehler
TXT_RT_OPEN_MMF	(Wert: 0x00000001)	Fehler beim Öffnen des Memory Mapped File
TXT_RT_CREATE_SERVICE_WINDOW	(Wert: 0x00000002)	Fehler beim Anlegen des Service Fensters

2.6 Funktionen des Textsystems (RT Professional)

TXT_RT_KEY_NOT_FOUND	(Wert: 0x00000003)	TextID nicht gefunden
TXT_RT_NO_READ_ACCESS	(Wert: 0x00000004)	Lesezugriff auf das Memory Mapped File verweigert
TXT_RT_NO_LANGUAGE_FOUND	(Wert: 0x00000005)	Keine Sprache gefunden
TXT_RT_LANGUAGE_NOT_FOUND	(Wert: 0x00000006)	Sprache nicht gefunden
TXT_RT_INTERNAL_ERROR	(Wert: 0x00000007)	Interner Fehler
TXT_RT_INVALIDPARAM	(Wert: 0x00000008)	Parameter Fehler
TXT_RT_NODEFAULTSERVER	(Wert: 0x00000011)	Es ist kein Standardserver projektiert
TXT_RT_NOLOCALSERVER	(Wert: 0x00000012)	Es ist kein lokaler Server verfügbar
TXT_RT_NOSERVER	(Wert: 0x00000013)	Es ist kein Standardserver projektiert und kein lokaler Server verfügbar
TXT_RT_NOMC	(Wert: 0x00000014)	Es ist kein Multiclient Projekt (wird hier nicht verwendet)
TXT_RT_NOMCDEFAULTSERVER	(Wert: 0x00000015)	Es ist kein Multiclient Projekt, aber "@default" wurde als Serverpräfix angegeben

2.6.1.3 Sprachcodes (RT Professional)

Sprachcodes des Textsystems

Von WinCC werden nur die SUBLANG_DEFAULT-Sprachen von Windows unterstützt. Dadurch können dem Parameter dwLocaleID den API-Funktionen des Textsystems die folgenden Werte zugeordnet werden:

LANG_ARABIC	0x0401
LANG_AFRIKAANS	0x0436
LANG_ALBANIAN	0x041C
LANG_BASQUE	0x042D
LANG_BULGARIAN	0x0402
LANG_BELARUSIAN	0x0423
LANG_CATALAN	0x0403
LANG_CHINESE	0x0804
LANG_CROATIAN	0x041A
LANG_CZECH	0x0405
LANG_DANISH	0x0406
LANG_DUTCH	0x0413
LANG_ENGLISH	0x0409
LANG_ESTONIAN	0x0425
LANG_FAEROESE	0x0438
LANG_FARSI	0x0429

LANG_FINNISH	0x040B
LANG_FRENCH	0x040C
LANG_GERMAN	0x0407
LANG_GREEK	0x0408
LANG_HEBREW	0x040D
LANG_HUNGARIAN	0x040E
LANG_ICELANDIC	0x040F
LANG_INDONESIAN	0x0421
LANG_ITALIAN	0x0410
LANG_JAPANESE	0x0411
LANG_KOREAN	0x0412
LANG_LATVIAN	0x0426
LANG_LITHUANIAN	0x0427
LANG_NORWEGIAN	0x0414
LANG_POLISH	0x0415
LANG_PORTUGUESE	0x0416
LANG_ROMANIAN	0x0418
LANG_RUSSIAN	0x0419
LANG_SLOVAK	0x041B
LANG_SLOVENIAN	0x0424
LANG_SPANISH	0x040A
LANG_SWEDISH	0x041D
LANG_THAI	0x041E
LANG_TURKISH	0x041F
LANG_UKRAINIAN	0x0422

Dabei kennzeichnen die oberen 4 Byte (0x04..) die Sprache als SUBLANG_DEFAULT-Sprache

2.6.2 Allgemeine Funktionen (RT Professional)

2.6.2.1 TXTCloseProject (RT Professional)

Beschreibung

Schließt bei erfolgreicher Ausführung die Projektdatenbank.

Deklaration

```
BOOL TXTCloseProject (  
    LPCTSTR      lpszProjectFile,  
    LPCMN_ERROR  lpdmError );
```

Parameter

lpszProjectFile

Zeiger auf den Namen des Projekts einschließlich Pfad und Extension.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Textbibliothek geschlossen.

FALSE

Fehler

Bemerkung

Die Funktion TXTOpenProject zählen mit jedem Aufruf einen internen Referenz-Zähler hoch. Jeder Aufruf von TXTCloseProject zählt diesen Zähler wieder herunter. Erst wenn der Referenz-Zähler wieder auf 0 steht, wird auch die Textbibliothek geschlossen.

Daher muss unbedingt die gleiche Anzahl von TXTCloseProject aufgerufen werden, da es sonst beim Beenden der Applikation zu einem Ausnahme-Fehler kommen kann.

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX, ...) benutzt werden. Aufgrund von Microsoft-spezifischen Mechanismen kann das ggfs. zum Hängenbleiben des Aufrufs und damit des Programms führen.

Fehlermeldungen

TXT_DISCONNECT	Fehler beim Abmelden der Datenbank
----------------	------------------------------------

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Verwandte Funktionen

TXTOpenProject (Seite 698)	Öffnet die Textbibliothek
----------------------------	---------------------------

Beispiele

Get Helptext (Seite 719) "TX01.cpp"

Enumerate Helptexts (Seite 721) "TX01.cpp"

Siehe auch

TXTOpenProject (Seite 698)

Get Helptext (Seite 719)

Enumerate Helptexts (Seite 721)

Übersicht über die Funktionen (Seite 692)

2.6.2.2 TXTGetMaxTextID (RT Professional)

Beschreibung

Ermittelt die höchste Text-ID, die in der DLL geführt wird.

Deklaration

```
BOOL TXTGetMaxTextID (  
    LPLONG          lpMaxTextID,  
    LPCMN_ERROR     lpdmError );
```

Parameter

lpMaxTextID

Zeiger auf den Puffer, in dem die Text-ID hinterlegt werden soll.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

maximale Text-ID ermittelt

FALSE

Fehler

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Siehe auch

Übersicht über die Funktionen (Seite 692)

2.6.2.3 TXTOpenProject (RT Professional)

Beschreibung

Öffnet bei erfolgreicher Ausführung die Datenbank.

Deklaration

```
BOOL TXTOpenProject (  
    LPCTSTR      lpszProjectFile,  
    LPCTSTR      lpszDSNName,  
    BOOL         fExclusive,  
    LPCMN_ERROR  lpdmError );
```

Parameter

lpszProjectFile

Zeiger auf den Namen des Projekts einschließlich Pfad und Extension.

lpszDSNName

Zeiger auf den Namen der Data Source.

fExclusive

Wenn TRUE, kann TEXTBIB.EXE nicht gestartet werden, weil z.B. ein anderes Programm auf die Text Library zugreift. Ist TEXTBIB.EXE schon gestartet, so wird der Aufruf abgewiesen

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Datenbank geöffnet

FALSE

Fehler

Bemerkung

Wenn noch keine Textbibliothek-Tabellen vorhanden sind, werden diese neu erzeugt.

Bei jedem Aufruf der Funktion wird ein Referenz-Zähler erhöht. Zum Beenden muss die gleiche Anzahl TXTCloseProject aufgerufen werden. Ansonsten kann es beim Beenden der Applikation zu einem Ausnahme-Fehler kommen.

Fehlermeldungen

TXT_CONNECT	Fehler beim Anbinden an die Datenbank.
TXT_TABLE_NOT_FOUND	Eine Datenbanktabelle wurde nicht gefunden. Das Feld dwError2 enthält nähere Informationen:
TXT_TEXT_LIBRARY_RUNNING	TEXTBIB.EXE ist bereits gestartet
TXT_WRONG_PROJECT	falscher Projektname

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Verwandte Funktionen

TXTCloseProject (Seite 695)	Schließt die Textbibliothek
-----------------------------	-----------------------------

Beispiele

Get Helptext (Seite 719) "TX01.cpp"

Enumerate Helptexts (Seite 721) "TX01.cpp"

Siehe auch

- TXTCloseProject (Seite 695)
- Get Helptext (Seite 719)
- Enumerate Helptexts (Seite 721)
- Übersicht über die Funktionen (Seite 692)

2.6.2.4 TXTRTConnect (RT Professional)

Beschreibung

Baut eine Verbindung zum Textserver auf. Diese Funktion beschleunigt die Ausführung der TXTRT-Funktionen. Es wird dann nicht in jedem Aufruf die Verbindung zum Textserver aufgebaut und am Ende wieder abgebaut.

Deklaration

```
BOOL TXTRTConnect ( );
```

Parameter

Keine

Rückgabewert

- TRUE**
Verbindung zum Textserver aufgebaut
- FALSE**
Fehler

Bemerkung

Vor Beenden der Applikation muss unbedingt die Verbindung wieder mit TXTRTDisconnect abgebaut werden, um nachfolgende Fehlfunktionen im System zu vermeiden.

Benötigte Dateien

- text_rt.h
- text_rt.lib
- text_rt.dll

Verwandte Funktionen

TXTRTDisconnect (Seite 701)	Verbindung zu Textserver freigeben
-----------------------------	------------------------------------

Siehe auch

TXTRTDisconnect (Seite 701)
Übersicht über die Funktionen (Seite 692)

2.6.2.5 TXTRTDisconnect (RT Professional)

Beschreibung

Baut eine Verbindung mit TXTRTConnect eingerichtete Verbindung zum Textserver ab.

Deklaration

```
BOOL TXTRTDisconnect ( );
```

Parameter

Keine

Rückgabewert

TRUE

Verbindung zum Textserver aufgebaut

Bemerkung

Diese Funktion beendet die Beschleunigung der TXTRT-Funktionen und liefert immer TRUE zurück.

Nachfolgend aufgerufene TXTRT-Funktionen bauen dann wieder bei jedem Aufruf die Verbindung zum Textserver neu auf und wieder ab.

Wurde TXTRTConnect verwendet und wird vor Beenden der Applikation kein TXTRTDisconnect ausgeführt, kann es zu nachfolgend zu Fehlfunktionen im System kommen.

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX, ...) benutzt werden. Aufgrund von Microsoft-spezifischen Mechanismen kann das ggfs. zum Hängenbleiben des Aufrufs und damit des Programms führen.

Benötigte Dateien

text_rt.h
text_rt.lib
text_rt.dll

Verwandte Funktionen

TXTRTConnect (Seite 700)	Verbindung zu Textserver festhalten
--------------------------	-------------------------------------

Siehe auch

TXTRTConnect (Seite 700)
Übersicht über die Funktionen (Seite 692)

2.6.3 Funktionen zur Bearbeitung von Hilfetexten (RT Professional)

2.6.3.1 TXTEnumInfoText (RT Professional)

Beschreibung

Listet alle Hilfetexte, die dem Filterkriterium in lpszFilter entsprechen, auf.

Deklaration

```
BOOL TXTEnumInfoText (
    LPCTSTR                lpszProjectFile,
    DWORD                  dwLocale,
    LPDWORD                lpdwItems,
    LPCTSTR                lpszFilter,
    TXT_ENUM_INFOTEXTS_PROC lpfnEnum,
    LPVOID                 lpvUser,
    LPCMN_ERROR            lpdmError );
```

Parameter

lpszProjectFile

Zeiger auf den Namen des Projekts einschließlich Pfad und Extension.

dwLocale

Sprachcode der Sprache, deren Infotexte enumeriert werden sollen.

lpdwItems

Zeiger auf eine Doppelwortvariable der Applikation, die nach dem Aufruf die Anzahl der enumerierten Infotexte beinhaltet.

lpszFilter

Zeiger auf die Bedingung eines SQL-Statements für den LIKE-Operator.

lpfnEnum

Ihre Callback-Funktion, die die Hilfetexte entgegennimmt.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Infotexte aufgelistet

FALSE

Fehler

Bemerkung

Wenn lpszFilter == NULL ist oder ein leerer String übergeben wird, werden alle Hilfetexte der Sprache enumeriert. Dabei werden dann auch leere, nicht besetzte, Infotexte mit enumeriert, und über lpdwItems mitgezählt.

Fehlermeldungen

TXT_CALLBACK	Callback lieferte FALSE zurück
TXT_LANGUAGE_NOT_FOUND	Die angegebene Sprache konnte nicht gefunden werden.
TXT_NO_CONNECT	Die Datenbank ist noch nicht geöffnet.
TXT_NO_LANGUAGE	ungültige Sprache angegeben
TXT_WRONG_PROJECT	falscher Projektname

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Verwandte Funktionen

TXT_ENUM_INFOTEXTS_PROC (Seite 704)	Hilfetexte auflisten (Callback)
-------------------------------------	---------------------------------

Beispiele

Enumerate Helptexts (Seite 721) "TX01.cpp"

Siehe auch

TXT_ENUM_INFOTEXTS_PROC (Seite 704)

Enumerate Helptexts (Seite 721)

Übersicht über die Funktionen (Seite 692)

2.6.3.2 TXT_ENUM_INFOTEXTS_PROC (RT Professional)

Beschreibung

Um die aufgelisteten Tooltips auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TXT_ENUM_INFOTEXTS_PROC bereitstellen.

Deklaration

```
BOOL ( * TXT_ENUM_INFOTEXTS_PROC) (  
    DWORD      dwTextID,  
    LPCTSTR    lpszInfoText,  
    LPVOID     lpvUser );
```

Parameter

dwTextID

ID des Texts, auf den lpszInfotext verweist.

lpszInfoText

Zeiger auf den Tooltip, der von der aufrufenden Funktion übergeben wurde.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert**TRUE**

Die Enumeration wird fortgesetzt

FALSE

Die Enumeration wird abgebrochen

Bemerkung**Hinweis**

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen, in denen auf eine Message-Loop zugegriffen wird, z. B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Verwandte Funktionen

TXTEnumInfoText (Seite 702)	Hilfetexte auflisten
---	--------------------------------------

Beispiele

Enumerate Helptexts (Seite 721) "TX01.cpp"

Siehe auch

[TXTEnumInfoText \(Seite 702\)](#)

[Enumerate Helptexts \(Seite 721\)](#)

[Übersicht über die Funktionen \(Seite 692\)](#)

2.6.3.3 TXTUpdateRuntime (RT Professional)**Beschreibung**

Im Runtime werden die Texte aus der Textbibliothek neu geladen.

Deklaration

```
BOOL TXTUpdateRuntime (  
    LPCTSTR      lpszProjectFile,  
    HWND         hwndParent,  
    LPCMN_ERROR  lpdmError );
```

Parameter

lpszProjectFile

Zeiger auf den Namen des Projekts einschließlich Pfad und Extension.

hwndParent

Handle auf das Vaterfenster

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Infotexte aktualisiert

Bemerkung

Bei Textreferenzvariablen erfolgt keine Aktualisierung. Die Funktion liefert immer TRUE zurück.

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Siehe auch

Übersicht über die Funktionen (Seite 692)

2.6.3.4 TXTRTGetInfoText (RT Professional)

Beschreibung

Füllt bei erfolgreicher Ausführung den in lpszBuffer übergebenen Puffer mit dem durch dwTextID beschriebenen Text. Ist der übergebene Puffer kleiner als die Textlänge (Pufferlänge in Zeichen in pdwSize) wird der Text entsprechend verkürzt. Der Rückgabewert ist trotzdem TRUE. Wird in lpszBuffer NULL übergeben, ermittelt die Funktion die erforderliche Puffergröße, und hinterlegt diese in pdwSize.

Deklaration

```
BOOL TXTRTGetInfoText (  
    DWORD          dwTextID,  
    LPTSTR         lpszBuffer,  
    LPDWORD        pdwSize,  
    LPCMN_ERROR    lpdmError );
```

Parameter

dwTextID

ID des zu lesenden Texts

lpszBuffer

Zeiger auf einen Puffer, in dem der Infotext abgelegt werden soll.

pdwSize

Zeiger auf das DWORD, das die Puffergröße enthält

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Hilfetext ermittelt

FALSE

Fehler

Bemerkung

Fehlermeldungen

TXT_RT_OPEN_MMF	Fehler beim Öffnen des Memory Mapped File
TXT_RT_CREATE_SERVICE_WINDOW	Fehler beim Anlegen des Service Fensters
TXT_KEY_NOT_FOUND	Die angegebene TextID konnte nicht gefunden werden.
TXT_RT_NO_READ_ACCESS	Lesezugriff auf das Memory Mapped File verweigert

Benötigte Dateien

- text_rt.h
- text_rt.lib
- text_rt.dll

Verwandte Funktionen

AUTOHOTSPOT	Tooltips vom Textserver lesen
-------------	-------------------------------

Beispiele

Get Helptext (Seite 719) "TX01.cpp"

Siehe auch

- Get Helptext (Seite 719)
- Übersicht über die Funktionen (Seite 692)

2.6.3.5 TXTRTGetInfoTextMC (RT Professional)

Beschreibung

Holt einen Text, z. B. Tooltip vom angegebenen Textserver oder aus der C-Textliste. Füllt bei erfolgreicher Ausführung den in `IpszBuffer` übergebenen Puffer mit durch `dwTextID` beschriebenen Text. Ist der übergebene Puffer kleiner als die Textlänge (Pufferlänge in Zeichen in `pdwSize`), wird der Text entsprechend verkürzt. Der Rückgabewert ist trotzdem `TRUE`. Wird in `IpszBuffer` `NULL` übergeben, ermittelt die Funktion die erforderliche Puffergröße, und hinterlegt diese in `pdwSize`.

Deklaration

```

BOOL TXTRTGetInfoTextMC (
    DWORD          dwTextID,
    LPTSTR         lpszBuffer,
    LPDWORD        pdwSize,
    LPTSTR         lpszServer,
    LPCMN_ERROR    lpdmError );

```

Parameter

dwTextID

ID des zu lesenden Texts

lpszBuffer

Zeiger auf einen Puffer, in dem der Infotext abgelegt werden soll.

pdwSize

Zeiger auf das DWORD, das die Puffergröße enthält

lpszServer

Zeiger auf den symbolischen Namen des Textservers (ohne Servertrennzeichen ::)

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Infotext ermittelt

FALSE

Fehler

Bemerkung

Diese Funktion arbeitet nur bei Multiclientprojekten

Fehlermeldungen

TXT_RT_OPEN_MMF	Fehler beim Öffnen des Memory Mapped File
TXT_RT_CREATE_SERVICE_WINDOW	Fehler beim Anlegen des Service Fensters
TXT_KEY_NOT_FOUND	Die angegebene TextID konnte nicht gefunden werden.
TXT_RT_NO_READ_ACCESS	Lesezugriff auf das Memory Mapped File verweigert

TXT_RT_NODEFAULTSERVER	Es ist kein Standardserver projiziert.
TXT_RT_NOLOCALSERVER	Es ist kein lokaler Server verfügbar.
TXT_RT_NOSERVER	Es ist kein Standardserver projiziert und kein lokaler Server verfügbar.
TXT_RT_NOMCDEFAULTSERVER	Es ist kein Multiclient Projekt, aber "@default" wurde als Serverpräfix angegeben.

Benötigte Dateien

- text_rt.h
- text_rt.lib
- text_rt.dll

Verwandte Funktionen

AUTOHOTSPOT	Tooltips lesen
-------------	----------------

Siehe auch

Übersicht über die Funktionen (Seite 692)

2.6.4 Funktionen zur Bearbeitung von Sprachen (RT Professional)

2.6.4.1 TXTEnumLanguages (RT Professional)

Beschreibung

Ruft die übergebene Callback-Funktion für alle projizierten Sprachen auf.

Deklaration

```
BOOL TXTEnumLanguages (
    LPCTSTR                lpszProjectFile,
    LPDWORD                lpdwItems,
    TXT_ENUM_LANGUAGES_PROC lpfEnum,
    LPVOID                 lpvUser,
    LPCMN_ERROR            lpdmError );
```

Parameter

IpszProjectFile

Zeiger auf den Namen des Projekts einschließlich Pfad und Extension.

IpdwItems

Zeiger auf eine Doppelwortvariable der Applikation, die nach dem Aufruf die Anzahl der enumerierten Sprachen beinhaltet.

IpfEnum

Ihre Callback-Funktion, die die projektierten Sprachen entgegennimmt.

IpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Projektierte Sprachen aufgelistet

FALSE

Fehler

Fehlermeldungen

TXT_CALLBACK	Callback lieferte FALSE zurück
TXT_NO_CONNECT	Die Datenbank ist noch nicht geöffnet.
TXT_WRONG_PROJECT	falscher Projektname

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Verwandte Funktionen

TXT_ENUM_LANGUAGES_PROC (Seite 712)	Projektierte Sprachen auflisten (Callback)
-------------------------------------	--

Beispiele

Enumerate Helptexts (Seite 721) "TX01.cpp"

Siehe auch

TXT_ENUM_LANGUAGES_PROC (Seite 712)

Enumerate Helptexts (Seite 721)

Übersicht über die Funktionen (Seite 692)

2.6.4.2 TXT_ENUM_LANGUAGES_PROC (RT Professional)

Beschreibung

Um die aufgelisteten Sprachen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TXT_ENUM_LANGUAGES_PROC bereitstellen.

Deklaration

```
BOOL ( * TXT_ENUM_LANGUAGES_PROC) (  
    DWORD      dwLocaleID,  
    LPCTSTR    lpszName,  
    LPVOID     lpvUser );
```

Parameter

dwLocaleID

Sprachcode

lpszName

Zeiger auf den Namen der Sprache.

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen, in denen auf eine Message-Loop zugegriffen wird, z. B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Benötigte Dateien

text_cs.h

Verwandte Funktionen

TXTEnumLanguages (Seite 710)	Pprojektierte Sprachen auflisten
--	--

Beispiele

Enumerate Helptexts (Seite 721) "TX01.cpp"

Siehe auch

[TXTEnumLanguages \(Seite 710\)](#)

[Enumerate Helptexts \(Seite 721\)](#)

[Übersicht über die Funktionen \(Seite 692\)](#)

2.6.4.3 TXTGetFont (RT Professional)

Beschreibung

Füllt bei erfolgreicher Ausführung die in lplf übergebene Struktur LOGFONT mit dem eingestellten Font der gewünschten Sprache.

Deklaration

```

BOOL TXTGetFont (
    LPCTSTR      lpszProjectFile,
    DWORD        dwLocale,
    LPLOGFONT    lplf,
    LPCMN_ERROR  lpdmError );

```

Parameter

lpszProjectFile

Zeiger auf den Namen des Projekts einschließlich Pfad und Extension.

dwLocale

Sprachcode der Sprache, deren Font geholt werden soll

lpf

Zeiger auf eine windows-spezifische Struktur vom Typ LOGFONT mit den Daten eines Zeichensatzes.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Zeichensatz ermittelt

FALSE

Fehler

Fehlermeldungen

TXT_NO_CONNECT	Die Datenbank ist noch nicht geöffnet.
TXT_NO_LANGUAGE	Ungültige Sprache angegeben
TXT_LANGUAGE_NOT_FOUND	Die angegeben Sprache konnte nicht gefunden werden.
TXT_WRONG_PROJECT	Falscher Projektname

Benötigte Dateien

text_cs.h

text_cs.lib

text_cs.dll

Siehe auch

Übersicht über die Funktionen (Seite 692)

2.6.4.4 TXTShowLanguagesDialog (RT Professional)

Beschreibung

Öffnet einen Dialog mit den vorhandenen Sprachen. Wird der Dialog mit "OK" verlassen, so wird die ausgewählte Sprache in `lpdwLocale` hinterlegt. Außerdem wird der dazugehörige Font in der Struktur `LOGFONT` übergeben.



Deklaration

```

BOOL TXTShowLanguagesDialog (
    LPCTSTR          lpszProjectFile,
    HWND             hwndParent,
    LPDWORD          lpdwLocale,
    LPLOGFONT        lplf,
    LPCMN_ERROR      lpdmError );

```

Parameter

lpszProjectFile

Zeiger auf den Namen des Projekts einschließlich Pfad und Extension.

hwndParent

Handle auf das Vaterfenster in dem der Dialog stattfindet. Der Parameter ist mit NULL vorzubeseetzen.

dwLocale

Zeiger auf den Sprachcode der zu installierenden Sprache.

lpf

Zeiger auf eine windows-spezifische Struktur vom Typ LOGFONT mit den Daten eines Zeichensatzes.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Sprache gewählt

FALSE

Fehler oder Dialog mit "Abbrechen" verlassen

Bemerkung

Um im Dialogfenster die installierten Sprachen anbieten zu können, werden von der Funktion die in der Datenbank vorhandenen Sprachen enumeriert. Sollte dabei ein Fehler auftreten, so wird in der Fehlerstruktur die Fehlermeldung TXT_CALLBACK hinterlegt.

Fehlermeldungen

TXT_CALLBACK	Callback lieferte FALSE zurück
TXT_NO_CONNECT	Die Datenbank ist noch nicht geöffnet.
TXT_WRONG_PROJECT	falscher Projektname

Benötigte Dateien

- text_cs.h
- text_cs.lib
- text_cs.dll

Siehe auch

Übersicht über die Funktionen (Seite 692)

2.6.4.5 TXTRTGetLanguageID (RT Professional)

Beschreibung

Füllt bei erfolgreicher Ausführung den in lpchLanguageID übergebenen Puffer mit der im Memory Mapped File geladenen Primary Language ID.

Deklaration

```

BOOL TXTRTGetLanguageID (
    LPBYTE          lpchLanguageID,
    LPCMN_ERROR    lpdmError );

```

Parameter

lpchLanguageID

Zeiger auf einen Puffer, in dem die Primary Language ID abgelegt werden soll.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Aktivierte Sprache ermittelt

FALSE

Fehler

Fehlermeldungen

TXT_RT_OPEN_MMF	Fehler beim Öffnen des Memory Mapped File
TXT_RT_CREATE_SERVICE_WINDOW	Fehler beim Anlegen des Service Fensters
TXT_RT_NO_LANGUAGE_FOUND	Keine Sprache gefunden

Benötigte Dateien

text_rt.h

text_rt.lib

text_rt.dll

Siehe auch

Übersicht über die Funktionen (Seite 692)

2.6.4.6 TXTRTSetLanguage (RT Professional)

Beschreibung

Setzt die Sprache für die Fehlermeldungen. Ist diese Sprache nicht vorhanden wird, so wird die Default-Sprache (Deutsch) eingestellt und FALSE zurückgegeben.

Diese Funktion wird nicht mehr unterstützt und liefert immer den Returnwert TRUE zurück.

Deklaration

```
BOOL TXTRTSetLanguage (  
    DWORD          dwLocaleID,  
    LPCMN_ERROR    lpdmError );
```

Parameter

dwLocaleID

Sprachcode der einzustellenden Sprache.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Sprache der Fehlermeldungen umgeschaltet

Benötigte Dateien

text_rt.h

text_rt.lib

text_rt.dll

Siehe auch

Übersicht über die Funktionen (Seite 692)

2.6.5 Beispiele (RT Professional)

2.6.5.1 Get Helptext (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Insert new Infotext (TXT)}
//{{FUNCTION}TXTOpenProject (TXT)}
//{{FUNCTION}TXTNewInfoText (TXT)}
//{{FUNCTION}TXTSetInfoText (TXT)}
//{{FUNCTION}TXTCloseProject (TXT)}
//{{FUNCTION}(END)}
// =====
// Function: MyTxtSetInfotext(void) ODK DM CS
// =====
// Abst.    :    Insert New Text
// =====
void MyTxtNewInfotext(void)
{
    TCHAR szText[255];
    CMN_ERROR Error;
    BOOL ret = FALSE;
    DWORD dwLocale = 0x0407; // german
    TCHAR szFilter[]="S%"; // all texts with 'S'
    DWORD dwSize =254;
    DWORD dwTextID = 0;
    ret = MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile and
g_szDSNName
    if(TRUE == ret)
    {
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = TXTOpenProject(g_szProjectFile, g_szDSNName, FALSE, &Error);
        if(FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TXTOpenProject:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
            ODKTrace(szText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TXTOpenProject"));
            ODKTrace(szText);
            dwTextID = 380;
            dwLocale = 0x0407; // german
            _tcsncpy_s(szText, _countof(szText), _T("NewInfoText_ODK"), _TRUNCATE);
            memset(&Error, 0, sizeof(CMN_ERROR));
            ret = TXTGetInfoText(g_szProjectFile, dwTextID, dwLocale, szText, &dwSize,
&Error);
            if(FALSE == ret)
            {
                _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TXTNewInfoText: E1= 0x%08lx ; E2= 0x%08lx ; %s"),

```

```

        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TXTNewInfoText"));
    }
    ODKTrace(szText);
    dwLocale=0x0409; // english
    //dwTextID = 380; // use ID from TXTNewInfoText
    _tcsncpy_s(szText, _countof(szText), _T("InfoText_ODK"), _TRUNCATE);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TXTSetInfoText(g_szProjectFile, NULL, dwTextID, dwLocale, szText, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TXTSetInfoText: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TXTSetInfoText"));
    }
    ODKTrace(szText);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TXTCloseProject(g_szProjectFile, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TXTCloseProject: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TXTCloseProject"));

        ODKTrace(szText);
    }
    }
}
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

[TXTCloseProject \(Seite 695\)](#)
[TXTOpenProject \(Seite 698\)](#)
[TXTRTGetInfoText \(Seite 707\)](#)

2.6.5.2 Enumerate Helptexts (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enumerate texts (TXT)}
//{{FUNCTION}TXTOpenProject (TXT)}
//{{FUNCTION}TXTCloseProject (TXT)}
//{{FUNCTION}TXTEnumLanguages (TXT)}
//{{FUNCTION}TXT_ENUM_LANGUAGES_PROC (TXT)}
//{{FUNCTION}TXTEnumInfoText (TXT)}
//{{FUNCTION}TXT_ENUM_INFOTEXTS_PROC (TXT)}
//{{FUNCTION}(END)}
// =====
// Function: MyTxtEnums(void) ODK DM CS
// =====
// Abst. : Enumerate Texts
// =====

BOOL MyTXTEnumLanguagesCallback (DWORD dwLocID, LPCTSTR lpszName, LPVOID lpvUser)
{
    lpvUser;
    TCHAR szText[255];
    BOOL ret = TRUE;
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    InfoLang: ID=%x ;
    ODKTrace(szText);
    return(ret);
}

BOOL MyTXTEnumInfotextsCallback(DWORD dwTextID, LPCTSTR lpszInfoText, LPVOID lpvUser)
{
    lpvUser;
    TCHAR szText[255];
    BOOL ret = TRUE;
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    InfoText: ID=%.4d ; Tx=%s "),
dwTextID, lpszInfoText);
    ODKTrace(szText);
    return(ret);
}

void MyTxtEnums(void)
{
    TCHAR szText[255];
    CMN_ERROR Error;
    BOOL ret = FALSE;
    DWORD dwLocale = 0x0407; // german
    DWORD dwItems = 0;
    DWORD dwUser = 0;
    TCHAR szFilter[]="S%"; // all texts with 'S'
    //DWORD dwSize =254;
    ret = MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile and
g_szDSNName
    if(TRUE == ret)
    {
        memset(&Error, 0, sizeof(CMN_ERROR));

```

2.6 Funktionen des Textsystems (RT Professional)

```

ret = TXTOpenProject(g_szProjectFile, g_szDSNName, FALSE, &Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TXTOpenProject:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TXTOpenProject"));
    ODKTrace(szText);
    dwItems = 0;
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TXTEnumLanguages(g_szProjectFile, &dwItems, MyTXTEnumLanguagesCallback,
&dwUser, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TXTOpenProject: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    dwItems = 0;
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TXTEnumInfoText(g_szProjectFile, dwLocale, &dwItems, szFilter,
MyTXTEnumInfotextsCallback, &dwUser, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TXTEnumInfoText: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TXTEnumInfoText: NItems=
%d ; LANG=%x ; Filter=%s "),
        dwItems, dwLocale, szFilter);
    ODKTrace(szText);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TXTCloseProject(g_szProjectFile, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TXTCloseProject: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
}
}
}
}
//{{ODK_EXAMPLE} (END) }

```

Siehe auch

TXTCloseProject (Seite 695)
 TXTOpenProject (Seite 698)
 TXTEnumInfoText (Seite 702)
 TXT_ENUM_INFOTEXTS_PROC (Seite 704)
 TXT_ENUM_LANGUAGES_PROC (Seite 712)
 TXTEnumLanguages (Seite 710)

2.7 Funktionen des Protokollsystems (RT Professional)**2.7.1 Grundlagen (RT Professional)****2.7.1.1 Übersicht über die Funktionen (RT Professional)****Übersicht**

RPJAttach (Seite 736)	Verbindung aufbauen
RPJCallJobMethod (Seite 759)	Druckauftragsmethode aufrufen
RPJCreateJob (Seite 746)	Druckauftrag anlegen
RPJCreatePropertyHandle (Seite 749)	Handle einer Druckauftragseigenschaft erzeugen
RPJDeleteJob (Seite 747)	Druckauftrag löschen
RPJDeletePropertyHandle (Seite 750)	Handle einer Druckauftragseigenschaft löschen
RPJDetach (Seite 737)	Verbindung abbauen
RPJGetJobMethodAt (Seite 760)	Druckauftragsmethode Namen ermitteln
RPJGetJobNameAt (Seite 752)	Druckauftrag Namen ermitteln
RPJGetJobPropertyAt (Seite 762)	Druckauftragseigenschaften Namen ermitteln
RPJGetJobProps (Seite 764)	Druckauftragseigenschaften ermitteln
RPJGetNumJobMethods (Seite 761)	Druckauftragsmethode Anzahl ermitteln
RPJGetNumJobProperties (Seite 765)	Druckauftragseigenschaften Anzahl ermitteln
RPJGetNumJobs (Seite 753)	Druckauftrag Anzahl ermitteln
RPJGetNumProjectProperties (Seite 739)	Anzahl der Eigenschaften im Projekt ermitteln
RPJGetProjectPropertyAt (Seite 740)	Eigenschaftsname und Typ im Projekt über Index ermitteln
RPJGetProjectProperty (Seite 741)	Wert der Eigenschaft im Projekt ermitteln
RPJGetProperty (Seite 766)	Druckauftragseigenschaft ermitteln
RPJJobLock (Seite 754)	Druckauftrag für andere Zugriffe sperren
RPJJobUnlockAll (Seite 757)	Alle Sperren des Druckauftrags wieder aufheben
RPJJobUnlock (Seite 756)	Eine Sperre auf den Druckauftrag wieder aufheben
RPJMemFree (Seite 738)	Speicher freigeben

RPJProjectLock (Seite 742)	Zugriff auf die Druckauftragsliste im Projekt sperren
RPJProjectUnlockAll (Seite 745)	Alle Sperren auf die Druckauftragsliste im Projekt wieder aufheben
RPJProjectUnlock (Seite 744)	Eine Sperre auf die Druckauftragsliste im Projekt wieder aufheben
RPJPropertyClear (Seite 769)	Handle einer Druckauftragseigenschaft löschen
RPJSetProperty (Seite 771)	Druckauftragseigenschaften festlegen

2.7.1.2 Konstanten (RT Professional)

Eigenschaften für Layouts (CS)

Root

- ObjectName
- Context

Geometrie

- Left
- Top
- Width
- Height
- RoundCornerWidth
- RoundCornerHeight
- StartAngle
- EndAngle
- Radius
- RadiusWidth
- RadiusHeight
- Columns
- Index
- PointCount
- ActualPointLeft
- ActualPointTop
- PaperSize
- Orientation
- DynMarginLeft
- DynMarginRight
- DynMarginTop

- DynMarginBottom
- PrintMarginLeft
- PrintMarginRight
- PrintMarginTop
- PrintMarginBottom

Styles

- BorderStyle
- BorderWidth
- FillStyle

Farben

- BorderBackColor
- BorderColor
- BackColor
- FillColor
- ForeColor

Zeichen

- FontName
- FontSize
- FontBold
- FontItalic
- FontUnderline
- Text
- AlignmentLeft
- AlignmentTop
- WordWrap

Sonstige

- FirstPage
- LastPage
- MetaFileName
- LayoutFileName
- Format
- List
- Tag
- DataType
- OutFormat

- Calculation
- PageBreak

Eigenschaften für Druckaufträge

AbsoluteSelectionFrom	
AbsoluteSelectionTo	
CycleSpan	
DestinationFile	
EnableCycle	
EnableStart	
EndPage	Mit dieser Angabe können Sie erreichen, dass der Ausdruck des Reports mit einer bestimmten Seite endet.
JobName	Der Name des Druckauftrags muss innerhalb eines Projekts eindeutig sein und den Windows Konventionen entsprechen.
LayoutName	Über den Layoutnamen weisen Sie dem Druckauftrag ein Layout zu.
PrinterName 1	Hier legen Sie den Drucker fest, auf dem zuerst versucht wird einen Ausdruck zu starten.
PrinterName 2	Hier legen Sie den Drucker fest, der angesprochen wird, wenn der erste Drucker nicht verfügbar ist.
PrinterName 3	Hier legen Sie den Drucker fest, der angesprochen wird, wenn die beiden ersten Drucker nicht verfügbar sind.
RelativeSelectionCount	
RelativeSelectionRange	
StartPage	Mit dieser Angabe können Sie erreichen, dass der Report erst ab einer bestimmten Seite ausgedruckt wird.
StartTime	
UseRelative	
UseOutputFile	
UseOutputPrinter	

2.7.1.3 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

Fehlermeldungen CS

ERR_NOERROR	0	Kein Fehler
ERR_ILLEGALPROJECT	1	ungültiger Projektname / Projektpfad
ERR_NOMEMORY	2	Speicherfehler
ERR_UNKNOWNERROR	3	unbekannter Fehler
ERR_THREADNOTINITIALIZED	4	Der verwendete Thread ist nicht initialisiert.
ERR_OLEEXCEPTION	5	Fehler im Zusammenhang mit der MFC und OLE
ERR_NOOLESERVERAVAIL	6	kein OLE-Server verfügbar

Fehlermeldungen RT

ERR_NOERROR	0	Kein Fehler
ERR_ILLEGALPROJECT	1	ungültiger Projektname / Projektpfad
ERR_NOMEMORY	2	Speicherfehler
ERR_UNKNOWNERROR	3	unbekannter Fehler
ERR_NULLHANDLE	4	Handle konnte nicht angelegt werden
ERR_ILLEGALPOINTER	5	Zeiger fehlerhaft oder ungültig
ERR_ILLEGALJOBINDEX	6	fehlerhafter Index eines Druckauftrags
ERR_UNKNOWNPROPERTY	7	unbekannte Druckauftragseigenschaft
ERR_UNKNOWNMETHOD	8	unbekannte Druckauftragsmethode

2.7.1.4 Liste der Objekteigenschaften (RT Professional)**Properties für alle Objekte****Hinweis**

Die Propertynamen mit grober Beschreibung können auch über die kontextsensitive Hilfe mit Rechtsklick der Maus in der Eigenschaften-Box zu dem betreffenden Objekt im Report-Designer abgefragt werden.

Propertyname	Datentyp
"Width"	VT_I4
"Height"	VT_I4
"ObjectName"	VT_BSTR
"PageBreak"	VT_I4
"BorderStyle"	VT_I4

"BorderWidth"	VT_I4
"BorderBackColor"	VT_I4
"BorderColor"	VT_I4

"Left"	VT_I4
"Top"	VT_I4

Properties für alle "2 dimensionalen Objekte":

"FillStyle"	VT_I4
"BackColor"	VT_I4
"FillColor"	VT_I4

Properties für "dynamische Objekte":

"DataLink"	VT_BSTR
------------	---------

Die nachfolgenden Properties sind objektspezifisch!

Properties für alle "System"-Objekte:

Propertyname	Datentyp
"FontName"	VT_BSTR
"FontSize"	VT_I4
"FontBold"	VT_I4
"FontItalic"	VT_I4
"FontUnderline"	VT_I4
"Format"	VT_BSTR
"AlignmentLeft"	VT_I4
"AlignmentTop"	VT_I4
"WordWrap"	VT_I4

Properties für "dynamische Tabellen":

"FontName"	VT_BSTR
"FontSize"	VT_I4
"FontBold"	VT_I4
"FontItalic"	VT_I4
"FontUnderline"	VT_I4
"Orientation"	VT_I4
"Columns"	VT_BSTR
"ForeColor"	VT_I4
"List"	VT_I4

Properties für "dynamische Texte":

"FontName"	VT_BSTR
"FontSize"	VT_I4
"FontBold"	VT_I4
"FontItalic"	VT_I4
"FontUnderline"	VT_I4
"Orientation"	VT_I4
"AlignmentLeft"	VT_I4
"AlignmentTop"	VT_I4
"WordWrap"	VT_I4
"ForeColor"	VT_I4

Properties für "dynamische Picture":

"DynHeight"	VT_I4
-------------	-------

Properties für "Kreisbogen":

"Radius"	VT_I4
"StartAngle"	VT_I4
"EndAngle"	VT_I4

Properties für "Ellipsenbogen":

"StartAngle"	VT_I4
"EndAngle"	VT_I4
"RadiusWidth"	VT_I4
"RadiusHeight"	VT_I4

Properties für "Kreise":

"Radius"	VT_I4
----------	-------

Properties für "Ellipsen":

"RadiusWidth"	VT_I4
"RadiusHeight"	VT_I4

Properties für "Polygone":

"Polyline"	VT_BSTR
"PointCount"	VT_I4
"Index"	VT_I4
"ActualPointLeft"	VT_I4
"ActualPointTop"	VT_I4

Properties für "Kreissegmente":

"Radius"	VT_I4
"StartAngle"	VT_I4
"EndAngle"	VT_I4

Properties für "Ellipsensegmente":

"StartAngle"	VT_I4
"EndAngle"	VT_I4
"RadiusWidth"	VT_I4
"RadiusHeight"	VT_I4

Properties für "Polyline":

"Polyline"	VT_BSTR
"PointCount"	VT_I4
"Index"	VT_I4
"ActualPointLeft"	VT_I4
"ActualPointTop"	VT_I4

Properties für "Rundrechteck":

"RoundCornerWidth"	VT_I4
"RoundCornerHeight"	VT_I4

Properties für "Textobjekte":

"FontName"	VT_BSTR
"FontSize"	VT_I4
"FontBold"	VT_I4
"FontItalic"	VT_I4
"FontUnderline"	VT_I4

"Orientation"	VT_I4
"Text"	VT_BSTR
"AlignmentLeft"	VT_I4
"AlignmentTop"	VT_I4
"WordWrap"	VT_I4
"ForeColor"	VT_I4

Properties für "eingefügte Layouts":

"LayoutFileName"	VT_BSTR
------------------	---------

Properties für "Layouts":

"PrintMarginLeft"	VT_I4
"PrintMarginRight"	VT_I4
"PrintMarginTop"	VT_I4
"PrintMarginBottom"	VT_I4
"PaperSize"	VT_BSTR
"FirstPage"	VT_I4
"LastPage"	VT_I4
"Orientation"	VT_I4
"Printer1"	VT_BSTR
"Printer2"	VT_BSTR
"Printer3"	VT_BSTR
"BackColor"	VT_I4
"BorderColor"	VT_I4
"FillColor"	VT_I4
"DynMarginLeft"	VT_I4
"DynMarginRight"	VT_I4
"DynMarginTop"	VT_I4
"DynMarginBottom"	VT_I4
"Context"	VT_BSTR

Properties für "MetaFiles":

"MetaFileName"	VT_BSTR
----------------	---------

Properties für "Variablen"-Objekte:

"Tag"	VT_BSTR
"FontSize"	VT_I4
"FontName"	VT_BSTR
"FontBold"	VT_I4

"FontItalic"	VT_I4
"FontUnderline"	VT_I4
"AlignmentLeft"	VT_I4
"AlignmentTop"	VT_I4
"WordWrap"	VT_I4
"ForeColor"	VT_I4
"OutFormat"	VT_BSTR
"DataType"	VT_I4
"Calculation"	VT_VARIANT

2.7.1.5 Allgemeine Vorgehensweise beim Bearbeiten von Druckauftragseigenschaften (Report Designer) (RT Professional)

Verbindung zu RPJAPI.DLL herstellen und initialisieren

RPJAttach.

Handle auf Druckauftragseigenschaften erzeugen

Um die Eigenschaften eines Druckauftrags bearbeiten zu können, muss ein Handle auf den Druckauftrag erzeugt werden. Über dieses Handle wird ein Speicherbereich identifiziert, über den die Eigenschaften des Druckauftrags angesprochen werden können.

Funktion: RPJCreatePropertyHandle.

Druckauftragseigenschaften lesen

Die Eigenschaften der Druckaufträge sind in der Projektdatenbank gespeichert. Um die Eigenschaften bearbeiten zu können, müssen diese aus der Datenbank in der durch das Handle spezifizierten Bereich geladen werden.

Funktion: RPJGetJobProps.

Druckauftragseigenschaften ermitteln

Eine bestimmte Druckauftragseigenschaft wird über deren Namen angesprochen. Um den Wert einer Eigenschaft zu ermitteln muss diese gelesen werden.

Funktion: RPJGetProperty.

Druckauftragseigenschaften festlegen

Eine bestimmte Druckauftragseigenschaft wird über deren Namen angesprochen. Das Setzen der Eigenschaft erfolgt über Aufruf der Funktion RPJSetProperty.

Druckauftragseigenschaften abspeichern

Ablegen der geänderten Druckauftragseigenschaften in der Projektdatenbank. Es muss der Name des Druckauftrags angegeben werden, der gespeichert werden soll.

Funktion: RPJSetJobProps.

Druckauftragseigenschaft neu initialisieren

Den durch den Handle referenzierten Speicherbereich neu initialisieren, ohne den Handle freizugeben.

Funktion: RPJPropertyClear.

Druckauftragseigenschaften festlegen

Eine bestimmte Druckauftragseigenschaft wird über deren Namen angesprochen. Das Setzen der Eigenschaft erfolgt über Aufruf der.

Funktion: RPJSetProperty.

Druckauftragseigenschaften abspeichern

Ablegen der geänderten Druckauftragseigenschaften in der Projektdatenbank. Es muss der Name des Druckauftrags angegeben werden, der gespeichert werden soll.

Funktion: RPJSetJobProps.

Handle löschen

Handle einer Druckauftragseigenschaft löschen. Jedes mit RPJCreatePropertyHandle angelegten Handle sollte, wenn es nicht mehr benötigt wird, gelöscht werden.

Funktion: RPJDeletePropertyHandle.

Verbindung zu RPJAPI.DLL wieder abbauen

RPJDetach.

Beispiel: Anstoßen eines Druckauftrags als Vorschau

Allgemein

Um die Eigenschaften eines Druckauftrags bearbeiten zu können, muss ein Handle auf den Druckauftrag eingerichtet werden. Über dieses Handle wird ein Speicherbereich identifiziert, über den die Eigenschaften des Druckauftrags angesprochen werden können.

Verbindung zu RPJAPI.DLL herstellen und initialisieren

RPJAttach.

Handle erzeugen:Funktion

RPJCreatePropertyHandle.

Eigenschaften des Druckauftrags einlesen

Die Eigenschaften des Druckauftrags: "Rückdokumentation Control Center" werden aus der Datenbank in den durch den Handle: hProp spezifizierten Bereich gelesen.

Funktion: RPJGetJobProps.

Druckauftragsmethode anstoßen

Der Druckauftrag wird mit Angabe der Methode PREVIEW angestoßen. Der Druckauftrag selbst ist über das Handle: hProp spezifiziert.

Funktion: RPJCallJobMethod.

Handle löschen

Handle einer Druckauftragseigenschaft löschen. Jedes mit RPJCreatePropertyHandle angelegte Handle sollte, wenn es nicht mehr benötigt wird, gelöscht werden.

Funktion: RPJDeletePropertyHandle.

Verbindung zu RPJAPI.DLL wieder abbauen

RPJDetach.

Beispiel

```
void DruckAuftrag(void)
{
    CMN_ERROR err; //error structure
    BOOL ret;
    HPROPERTIES hProp;
    char jobname[200];
    char method[200];
    LPCSTR szProjectName = "c:\\rest\\test.mcp"; //name of project
    strcpy(method, "PREVIEW");
    strcpy(jobname, "Backdokumentation Control Center");
    ret = RPJAttach(&err);
    if (TRUE == ret)
    {
        hProp = RPJCreatePropertyHandle(szProjectName, &err);
        if (NULL == hProp)
        {
            ErrMsg("Error RPJCreatePropertyHandle", &err);
        }
        else
        {
            ret = RPJGetJobProps(hProp, jobname, &err);
            if (FALSE == ret)
            {
                ErrMsg("Error RPJGetJobProbs", &err);
            }
            else
            {
                ret = RPJCallJobMethod(hProp, method, &err);
                if (FALSE == ret)
                {
                    ErrMsg("Error executing RPJCallJobMethod", &err);
                }
                else
                {
                    Msg("Print job started.");
                }
            }
            RPJDeletePropertyHandle(hProp, &err);
        }
        RPJDetach(&err);
    }
    else
    {
        Msg("No connection to report designer DLL");
    }
}
```

2.7.2 Funktionen zum Verbindungsaufbau (RT Professional)

2.7.2.1 RPJAttach (RT Professional)

Beschreibung

Mit der Funktion wird eine Verbindung zu RPJAPI.DLL aufgebaut und initialisiert.

Deklaration

```
BOOL RPJAttach (
    CMN_ERROR*   pcmnerror );
```

Parameter

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung hergestellt.

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJDetach (Seite 737)	Verbindung abbauen
-----------------------	--------------------

Siehe auch

RPJDetach (Seite 737)

Übersicht über die Funktionen (Seite 723)

2.7.2.2 RPJDetach (RT Professional)

Beschreibung

Mit der Funktion wird eine bestehende Verbindung zu RPJAPI.DLL wieder abgebaut.

Deklaration

```
BOOL RPJDetach (
    CMN_ERROR*   pcmnerror );
```

Parameter

pcmerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung ist wieder abgebaut.

FALSE

Fehler

Bemerkung

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX, ...) benutzt werden. Aufgrund von Microsoft-spezifischen Mechanismen kann das gegebenenfalls zum Hängenbleiben des Aufrufs und damit des Programms führen.

Benötigte Dateien

rpjapi.h
rpjapi.lib
rpjapi.dll

Verwandte Funktionen

RPJAttach (Seite 736)	Verbindung aufbauen
-----------------------	---------------------

Siehe auch

RPJAttach (Seite 736)

Übersicht über die Funktionen (Seite 723)

2.7.2.3 RPJMemFree (RT Professional)

Beschreibung

Mit der Funktion wird eine bestehende Verbindung zu RPJAPI.DLL wieder abgebaut und der allokierte Speicher wird freigegeben.

Deklaration

```
BOOL RPJMemFree (  
    const PVOID    pvMemBlock,  
    CMN_ERROR*     pcmnerror );
```

Parameter

pvMemBlock

Zeiger auf einen freizugebenden Speicherbereich, der zuvor von einer RPJ-Funktion allokiert wurde.

Wenn pvMemBlock = NULL ist, wird nichts ausgeführt.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Der Speicherbereich wurde freigegeben.

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Siehe auch

Übersicht über die Funktionen (Seite 723)

2.7.3 Funktionen zum Bearbeiten von Projekteigenschaften (RT Professional)**2.7.3.1 RPJGetNumProjectProperties (RT Professional)****Beschreibung**

Die Funktion ermittelt die Anzahl der Eigenschaften des Projekts.

Deklaration

```
BOOL RPJGetNumProjectProperties (  
    LPCSTR      pszReserved,  
    DWORD*     pdwNumProperties  
    CMN_ERROR* pcmnerror );
```

Parameter**pszReserved**

Der Parameter ist für spätere Erweiterungen reserviert.

pdwNumProperties

Zeiger auf eine DWORD Variable, in die die Anzahl der Eigenschaften zurückgeliefert wird.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Die Anzahl der Eigenschaften wurde ermittelt.

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib
rpjapi.dll

Siehe auch

Übersicht über die Funktionen (Seite 723)

2.7.3.2 RPJGetPropertyAt (RT Professional)

Beschreibung

Die Funktion ermittelt Name und Typ einer Eigenschaft anhand des Index im Projekt.

Deklaration

```
BOOL RPJGetPropertyAt (  
    LPCSTR      pszReserved,  
    DWORD       dwPropIndex,  
    LPSTR       pszBuffer,  
    DWORD       dwCharMax,  
    DWORD*      pdwPrpType,  
    CMN_ERROR*  pcmnerror );
```

Parameter

pszReserved

Der Parameter ist für spätere Erweiterungen reserviert.

dwPropIndex

Index der Eigenschaft im Projekt, von der die Informationen zurückgeliefert werden sollen.

pszBuffer

Zeiger auf einen Puffer, in dem der Name der Eigenschaft zurückgegeben werden soll. Die Größe des Puffers wird mit dwCharMax angegeben.

dwCharMax

Maximale Größe des Puffers der bei pszBuffer angegeben ist

pdwPropType

Zeiger auf eine DWORD Variable, in die der Typ der Eigenschaft im Projekt zurückgegeben wird.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Der Name und Type der Eigenschaft im Projekt wurde ermittelt.

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Siehe auch

Übersicht über die Funktionen (Seite 723)

2.7.3.3 RPJGetProperty (RT Professional)

Beschreibung

Die Funktion ermittelt den Wert einer Eigenschaft. Der Projektname und der Eigenschaftsname müssen übergeben werden.

Deklaration

```
BOOL RPJGetProperty (
    LPCSTR      pszProjectName,
    LPCSTR      pszPropName,
    PVOID       pvPropValue,
    VARTYPE     vtPropType,
    DWORD       dwBufferSize,
    CMN_ERROR*  pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pszPropName

Name der Eigenschaft im Projekt.

pvPropValue

Zeiger auf einen Puffer, in dem der Wert der Eigenschaft zurückgegeben werden soll. Die Größe des Puffers in pvPropValue wird mit dwBufferSize angegeben.

vtPropType

Typ, in dem der Wert der Eigenschaft im Puffer pvPropValue erwartet wird.

VT_I4	Integer oder Boolean
VT_LPSTR	Text in ANSI-Form
VT_LPWSTR	Text in UNICODE-Form
VT_DATE	Datum/Uhrzeit

dwBufferSize

Größe des Puffers in pvPropValue in den der Wert der Eigenschaft zurückgegeben werden soll. Die Größe darf nur in BYTE und nicht in CHAR angegeben werden.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Der Wert der Eigenschaft wurde ermittelt.

FALSE

Fehler

Benötigte Dateien

- rpjapi.h
- rpjapi.lib
- rpjapi.dll

Siehe auch

Übersicht über die Funktionen (Seite 723)

2.7.3.4 RPJProjectLock (RT Professional)

Beschreibung

Der Zugriff auf die Druckauftragsliste im Projekt wird für andere gesperrt. Der Name des Sperrenden muss in pszLockerNameNew angegeben werden. Bei einem Zugriffskonflikt wird der Name des Sperrenden über ppszLockerNameCur zurückgegeben.

Deklaration

```

BOOL RPJProjectLock (
    LPCSTR      pszProjectName,
    BOOL        fWriteLock,
    BOOL        fDoNotWait,
    LPCSTR      pszLockerNameNew,
    LPSTR*      ppszLockerNameCur,
    CMN_ERROR*  pcmnerror );

```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

fWriteLock

Legt die Art der Sperrung fest.

TRUE	Sperrt für Schreiben und Lesen.
FALSE	Sperrt nur für Lesen.

fDoNotWait

TRUE	Wenn eine Sperre existiert wird sofort ein Fehler gemeldet
FALSE	Wenn eine Sperre existiert, wird diese zunächst zyklisch geprüft. Nach einigen Sekunden wird mit Timeout-Fehler abgebrochen.

pszLockerNameNew

Enthält den Namen des Sperrenden. Bei einem Zugriffskonflikt wird dieser Name über ppszLockerNameCur zurückgegeben.

ppszLockerNameCur

Bei einem Zugriffskonflikt wird in ppszLockerNameCur der Name des aktuell Sperrenden zurückgegeben.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Druckauftragsliste gesperrt.

FALSE

Es trat ein Zugriffskonflikt auf. Die Druckauftragsliste des Projekts ist bereits gesperrt. Der Name des Sperrenden wird über ppszLockerNameCur zurückgegeben.

Benötigte Dateien

rpjapi.h
rpjapi.lib
rpjapi.dll

Verwandte Funktionen

RPJProjectUnlock (Seite 744)	Eine Sperre aufheben
RPJProjectUnlockAll (Seite 745)	Alle Sperren aufheben

Siehe auch

RPJProjectUnlock (Seite 744)
RPJProjectUnlockAll (Seite 745)
Übersicht über die Funktionen (Seite 723)

2.7.3.5 RPJProjectUnlock (RT Professional)

Beschreibung

Die Sperre einer Druckauftragsliste wird aufgehoben.

Deklaration

```
BOOL RPJProjectUnlock (  
    LPCSTR      pszProjectName,  
    CMN_ERROR*  pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Sperre der Druckauftragsliste wurde aufgehoben.

FALSE

Es trat ein Zugriffskonflikt auf.

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJProjectLock (Seite 742)	Sperre anlegen
RPJProjectUnlockAll (Seite 745)	Alle Sperren aufheben

Siehe auch

RPJProjectLock (Seite 742)

RPJProjectUnlockAll (Seite 745)

Übersicht über die Funktionen (Seite 723)

2.7.3.6 RPJProjectUnlockAll (RT Professional)

Beschreibung

Alle Sperre einer Druckauftragsliste werden aufgehoben.

Deklaration

```
BOOL RPJProjectUnlockAll (  
    LPCSTR      pszProjectName,  
    CMN_ERROR*  pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Alle Sperren der Druckauftragsliste wurden aufgehoben.

FALSE

Es trat ein Zugriffskonflikt auf.

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJProjectLock (Seite 742)	Sperre anlegen
RPJProjectUnlock (Seite 744)	Eine Sperre aufheben

Siehe auch

RPJProjectLock (Seite 742)

RPJProjectUnlock (Seite 744)

Übersicht über die Funktionen (Seite 723)

2.7.4 Funktionen zum Bearbeiten von Druckaufträgen (RT Professional)

2.7.4.1 RPJCreateJob (RT Professional)

Beschreibung

Legt einen neuen Druckauftrag an.

Deklaration

```
BOOL RPJCreateJob (  
    LPCSTR      pszProjectName,  
    LPCSTR      pszJobName,  
    CMN_ERROR*  pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pszJobName

Zeiger auf den Namen des anzulegenden Druckauftrags.

pcmncerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Druckauftrag angelegt.

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJDeleteJob (Seite 747)	Druckauftrag löschen
--------------------------	----------------------

Siehe auch

RPJDeleteJob (Seite 747)

Übersicht über die Funktionen (Seite 723)

2.7.4.2 RPJDeleteJob (RT Professional)

Beschreibung

Ein durch pszJobName spezifizierter Druckauftrag wird gelöscht.

Deklaration

```
BOOL RPJDeleteJob (  
    LPCSTR      pszProjectName,  
    LPCSTR      pszJobName,  
    CMN_ERROR*  pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pszJobName

Zeiger auf den Namen des zu löschenden Druckauftrags.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Druckauftrag gelöscht.

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJCreateJob (Seite 746)	Druckauftrag anlegen
--------------------------	----------------------

Siehe auch

RPJCreateJob (Seite 746)

Übersicht über die Funktionen (Seite 723)

2.7.4.3 RPJCreatePropertyHandle (RT Professional)

Beschreibung

Um die Eigenschaften eines Druckauftrags bearbeiten zu können, müssen Sie vorher mit dieser Funktion ein Handle auf den Druckauftrag eingerichtet haben. Das Handle erhalten Sie im Rückgabewert geliefert.

Deklaration

```
HPROPERTIES RPJCreatePropertyHandle (
    LPCSTR      pszProjectName,
    CMN_ERROR*  pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

Handle auf die Druckauftragseigenschaft.

Bei Fehler ist der Rückgabewert NULL.

Bemerkung

Der hier gelieferte Handle wird z.B. bei den Funktionen RPJGetJobProps, RPJGetProperty und RPJSetProperty und benötigt.

Benötigte Dateien

rpjapi.h
rpjapi.lib
rpjapi.dll

Verwandte Funktionen

RPJDeletePropertyHandle (Seite 750)	Handle löschen
RPJPropertyClear (Seite 769)	Speicherbereich initialisieren

RPJGetJobProps (Seite 764)	Druckauftragseigenschaften ermitteln
RPJGetProperty (Seite 766)	Druckauftragseigenschaft ermitteln
RPJSetProperty (Seite 771)	Druckauftragseigenschaften festlegen

Beispiele

- Show print job preview (Seite 788) "RD02.cpp"
- Modify print job properties (Seite 783) "RD02.cpp"
- Get print job names (Seite 777) "RD02.cpp"
- Get print job methodname (Seite 774) "RD02.cpp"
- Get print job properties (Seite 780) "RD02.cpp"

Siehe auch

- Get print job methodname (Seite 774)
- Get print job names (Seite 777)
- Get print job properties (Seite 780)
- Modify print job properties (Seite 783)
- Show print job preview (Seite 788)
- RPJDeletePropertyHandle (Seite 750)
- RPJGetProperty (Seite 766)
- RPJGetJobProps (Seite 764)
- RPJSetProperty (Seite 771)
- RPJPropertyClear (Seite 769)
- Übersicht über die Funktionen (Seite 723)

2.7.4.4 RPJDeletePropertyHandle (RT Professional)

Beschreibung

Jedes mit RPJCreatePropertyHandle angelegte Handle sollte, wenn es nicht mehr benötigt wird, mit dieser Funktion gelöscht werden.

Deklaration

```
BOOL RPJDeletePropertyHandle (  
    HPROPERTIES    hproperties,  
    CMN_ERROR*     pcmnerror );
```

Parameter

hproperties

Das Handle der Druckauftrageigenschaft wird durch die Funktion RPJCreatePropertyHandle angelegt.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Handle gelöscht

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJCreatePropertyHandle (Seite 749)	Handle anlegen
-------------------------------------	----------------

Beispiele

Show print job preview (Seite 788) "RD02.cpp"

Modify print job properties (Seite 783) "RD02.cpp"

Get print job names (Seite 777) "RD02.cpp"

Get print job methodname (Seite 774) "RD02.cpp"

Get print job properties (Seite 780) "RD02.cpp"

Siehe auch

RPJCreatePropertyHandle (Seite 749)

Get print job methodname (Seite 774)

Get print job names (Seite 777)

Get print job properties (Seite 780)

- Modify print job properties (Seite 783)
- Show print job preview (Seite 788)
- Übersicht über die Funktionen (Seite 723)

2.7.4.5 RPJGetJobNameAt (RT Professional)

Beschreibung

Ermittelt den Namen eines durch dwJobIndex spezifizierten Druckauftrags.

Deklaration

```
BOOL RPJGetJobNameAt (  
    LPCSTR      pszProjectName,  
    DWORD       dwJobIndex,  
    LPSTR       pszBuffer,  
    DWORD       dwCharMax,  
    CMN_ERROR*  pcmnerror );
```

Parameter

PszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

dwJobIndex

Index des Druckauftrags, dessen Name ermittelt werden soll.

pszBuffer

Zeiger auf den Empfangspuffer, in dem der Name des Druckauftrags abgelegt werden soll.

dwCharMax

Größe des Empfangspuffers in Anzahl Zeichen inklusive der Null-Terminierung. Dieser Wert muss größer 1 sein.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Name des Druckauftrags ermittelt.

FALSE

Fehler

Bemerkung

Ein durch RPJGetJobNameAt angezeigter Index-Fehler ist möglicherweise darauf zurückzuführen, dass sich die Anzahl der Druckaufträge nach dem Aufruf von RPJGetNumJobs verändert hat.

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Beispiele

Get print job names (Seite 777) "RD02.cpp"

Siehe auch

Get print job names (Seite 777)

Übersicht über die Funktionen (Seite 723)

2.7.4.6 RPJGetNumJobs (RT Professional)

Beschreibung

Ermittelt die Anzahl der aktuell anstehenden Druckaufträge. Nachdem Aufruf der Funktion kann sich die Anzahl durch Löschen oder Hinzufügen von Druckaufträgen ändern.

Deklaration

```
BOOL RPJGetNumJobs (  
    LPCSTR      pszProjectName,  
    DWORD*     pdwNumJobs,  
    CMN_ERROR* pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname, Extension.

pdwNumJobs

Zeiger auf die Speicherstelle an der die Anzahl der Druckaufträge abgelegt werden soll.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Anzahl der Druckaufträge ermittelt.

FALSE

Fehler

Benötigte Files

rpjapi.h

rpjapi.lib

rpjapi.dll

Beispiele

Get print job names (Seite 777) "RD02.cpp"

Siehe auch

Get print job names (Seite 777)

Übersicht über die Funktionen (Seite 723)

2.7.4.7 RPJJobLock (RT Professional)

Beschreibung

Der Zugriff auf den angegebenen Druckauftrag wird für andere gesperrt. Der Name des Sperrenden muss in pszLockerNameNew angegeben werden. Bei einem Zugriffskonflikt wird der Name des Sperrenden über ppszLockerNameCur zurückgegeben.

Deklaration

```

BOOL RPJJobLock (
    LPCSTR    pszProjectName,
    LPCSTR    pszJobName,
    BOOL      fWriteLock,
    BOOL      fDoNotWait,
    LPCSTR    pszLockerNameNew,
    LPSTR*    ppszLockerNameCur,
    CMN_ERROR* pcmnerror );

```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pszJobName

Zeiger auf den Namen des zu sperrenden Druckauftrags.

fWriteLock

Legt die Art der Sperrung fest.

TRUE	Sperrt für Schreiben und Lesen.
FALSE	Sperrt nur für Lesen.

fDoNotWait

TRUE	Wenn eine Sperre existiert wird sofort ein Fehler gemeldet
FALSE	Wenn eine Sperre existiert, wird diese zunächst zyklisch geprüft. Nach einigen Sekunden wird mit Timeout-Fehler abgebrochen.

pszLockerNameNew

Enthält den Namen des Sperrenden. Bei einem Zugriffskonflikt wird dieser Name über ppszLockerNameCur zurückgegeben.

ppszLockerNameCur

Bei einem Zugriffskonflikt wird in ppszLockerNameCur der Name des aktuell Sperrenden zurückgegeben.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Druckauftrag gesperrt.

FALSE

Es trat ein Zugriffskonflikt auf. Die Druckauftragsliste des Projekts ist bereits gesperrt. Der Name des Sperrenden wird über ppszLockerNameCur zurückgegeben.

Benötigte Dateien

- rpjapi.h
- rpjapi.lib
- rpjapi.dll

Verwandte Funktionen

RPJJobUnlock (Seite 756)	Eine Sperre aufheben
RPJJobUnlockAll (Seite 757)	Alle Sperren aufheben

Siehe auch

- RPJJobUnlockAll (Seite 757)
- RPJJobUnlock (Seite 756)
- Übersicht über die Funktionen (Seite 723)

2.7.4.8 RPJJobUnlock (RT Professional)

Beschreibung

Die Sperre eines Druckauftrags wird aufgehoben.

Deklaration

```
BOOL RPJJobUnlock (
    LPCSTR      pszProjectName,
    LPCSTR      pszJobName,
    CMN_ERROR*  pcmnerror );
```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pszJobName

Zeiger auf den Namen des Druckauftrags.

pcmerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Die Sperren des Druckauftrags wurde aufgehoben.

FALSE

Es trat ein Zugriffskonflikt auf.

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJJobLock (Seite 754)	Sperre anlegen
RPJJobUnlockAll (Seite 757)	Alle Sperren aufheben

Siehe auch

RPJJobUnlockAll (Seite 757)

RPJJobLock (Seite 754)

Übersicht über die Funktionen (Seite 723)

2.7.4.9 RPJJobUnlockAll (RT Professional)**Beschreibung**

Alle Sperren des angegebenen Druckauftrags werden aufgehoben.

Deklaration

```

BOOL RPJJobUnlockAll (
    LPCSTR      pszProjectName,
    LPCSTR      pszJobName,
    CMN_ERROR*  pcmerror );

```

Parameter

pszProjectName

Zeiger auf den Namen des Projekts, bestehend aus Pfadangabe, Projektname und Extension.

pszJobName

Zeiger auf den Namen des Druckauftrags.

pcmnererror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Alle Sperren des angegebenen Druckauftrags wurden aufgehoben.

FALSE

Es trat ein Zugriffskonflikt auf

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJJobLock (Seite 754)	Sperre anlegen
RPJJobUnlock (Seite 756)	Eine Sperre aufheben

Siehe auch

RPJJobUnlock (Seite 756)

RPJJobLock (Seite 754)

Übersicht über die Funktionen (Seite 723)

2.7.5 Funktionen zum Bearbeiten von Druckauftragsmethoden (RT Professional)

2.7.5.1 RPJCallJobMethod (RT Professional)

Beschreibung

Mit dieser Funktion können Sie eine bestimmte Auftragsmethode festlegen. Derzeit werden zwei Methoden unterschieden: PrintJob und PreviewJob.

Deklaration

```
RPJCallJobMethod (
    HPROPERTIES    hproperties,
    LPCSTR         pszMethodName
    CMN_ERROR*    pcmnerror );
```

Parameter

hproperties

Das Handle der Druckauftragseigenschaft wird durch die Funktion RPJCreatePropertyHandle angelegt.

pszMethodName

Zeiger auf den Namen der zu verwendenden Auftragsmethode.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Druckauftragsmethode festgelegt

FALSE

Fehler

Bemerkung

Um mit RPJCallJobMethod eine Druckauftragsmethode festlegen zu können, muss vorher die Funktion RPJGetJobProps ausgeführt worden sein.

Benötigte Dateien

rpjapi.h
rpjapi.lib
rpjapi.dll

Beispiele

Show print job preview (Seite 788) "RD02.cpp"

Siehe auch

Show print job preview (Seite 788)
Übersicht über die Funktionen (Seite 723)

2.7.5.2 RPJGetMethodAt (RT Professional)

Beschreibung

Die Funktion ermittelt zu den Namen der in dwMethodIndex spezifizierten Druckmethode.

Deklaration

```
BOOL RPJGetMethodAt (  
    DWORD          dwMethodIndex,  
    LPSTR          pszBuffer,  
    DWORD          dwCharMax,  
    CMN_ERROR*     pcmnerror );
```

Parameter

dwMethodIndex

Index der Druckmethode deren Name ermittelt wird.

pszBuffer

Zeiger auf einen Puffer, in dem der Name der Druckmethode abgelegt werden soll.

dwCharMax

Größe des Puffers, der den Namen aufnehmen soll in Anzahl Zeichen inklusive Null-Terminierung. Dieser Wert muss größer 1 sein.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Name der Druckauftragsmethode ermittelt

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Beispiele

Get print job methodname (Seite 774) "RD02.cpp"

Siehe auch

Get print job methodname (Seite 774)

Übersicht über die Funktionen (Seite 723)

2.7.5.3 RPJGetNumJobMethods (RT Professional)

Beschreibung

Die Funktion ermittelt, wieviele Druckauftragsmethoden derzeit zur Verfügung stehen.

Deklaration

```
BOOL RPJGetNumJobMethods (  
    DWORD* pdwNumMethods,  
    CMN_ERROR* pcmnerror );
```

Parameter

pdwNumMethods

Zeiger auf eine Speicherstelle, an der die Anzahl der Druckauftragsmethoden hinterlegt werden soll.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Anzahl der Druckauftragsmethoden ermittelt

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Beispiele

Get print job methodname (Seite 774) "RD02.cpp"

Siehe auch

Get print job methodname (Seite 774)

Übersicht über die Funktionen (Seite 723)

2.7.6 Funktionen zum Bearbeiten von Druckauftragseigenschaften (RT Professional)

2.7.6.1 RPJGetJobPropertyAt (RT Professional)

Beschreibung

Ermittelt den Namen und den Typ der in dwPropindex spezifizierten Auftragseigenschaft.

Deklaration

```
BOOL RPJGetJobPropertyAt (  
    DWORD          dwPropIndex,  
    LPSTR          pszBuffer,  
    DWORD          dwCharMax,  
    DWORD*         pdwPropType,  
    CMN_ERROR*     pcmnerror );
```

Parameter

dwPropIndex

Index der Druckauftragseigenschaften, deren Name ermittelt werden soll.

pszBuffer

Zeiger auf einen Puffer, der den Namen der Auftragseigenschaft entgegennimmt.

dwCharMax

Größe des Empfangspuffers in Anzahl Zeichen inklusive der Null-Terminierung. Dieser Wert muss größer 1 sein.

pdwPropType

Zeiger auf eine Speicherstelle, an der der Typ der Auftragseigenschaft abgelegt wird.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Name der Druckauftragseigenschaft ermittelt

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Beispiele

Get print job properties (Seite 780) "RD02.cpp"

Siehe auch

Get print job properties (Seite 780)

Übersicht über die Funktionen (Seite 723)

2.7.6.2 RPJGetJobProps (RT Professional)

Beschreibung

Die Funktion ermittelt die Eigenschaften eines Druckauftrags und hinterlegt sie im internen Puffer hProps.

Deklaration

```
BOOL RPJGetJobProps (  
    HPROPERTIES    hproperties,  
    LPCSTR         pszJobName,  
    CMN_ERROR*     pcmnerror );
```

Parameter

hproperties

Das Handle der Druckauftragseigenschaft wird durch die Funktion RPJCreatePropertyHandle angelegt.

pszJobName

Zeiger auf den Namen des Druckauftrags.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Bemerkung

Diese Funktion ist eine Voraussetzung für Funktionen wie RPJGetProperty und RPJSetProperty.

Rückgabewert

TRUE

Druckauftragseigenschaften ermittelt

FALSE

Fehler

Benötigte Dateien

rpjapi.h
 rpjapi.lib
 rpjapi.dll

Verwandte Funktionen

RPJCreatePropertyHandle (Seite 749)	Handle anlegen
RPJGetProperty (Seite 766)	Druckauftragseigenschaft ermitteln
RPJSetProperty (Seite 771)	Druckauftragseigenschaften festlegen

Beispiele

Show print job preview (Seite 788) "RD02.cpp"
 Modify print job properties (Seite 783) "RD02.cpp"

Siehe auch

RPJGetProperty (Seite 766)
 RPJSetProperty (Seite 771)
 Modify print job properties (Seite 783)
 Show print job preview (Seite 788)
 RPJCreatePropertyHandle (Seite 749)
 Übersicht über die Funktionen (Seite 723)

2.7.6.3 RPJGetNumJobProperties (RT Professional)**Beschreibung**

Ermittelt die Anzahl der verfügbaren Druckauftragseigenschaften.

Deklaration

```

BOOL RPJGetNumJobProperties (
    DWORD*          pdwNumProperties,
    CMN_ERROR*      pcmnerror );

```

Parameter

pdwNumProperties

Zeiger auf eine Speicherstelle, an der die Anzahl der Druckauftragseigenschaften abgelegt werden soll.

pcmerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Anzahl der Druckauftragseigenschaften ermittelt

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Beispiele

Get print job properties (Seite 780) "RD02.cpp"

Siehe auch

Get print job properties (Seite 780)

Übersicht über die Funktionen (Seite 723)

2.7.6.4 RPJGetProperty (RT Professional)

Beschreibung

Bevor Sie mit RPJGetProperty den Wert der die durch pszPropName spezifizierte Druckauftragseigenschaft ermitteln können, müssen Sie mit RPJGetJobProps die Auftragseigenschaften im internen Puffer hProp hinterlegt haben.

Deklaration

```

BOOL RPJGetProperty (
    HPROPERTIES    hproperties,
    LPCSTR         pszPropName,
    PVOID          pvPropValue,
    VARTYPE        vtPropType,
    DWORD          dwBufferSize,
    CMN_ERROR*     pcmnerror );

```

Parameter

hproperties

Das Handle der Druckauftragseigenschaft wird durch die Funktion RPJCreatePropertyHandle angelegt.

pszPropName

Zeiger auf den Namen der Druckauftragseigenschaft:

AbsoluteSelectionFrom	
AbsoluteSelectionTo	
CycleSpan	
DestinationFile	
EnableCycle	
EnableStart	
EndPage	Mit dieser Angabe können Sie erreichen, dass der Ausdruck des Protokolls mit einer bestimmten Seite endet.
JobName	Der Name des Druckauftrags muss innerhalb eines Projekts eindeutig sein und den Windows Konventionen entsprechen.
LayoutName	Über den Layoutnamen weisen Sie dem Druckauftrag ein Layout zu.
PrinterName 1	Hier legen Sie den Drucker fest, auf dem zuerst versucht wird einen Ausdruck zu starten.
PrinterName 2	Hier legen Sie den Drucker fest, der angesprochen wird, wenn der erste Drucker nicht verfügbar ist.
PrinterName 3	Hier legen Sie den Drucker fest, der angesprochen wird, wenn die beiden ersten Drucker nicht verfügbar sind.
RelativeSelectionCount	
RelativeSelectionRange	
StartPage	Mit dieser Angabe können Sie erreichen, dass das Protokoll erst ab einer bestimmten Seite ausgedruckt wird.
StartTime	
UseRelative	
UseOutputFile	
UseOutputPrinter	

pvPropValue

Zeiger auf einen Puffer, der den Namen entgegennehmen soll.

vtPropType

Der Typ der Objekteigenschaft ist abhängig von pszPropName:

AbsoluteSelectionFrom	VT_DATE
AbsoluteSelectionTo	VT_DATE
CycleSpan	VT_I4
DestinationFile	VT_LPSTR
EnableCycle	VT_I4
EnableStart	VT_I4
EndPage	VT_I4
JobName	VT_LPSTR / VT_LPWSTR
LayoutName	VT_LPSTR / VT_LPWSTR
PrinterName	VT_LPSTR / VT_LPWSTR
PrinterName2	VT_LPSTR / VT_LPWSTR
PrinterName3	VT_LPSTR / VT_LPWSTR
RelativeSelectionCount	VT_I4
RelativeSelectionRange	VT_I4
StartPage	VT_I4
StartTime	VT_DATE
UseRelative	VT_I4
UseOutputFile	VT_I4
UseOutputPrinter	VT_I4

Wenn Sie eine Zeichenkette verändern wollen und Ihre Applikation unter Unicode kompiliert wird, müssen Sie als Datentyp von vtProp VT_LPWSTR verwenden. Ansonsten ist vtProp vom Datentyp VT_LPTSTR.

Ist vtProp vom Datentyp VT_DATE so muss der Parameter prop auf eine SYSTEMTIME-Struktur verweisen.

dwBufferSize

Größe des Puffers, auf den prop verweist in Anzahl Bytes.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Druckauftragseigenschaft ermittelt

FALSE

Fehler

Bemerkung

Falls der Puffer für die Aufnahme der Auftragseigenschaft zu klein gewählt wird, denkbar bei z.B. Jobname, Layoutname, so wird der Puffer gefüllt, die Funktion liefert aber die Fehlermeldung ERR_UNKNOWNERROR zurück.

Fehlermeldungen

ERR_UNKNOWNERROR	Unbekannter Fehler
------------------	--------------------

Benötigte Dateien

rpjapi.h
 rpjapi.lib
 rpjapi.dll

Verwandte Funktionen

RPJCreatePropertyHandle (Seite 749)	Handle anlegen
RPJGetJobProps (Seite 764)	Druckauftragseigenschaften ermitteln
RPJSetProperty (Seite 771)	Druckauftragseigenschaften festlegen

Beispiele

Modify print job properties (Seite 783) "RD02.cpp"

Siehe auch

RPJGetJobProps (Seite 764)
 RPJSetProperty (Seite 771)
 RPJCreatePropertyHandle (Seite 749)
 Modify print job properties (Seite 783)
 Übersicht über die Funktionen (Seite 723)

2.7.6.5 RPJPropertyClear (RT Professional)**Beschreibung**

Die Verwendung der Funktion ist sinnvoll, wenn der durch ein Handle referenzierte Speicherbereich initialisiert werden soll, ohne dass das Handle gelöscht wird.

Deklaration

```
BOOL RPJPropertyClear (  
    HPROPERTIES    hproperties,  
    CMN_ERROR*     pcmnerror );
```

Parameter

hproperties

Das Handle der Druckauftragseigenschaft wird durch die Funktion RPJCreatePropertyHandle angelegt.

pcmnerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Speicher initialisiert.

FALSE

Fehler

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJCreatePropertyHandle (Seite 749)	Handle anlegen
-------------------------------------	----------------

Siehe auch

RPJCreatePropertyHandle (Seite 749)

Übersicht über die Funktionen (Seite 723)

2.7.6.6 RPJSetProperty (RT Professional)

Beschreibung

Mit dieser Funktion können Sie einzelne Druckauftragseigenschaften temporär verändern.

Bevor Sie mit RPJSetProperty den Wert der durch pszPropName spezifizierten Druckauftragseigenschaft neu setzen können, müssen Sie mit RPJGetJobProps die Auftragseigenschaften im internen Puffer hProp hinterlegt haben.

Deklaration

```

BOOL RPJSetProperty (
    HPROPERTIES      hproperties,
    LPCSTR           pszPropName,
    PVOID            pvPropValue,
    VARTYPE          vtPropType,
    DWORD            dwReserved,
    CMN_ERROR*       pcmnerror );

```

Parameter

hproperties

Handle der Druckauftragseigenschaften. Das Handle wird durch die Funktion RPJCreatePropertyHandle angelegt.

pszPropName

Zeiger auf den Namen der Druckauftragseigenschaft:

AbsoluteSelectionFrom	
AbsoluteSelectionTo	
CycleSpan	
DestinationFile	
EnableCycle	
EnableStart	
EndPage	Mit dieser Angabe können Sie erreichen, dass der Ausdruck des Protokolls mit einer bestimmten Seite endet.
JobName	Der Name des Druckauftrags muss innerhalb eines Projekts eindeutig sein und den Windows Konventionen entsprechen.
LayoutName	Über den Layoutnamen weisen Sie dem Druckauftrag ein Layout zu.
PrinterName 1	Hier legen Sie den Drucker fest, auf dem zuerst versucht wird einen Ausdruck zu starten.
PrinterName 2	Hier legen Sie den Drucker fest, der angesprochen wird, wenn der erste Drucker nicht verfügbar ist.
PrinterName 3	Hier legen Sie den Drucker fest, der angesprochen wird, wenn die beiden ersten Drucker nicht verfügbar sind.
RelativeSelectionCount	

RelativeSelectionRange	
StartPage	Mit dieser Angabe können Sie erreichen, dass das Protokoll erst ab einer bestimmten Seite ausgedruckt wird.
StartTime	
UseRelative	
UseOutputFile	
UseOutputPrinter	

pvPropValue

Zeiger auf den Wert der Objekteigenschaft.

vtPropType

Der Typ der Objekteigenschaft ist abhängig von pszPropName:

AbsoluteSelectionFrom	VT_DATE
AbsoluteSelectionTo	VT_DATE
CycleSpan	VT_I4
DestinationFile	VT_LPSTR
EnableCycle	VT_I4
EnableStart	VT_I4
EndPage	VT_I4
JobName	VT_LPSTR / VT_LPWSTR
LayoutName	VT_LPSTR / VT_LPWSTR
PrinterName	VT_LPSTR / VT_LPWSTR
PrinterName2	VT_LPSTR / VT_LPWSTR
PrinterName3	VT_LPSTR / VT_LPWSTR
RelativeSelectionCount	VT_I4
RelativeSelectionRange	VT_I4
StartPage	VT_I4
StartTime	VT_DATE
UseRelative	VT_I4
UseOutputFile	VT_I4
UseOutputPrinter	VT_I4

Wenn Sie eine Zeichenkette verändern wollen und Ihre Applikation unter Unicode kompiliert wird, müssen Sie als Datentyp von vtProp VT_LPWSTR verwenden. Ansonsten ist vtProp vom Datentyp VT_LPTSTR.

Ist vtProp vom Datentyp VT_DATE so muss der Parameter prop auf eine SYSTEMTIME-Struktur verweisen.

dwReserved

Der Parameter ist für spätere Erweiterungen reserviert.

pcmerror

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Druckauftragseigenschaft temporär geändert

FALSE

Fehler

Bemerkung

Über die Funktionen RPJGetNumJobProperties und RPJGetJobPropertyAt können alle verfügbaren Druckauftragseigenschaften ermittelt werden.

Benötigte Dateien

rpjapi.h

rpjapi.lib

rpjapi.dll

Verwandte Funktionen

RPJCreatePropertyHandle (Seite 749)	Handle anlegen
RPJGetJobProps (Seite 764)	Druckauftragseigenschaften ermitteln
RPJGetProperty (Seite 766)	Druckauftragseigenschaft ermitteln

Beispiele

Modify print job properties (Seite 783) "RD02.cpp"

Siehe auch

RPJGetJobProps (Seite 764)

RPJGetProperty (Seite 766)

RPJCreatePropertyHandle (Seite 749)

Modify print job properties (Seite 783)

Übersicht über die Funktionen (Seite 723)

2.7.7 Beispiele (RT Professional)

2.7.7.1 Get print job methodname (RT Professional)

Beispiel

```
//{{ODK_EXAMPLE}Get print job methodname (RPT)}
//{{FUNCTION}RPJCreatePropertyHandle (RPT)}
//{{FUNCTION}RPJDeletePropertyHandle (RPT)}
//{{FUNCTION}RPJGetNumJobMethods (RPT)}
//{{FUNCTION}RPJGetJobMethodAt (RPT)}
//{{FUNCTION}(END)}
// =====
// Function: MyPrintJobPropertyInquire( void ) ODK RD RT
// =====
// : Get print job methodname (Enum) !
// =====
void MyRPJGetJobMethodAt(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    CMN_ERROR Error;
    HPROPERTIES hProp = NULL;
    TCHAR jobname[255];
    TCHAR szText[255];
    DWORD d = 0L;
    DWORD i = 0L;
    char buf[500+1];
    DWORD dwBufsize = 500L;
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("-----"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("MyRPJGetJobMethodAt:"));
    ODKTrace(szText);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJAttach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJAttach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJAttach"));
        ODKTrace(szText);
    }
    _tcsncpy_s(jobname, _countof(jobname), _T("ODK_PRINTJOB"), _TRUNCATE); // print job
    memset(&Error, 0, sizeof(CMN_ERROR));
    hProp = RPJCreatePropertyHandle (/*PROJ_PATH*/g_szProjectFile, &Error);
    if(NULL == hProp)

```

```

{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJCreatePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    ret = RPJDetach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
    }
    ODKTrace(szText);
    return;
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJCreatePropertyHandle = 0x
%08lx"), hProp);
    ODKTrace(szText);
}
memset(&Error, 0, sizeof(CMN_ERROR));
ret = RPJGetNumJobMethods(&d, &Error);
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJGetNumJobMethods:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
}
else
{
    for (i = 0; i < d; i++)
    {
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = RPJGetJobMethodAt(i, buf, dwBufsize, &Error);
        if (ret == FALSE)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    %d PrintJobMethod=
%s"), i+1, buf);
        }
        ODKTrace(szText);
    }
}
memset(&Error, 0, sizeof(CMN_ERROR));
ret = RPJDeletePropertyHandle(hProp, &Error);

```

```
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDeletePropertyHandle"));
}
ODKTrace(szText);
memset(&Error, 0, sizeof(CMN_ERROR));
ret = RPJDetach(&Error);
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
}
ODKTrace(szText);
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

- RPJCreatePropertyHandle (Seite 749)
- RPJDeletePropertyHandle (Seite 750)
- RPJGetJobMethodAt (Seite 760)
- RPJGetNumJobMethods (Seite 761)

2.7.7.2 Get print job names (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Get print job names (RPT)}
//{{FUNCTION}RPJCreatePropertyHandle (RPT)}
//{{FUNCTION}RPJDeletePropertyHandle (RPT)}
//{{FUNCTION}RPJGetNumJobs (RPT)}
//{{FUNCTION}RPJGetJobNameAt (RPT)}
//{{FUNCTION}(END)}
// =====
// Function: MyRPJGetJobNameAt( void ) ODK RD RT
// =====
// : Get all print job names (Enum) !
// =====
void MyRPJGetJobNameAt(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    CMN_ERROR Error;
    HPROPERTIES hProp = NULL;
    TCHAR jobname[255];
    TCHAR szText[255];
    DWORD d = 0L;
    DWORD i = 0L;
    char buf[500+1];
    DWORD dwBufsize = 500;
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("-----"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("MyRPJGetJobNameAt:"));
    ODKTrace(szText);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJAttach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJAttach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJAttach"));
        ODKTrace(szText);
    }
    _tcsncpy_s(jobname, _countof(jobname), _T("ODK_PRINTJOB"), _TRUNCATE); // print job
    memset(&Error, 0, sizeof(CMN_ERROR));
    hProp = RPJCreatePropertyHandle (/*PROJ_PATH*/g_szProjectFile, &Error);
    if(NULL == hProp)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJCreatePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),

```

2.7 Funktionen des Protokollsystems (RT Professional)

```

        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    ret = RPJDetach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
    }
    ODKTrace(szText);
    return;
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJCreatePropertyHandle = 0x
%08lx"), hProp);
    ODKTrace(szText);
}
memset(&Error,0,sizeof(CMN_ERROR));
ret = RPJGetNumJobs(/*PROJ_PATH*/g_szProjectFile, &d, &Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJGetNumJobs: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
}
else
{
    for (i = 0; i < d; i++)
    {
        memset(&Error,0,sizeof(CMN_ERROR));
        ret = FALSE;
        ret = RPJGetJobNameAt(/*PROJ_PATH*/g_szProjectFile, i, buf, dwBufsize, &Error);
        if (ret == FALSE)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    %d PrintJobName=
%s"),i+1, buf);
        }
        ODKTrace(szText);
    }
}
memset(&Error,0,sizeof(CMN_ERROR));
ret = FALSE;
ret = RPJDeletePropertyHandle(hProp, &Error);
if (ret == FALSE)

```

```
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDeletePropertyHandle"));
}
ODKTrace(szText);
memset(&Error,0,sizeof(CMN_ERROR));
ret = RPJDetach(&Error);
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
}
ODKTrace(szText);
}
//{{ODK_EXAMPLE} (END)}
```

Siehe auch

- RPJCreatePropertyHandle (Seite 749)
- RPJDeletePropertyHandle (Seite 750)
- RPJGetJobNameAt (Seite 752)
- RPJGetNumJobs (Seite 753)

2.7.7.3 Get print job properties (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Get print job properties (RPT)}
//{{FUNCTION}RPJCreatePropertyHandle (RPT)}
//{{FUNCTION}RPJDeletePropertyHandle (RPT)}
//{{FUNCTION}RPJGetNumJobProperties (RPT)}
//{{FUNCTION}RPJGetJobPropertyAt (RPT)}
//{{FUNCTION}(END)}
// =====
// Function: MyRPJGetJobPropertyAt( void ) ODK RD RT
// =====
// : Inquire print job properties (Enum) !
// =====
void MyRPJGetJobPropertyAt(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    CMN_ERROR Error;
    HPROPERTIES hProp = NULL;
    TCHAR jobname[255];
    TCHAR szText[255];
    DWORD d = 0L;
    DWORD f = 0L;
    DWORD i = 0L;
    char buf[500+1];
    DWORD dwBufsize = 500L;
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("-----"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("MyRPJGetJobPropertyAt:"));
    ODKTrace(szText);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJAttach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJAttach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJAttach"));
        ODKTrace(szText);
    }
    _tcsncpy_s(jobname, _countof(jobname), _T("ODK_PRINTJOB"), _TRUNCATE); // print job
    memset(&Error, 0, sizeof(CMN_ERROR));
    hProp = RPJCreatePropertyHandle (/*PROJ_PATH*/g_szProjectFile, &Error);
    if(NULL == hProp)
    {

```

```

        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJCreatePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
        ret = RPJDetach(&Error);
        if (FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
        }
        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJCreatePropertyHandle = 0x
%08lx"), hProp);
        ODKTrace(szText);
    }
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = RPJGetNumJobProperties(&d, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJGetNumJobProperties: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        for (i = 0; i < d; i++)
        {
            memset(&Error,0,sizeof(CMN_ERROR));
            ret = RPJGetJobPropertyAt(i, buf, dwBufsize, &f, &Error);
            if (FALSE == ret)
            {
                _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJGetJobPropertyAt: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
            }
            else
            {
                _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  %d PrintJobProperty=
%s"),i+1, buf);
            }
            ODKTrace(szText);
        }
    }
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = RPJDeletePropertyHandle(hProp, &Error);
    if (FALSE == ret)

```

```
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDeletePropertyHandle"));
}
ODKTrace(szText);
memset(&Error,0,sizeof(CMN_ERROR));
ret = RPJDetach(&Error);
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
}
ODKTrace(szText);
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

- RPJCreatePropertyHandle (Seite 749)
- RPJDeletePropertyHandle (Seite 750)
- RPJGetJobPropertyAt (Seite 762)
- RPJGetNumJobProperties (Seite 765)

2.7.7.4 Modify print job properties (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Modify print job properties (RPT)}
//{{FUNCTION}RPJCreatePropertyHandle (RPT)}
//{{FUNCTION}RPJDeletePropertyHandle (RPT)}
//{{FUNCTION}RPJGetJobProps (RPT)}
//{{FUNCTION}RPJGetProperty (RPT)}
//{{FUNCTION}RPJSetJobProps (RPT)}
//{{FUNCTION}RPJSetProperty (RPT)}
//{{FUNCTION}(END)}
// =====
// Function: MyModifyPrintJob( void ) ODK RD RT
// =====
// : Modify print job properties (change starttime) !
// =====
void MyModifyPrintJob(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    CMN_ERROR Error;
    TCHAR szText[255];
    HPROPERTIES hProp = NULL;
    SYSTEMTIME st;
    LPVOID ptr;
    DWORD typ;
    TCHAR jobname[200];
    TCHAR propname[200];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("-----"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("MyModifyPrintJob:"));
    ODKTrace(szText);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJAttach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJAttach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJAttach"));
        ODKTrace(szText);
    }
    // read properties
    memset(&Error, 0, sizeof(CMN_ERROR));
    hProp = RPJCreatePropertyHandle (/*PROJ_PATH*/g_szProjectFile, &Error);
    if (NULL == hProp)
    {

```

```

    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJCreatePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
    Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    ret = RPJDetach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
    }
    ODKTrace(szText);
    return;
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJCreatePropertyHandle = 0x
%08lx"), hProp);
    ODKTrace(szText);
}
_tcsncpy_s(jobname, _countof(jobname), _T("ODK_PRINTJOB"), _TRUNCATE);
memset(&Error, 0, sizeof(CMN_ERROR));
ret = RPJGetJobProps (hProp, jobname, &Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJGetJobProps: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
    Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    ret = RPJDeletePropertyHandle (hProp, &Error);
    if(ret == FALSE)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDeletePropertyHandle"));
    }
    ODKTrace(szText);
    ret = RPJDetach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
    }
}

```

```

        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    RPJGetJobProps"));
        ODKTrace(szText);
    }
    typ = VT_DATE;
    _tcsncpy_s(jobname, _countof(jobname), _T("ODK_PRINTJOB"), _TRUNCATE);
    _tcsncpy_s(propname, _countof(propname), _T("STARTTIME"), _TRUNCATE);
    memset(&st, 0, sizeof(SYSTEMTIME));
    ptr = (LPVOID)&st;
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJGetProperty (hProp, propname, ptr, (VARTYPE)typ, 16, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJGetProperty: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    RPJGetProperty"));
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    Jobname=%s Propname=
%s"), jobname, propname);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    %02d.%02d.%04d %02d:
%02d:%02d"),
            (WORD)st.wDay,
            (WORD)st.wMonth,
            (WORD)st.wYear,
            (WORD)st.wHour,
            (WORD)st.wMinute,
            (WORD)st.wSecond);
    }
    ODKTrace(szText);
    // write properties
    st.wHour = 11;
    st.wMinute = 12;
    st.wSecond = 13;
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJSetProperty (hProp, propname, ptr, (VARTYPE) typ, 16, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJSetProperty: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    RPJSetProperty"));
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    Jobname=%s ;Propname=
%s"), jobname, propname);
    }
}

```

```

        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    %02d.%02d.%04d %02d:
%02d:%02d"),
            (WORD)st.wDay,
            (WORD)st.wMonth,
            (WORD)st.wYear,
            (WORD)st.wHour,
            (WORD)st.wMinute,
            (WORD)st.wSecond);
    }
    ODKTrace(szText);
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = RPJSetJobProps (hProp, jobname, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJSetJobProps: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("    RPJSetJobProps"));
    }
    ODKTrace(szText);
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = RPJDeletePropertyHandle (hProp, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDeletePropertyHandle"));
    }
    ODKTrace(szText);
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = RPJDetach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE,
_T("RPJDeletePropertyHandle"));    }
    ODKTrace(szText);
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

RPJCreatePropertyHandle (Seite 749)

RPJDeletePropertyHandle (Seite 750)

RPJGetJobProps (Seite 764)

RPJGetProperty (Seite 766)

RPJSetProperty (Seite 771)

2.7.7.5 Show print job preview (RT Professional)

Beispiel

```
// =====
// =====
// : Modul with examples of Report Designer
// *****
// Copyright (C) 1995-99 SIEMENS AG, A&D PT1 D2 All rights reserved
// *****
#include "stdafx.h" // if MFC classes
#include "RD02.h"
#include "DM01.h"

//{{ODK_EXAMPLE}Show print job preview (RPT)}
//{{FUNCTION}RPJCreatePropertyHandle (RPT)}
//{{FUNCTION}RPJDeletePropertyHandle (RPT)}
//{{FUNCTION}RPJGetJobProps (RPT)}
//{{FUNCTION}RPJCallJobMethod (RPT)}
//{{FUNCTION}(END)}
// =====
// Function: MyShowPrintJobPreview( void ) ODK RD RT
// =====
// : Show print job preview
// =====
void MyShowPrintJobPreview(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    TCHAR szText[255];
    CMN_ERROR Error;
    HPROPERTIES hProp = NULL;
    TCHAR jobname[200];
    TCHAR methode[200];
    _tcsncpy_s(jobname, _countof(jobname), _T("ODK_PRINTJOB"), _TRUNCATE); // print job
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("-----"));
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("MyShowPrintJobPreview:"));
    ODKTrace(szText);
    MyDMMenumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJAttach(&Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJAttach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJAttach"));
        ODKTrace(szText);
    }
}

```

```

}
memset(&Error, 0, sizeof(CMN_ERROR));
hProp = RPJCreatePropertyHandle (/*PROJ_PATH*/g_szProjectFile, &Error);
if(NULL == hProp)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJCreatePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    ret = RPJDetach(&Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
    }
    ODKTrace(szText);
    return;
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJCreatePropertyHandle = 0x
%08lx"), hProp);
    ODKTrace(szText);
}
memset(&Error, 0, sizeof(CMN_ERROR));
ret = RPJGetJobProps (hProp, jobname, &Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJGetJobProps: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    ret = RPJDeletePropertyHandle (hProp, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDeletePropertyHandle"));
    }
    ODKTrace(szText);
    ret = RPJDetach(&Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
}
}

```

```

        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
        }
        ODKTrace(szText);
        return;
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  RPJGetJobProps"));
        ODKTrace(szText);
    }
    _tcsncpy_s(methode, _countof(methode), _T("PREVIEW"), _TRUNCATE); // preview
    // _tcsncpy_s(methode, _countof(methode), _T("PRINTJOB"), _TRUNCATE); // print
    // _tcsncpy_s(methode, _countof(methode), _T("SETSELECTIONALLPAGES"), _TRUNCATE);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJCallJobMethod (hProp, methode, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJGetJobProps: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  RPJCallJobMethod"));
    }
    ODKTrace(szText);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJDeletePropertyHandle (hProp, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
RPJDeletePropertyHandle: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDeletePropertyHandle"));
    }
    ODKTrace(szText);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = RPJDetach(&Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in RPJDetach: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("RPJDetach"));
    }
    ODKTrace(szText);
}
//{{ODK_EXAMPLE} (END) }

```

Siehe auch

RPJCreatePropertyHandle (Seite 749)

RPJDeletePropertyHandle (Seite 750)

RPJCallJobMethod (Seite 759)

RPJGetJobProps (Seite 764)

2.8 Funktionen für Variablen und Archive (RT Professional)**2.8.1 Grundlagen (RT Professional)****2.8.1.1 Übersicht über die Funktionen (RT Professional)****Übersicht**

TLG_ENUM_ARCHIV_CALLBACK (Seite 867)	Archive auflisten (Callback)
TLG_ENUM_PROJECT_NAME_CALLBACK (Seite 857)	Projekte auflisten (Callback)
AUTOHOTSPOT	Einträge in Kurvenvorlagen auflisten (Callback)
TLG_ENUM_TIME_NAME_CALLBACK (Seite 904)	Zeitobjekte auflisten (Callback)
TLG_ENUM_VARIABLE_NAME_CALLBACK (Seite 860)	Variablen auflisten (Callback)
TLG_ENUMBACKUP_ENTRIES (Seite 911)	Backups auflisten (Callback)
TLG_ENUMTABLES (Seite 874)	Archive auflisten (Callback)
TLG_ENUMTIMES_CALLBACK (Seite 907)	Zeitobjekte auflisten (Callback)
TLG_ENUMVARIABLES (Seite 863)	Variablen auflisten (Callback)
TLG_GETARCHIVDATA_CALLBACK (Seite 879)	Daten eines Archives auslesen (Callback)
TLGChangeLanguage (Seite 849)	Sprachumschaltung
TLGCloseProject (Seite 853)	Projekt schließen
TLGCloseWindow (Seite 894)	Druckauftrag/Skriptdiagnose schließen
TLGConnect (Seite 850)	Verbindung zum Tag Logging RT aufbauen
TLGCSCConnectEx (Seite 846)	Verbindung zum Tag Logging CS aufbauen
TLGCSCConnect (Seite 845)	Verbindung zum Tag Logging CS aufbauen
TLGDisconnect (Seite 851)	Verbindung zum Tag Logging RT abbauen
TLGDrawCurvesInDC (Seite 895)	Kurven darstellen
TLGEnumArchives (Seite 866)	Archive auflisten
TLGEnumArchivsEx (Seite 870)	Archive auflisten
TLGEnumArchivs (Seite 868)	Archive auflisten
TLGEnumArchivsSel (Seite 872)	Archive auflisten
TLGEnumBackupEntries (Seite 909)	Backups auflisten

2.8 Funktionen für Variablen und Archive (RT Professional)

TLGEnumProject (Seite 856)	Projekte auflisten
TLGEnumTime (Seite 902)	Zeitobjekte auflisten
TLGEnumTimes (Seite 905)	Zeitobjekte auflisten
TLGEnumVariablesEx (Seite 861)	Variablen auflisten
TLGEnumVariables (Seite 858)	Variablen auflisten
TLGExport (Seite 912)	Archiv auslagern
TLGFreeMemory (Seite 876)	Speicher freigeben
TLGGetArchivDataEx (Seite 881)	Daten eines Archives auslesen
TLGGetArchivData (Seite 877)	Daten eines Archives auslesen
TLGGetBackupSize (Seite 914)	Größe eines exportierten Datensatzes bestimmen
TLGGetClosestTimeEx (Seite 885)	Archivierungszeitpunkt ermitteln
TLGGetClosestTime (Seite 883)	Archivierungszeitpunkt ermitteln
TLGInsertArchivData (Seite 887)	Daten in ein Archiv einfügen
TLGInsertTemplateltem (Seite 897)	Einträge in Kurvenfenster-Vorlagen schreiben, Einträge in Tabellenfenster-Vorlagen schreiben
TLGLockArchiv (Seite 890)	Archiv freigeben, Archiv sperren
TLGLockVariable (Seite 891)	Variable freigeben, Variable sperren
TLGOpenProject (Seite 854)	Projekt öffnen
TLGPressToolbarButton (Seite 898)	Schaltflächen der Symbolleiste auslösen
TLGReadArchiv (Seite 893)	Archivparameter lesen
TLGReadTime (Seite 908)	Zeitobjekt-Parameter ermitteln
TLGReadVariable (Seite 864)	Variablen-Parameter ermitteln
TLGSetRulerWindowVisible (Seite 900)	Leselinienfenster mit anzeigen
TLGShowWindow (Seite 901)	Druckauftrag/Skriptdiagnose anzeigen

2.8.1.2 Übersicht über die Strukturen (RT Professional)

Übersicht

TLG_ARCHIV_STR (Seite 805)	Archivparameter
TLG_ARCHIVDATARAW (Seite 808)	Archivdaten
TLG_BACKUP_TABLE_INFO (Seite 811)	Informationen zur Auslagerungstabelle
TLG_CURVESCALEX (Seite 813)	Skalierung der x-Achse
TLG_CURVESCALEY (Seite 818)	Skalierung der y-Achse
TLG_GETARCHIVDATA (Seite 822)	Archivdaten von Prozesswertarchiven, Archivdaten von Verdichtungsarchiven
TLG_IO_BACKUP_SELECT (Seite 823)	Selektionskriterien für Backup-Funktionen
TLG_PROT_CURVE_INFOS (Seite 824)	Protokollstruktur von Kurven
TLG_TABLE_INFO (Seite 827)	Protokollstruktur von Tabellen
TLG_SCAL_STR (Seite 826)	Variablengrenzwerte
TLG_TABLESCALE (Seite 828)	Tabellenspalte (Eigenschaften)
TLG_TEMPLATEITEM_INFO (Seite 831)	Kurve (Eigenschaften)Tabellenspalte (Eigenschaften)

TLG_TIME_STR (Seite 833)	Zeitobjekt
TLG_TIMEDATA (Seite 834)	Zeitobjekt
TLG_TPLITEM_CURVE (Seite 835)	Kurve (Eigenschaften)
TLG_TPLITEM_INFO (Seite 836)	Kurve (Eigenschaften)Tabellenspalte (Eigenschaften)
TLG_TPLITEM_TABLE (Seite 837)	Tabellenspalte (Eigenschaften)
TLG_VAR_STR (Seite 838)	Variable (Eigenschaften)
TLG_VARIABLE_INFO (Seite 844)	Variable

2.8.1.3 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

TLG_API_ERROR	0x10000000	nicht spezifizierter Fehler
TLG_API_NO_TIME_EXIST	0x10000001	es existiert kein Zeitobjekt
TLG_API_TIME_NAME_EXIST	0x10000002	ein Zeitobjekt mit diesem Namen existiert bereits
TLG_API_TIME_NAME_NOT_EXIST	0x10000003	kein Zeitobjekt mit diesem Namen gefunden
TLG_API_NO_ARCHIV_EXIST	0x10000004	kein Archiv verfügbar
TLG_API_ARCHIV_NAME_EXIST	0x10000005	es gibt bereits ein Archiv mit diesem Namen
TLG_API_ARCHIV_NAME_NOT_EXIST	0x10000006	kein Archiv mit diesem Namen gefunden
TLG_API_INVALID_ARCHIV_TYP	0x10000007	ungültiger Archivtyp
TLG_API_NO_VARIABLE_EXIST	0x10000008	keine Variablen vorhanden
TLG_API_VARIABLE_NAME_EXIST	0x10000009	eine Variable mit diesem Namen existiert bereits
TLG_API_VARIABLE_NAME_NOT_EXIST	0x1000000A	keine Variable mit diesem Namen gefunden
TLG_API_VARTYPE_MISMATCH	0x1000000B	falscher Variablentyp
TLG_API_SRC_NO_VARIABLE_EXIST	0x1000000C	SRC: keine Variablen vorhanden
TLG_API_SRC_VARIABLE_NAME_NOT_EXIST	0x1000000D	SRC: keine Variable mit diesem Namen gefunden
TLG_API_SRC_ARCHIV_NOT_EXIST	0x1000000E	SRC: Archiv nicht gefunden
TLG_API_NO_FIELD_EXIST	0x1000000F	es sind keine Felder verfügbar
TLG_API_FIELD_NAME_EXIST	0x10000010	ein Feld mit diesem Namen ist bereits vorhanden
TLG_API_FIELD_NAME_NOT_EXIST	0x10000011	kein Feld mit diesem Namen gefunden
TLG_API_NO_PROJECT_EXIST	0x10000012	kein Projekt vorhanden
TLG_API_PROJECT_NAME_NOT_EXIST	0x10000013	Projektname nicht gefunden
TLG_API_PROJECT_IS_ACTIVE	0x20000014	das Projekt ist aktiviert

2.8 Funktionen für Variablen und Archive (RT Professional)

TLG_API_NO_CONNECTION	0x10000015	keine Verbindung zu einem aktiven Projekt hergestellt
TLG_API_RENAME_NAME_EXIST	0x10000016	der Name in den umbenannt werden soll existiert bereits
TLG_API_NAME_WRONG_CHAR	0x10000017	falsches Zeichen im Namen
TLG_API_NAME_TOLONG	0x10000018	der Name ist zu lang
TLG_API_ERR_ARCHIVSTYLE	0x10000019	Fehler im Archivdesign
TLG_API_NO_TYP_CHANGE_ALLOWED	0x1000001A	Typänderung ist nicht erlaubt
TLG_API_ERR_SUPPLY	0x1000001B	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	0x1000001C	ungültige/falsche Parameterversorgung
TLG_API_NOT_SUPPORTED	0x1000001D	diese Funktion wird nicht unterstützt
TLG_API_NO_INTERFACE	0x1000001E	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NODEFAULTSERVER	0x10000021	Kein Standardserver projektiert
TLG_API_NOLOCALSERVER	0x10000022	Kein lokaler Server verfügbar
TLG_API_NOSERVER	0x10000023	Kein Standardserver projektiert und kein lokaler Server verfügbar
TLG_API_NOMC	0x10000024	Kein Multiclient Projekt (wird hier nicht verwendet)
TLG_API_NOMCDEFAULTSERVER	0x10000025	Kein Multiclient Projekt aber "@default:" als Serverpräfix angegeben (wird hier nicht verwendet)
TLG_API_UNKNOWN_ERROR	0x10000030	unbekannter Callback Fehler (ab WinCC V6.2)
TLG_API_ASYNC_TIMEOUT	0x10000031	asynchrone Funktion mit Timeout abgebrochen (ab WinCC V6.2)
TLG_API_EXPORT_NO_WRITE	0x10000032	Export-Datei wurde nicht geschrieben (ab WinCC V6.2)
TLG_API_CORRUPT_DATA	0x10000033	Daten unbrauchbar / defekt (ab WinCC V6.2)
IDS_API_ERROR_CREATING_CLIENT_WND	30001	Client Fenster konnte nicht erzeugt werden
IDS_API_ERROR_WINDOW_NOT_FOUND	30002	Button (Fenster) konnte nicht gefunden werden

2.8.1.4 Konstanten (RT Professional)

Runtime-Fenster

TLG_CURVE_WINDOW	0x00000001	Kurvenfenster
TLG_TABLE_WINDOW	0x00000002	Tabellenfenster
TLG_CTRL_WINDOW	0x00000004	TLGRT Kontrollfenster

Auslagerungsfunktionen

TLG_BACKUP_EXPORT	0x00000001	Auslagern ohne Löschen der Quelldaten.
TLG_BACKUP_EVACUATE	0x00000002	Auslagern mit Löschen der Quelldaten.
TLG_BACKUP_DELETE	0x00000004	Löschen der Datensätze ohne Auslagerung.
TLG_BACKUP_RAW	0x00000008	Arch. auf Änderung nicht besonders behandeln.

Export/Import-Formate

TLG_BAKFMT_CSV	0x00000001	CSV-Format (Comma Separated Value)
----------------	------------	------------------------------------

Einlagerungsfunktionen

TLG_RESTORE_OVERWRITE	0x00000001	Import mit Überschreiben.
TLG_RESTORE_MERGE	0x00000002	Import mit Überprüfung, Fehler über Callback.
TLG_RESTORE_BREAK	0x00000004	Import - Abbruch bei Fehler

Archivtypen

TLG_ARCTYP_USER	0x00000001	Anwenderarchiv (ab WinCC V5.0 nicht mehr unterstützt)
TLG_ARCTYP_PROCESS	0x00000002	Prozesswertarchiv
TLG_ARCTYP_COMPRESS	0x00000004	Verdichtungsarchiv

Archivarten

TLG_ARCTYP_ALL	0x00000000	Umlauf oder Folgearchiv
TLG_ARCTYP_CIRCULAR	0x00010000	Umlaufarchiv
TLG_ARCTYP_FOLLOW	0x00020000	Folgearchiv

Archivflags

TLG_API_FLG_FAST_INSERT	0x00000001	Ab WinCC V6.0 SP1: Schnelles Einfügen von Daten in SQL Server mit TLGInsertArchivData
-------------------------	------------	---

Fenstervorlagen

TLG_TEMPLATE_ALL	0x00000000	Alle Fenstervorlagen
TLG_TEMPLATE_CURVE	0x00000001	Kurvenfenster-Vorlage
TLG_TEMPLATE_TABLE	0x00000002	Tabellenfenster-Vorlage
TLG_TYP_TEMPLATE_CURVE	0x00800001	Kurvenfenster-Vorlage
TLG_TYP_TEMPLATE_TABLE	0x00800002	Tabellenfenstervorlage

Vorlagen

TLG_TEMPLATEITEM_ALL	0x00000000	Alle Template Items
TLG_TEMPLATEITEM_CURVE	0x00000001	Kurven-Template Items
TLG_TEMPLATEITEM_TABLE	0x00000002	Tabellen-Template Items

Modify Templates Items

TLG_TI_ACTION_COLOR	0x00000001	Farbe ändern
TLG_TI_ACTION_SCALE_X	0x00000002	X-Skalierung ändern
TLG_TI_ACTION_SCALE_Y	0x00000004	Y-Skalierung ändern
TLG_TI_ACTION_VISIBLE	0x00000005	sichtbar/unsichtbar schalten
TLG_TI_ACTION_RANGE_X	0x00000006	Anzeigebereich X-Achse
TLG_TI_ACTION_RANGE_Y	0x00000007	Anzeigebereich Y-Achse

Grenzwerte

TLG_MAX_SQL_SELECT	512	Maximale Länge eines SQL-Befehles
TLG_MAX_TEMPLATE_NAME	32	Maximale Länge des Namens einer Kurven- bzw. Tabellenfenstervorlage
TLG_MAX_TEMPLATEITEM_NAME	32	Maximale Länge des Namens einer Kurven- bzw. Tabellenvorlage
TLG_MAX_FUNCTION_NAME	32	Maximale Länge des Namens einer Aktion bzw. einer Funktion.
TLG_MAX_DLL_NAME	32	Maximale Länge des Namens einer DLL.
TLG_MAX_STD_TEXT_NAME	32	TLGSetTemplateData
TLG_SETDATA_DEFAULT	0x00000000	y=f(x)
TLG_SETDATA_RESET	0x00000001	Der Kurvenpuffer wird gelöscht.
TLG_SETDATA_TIME_RANGE_X	0x00000002	Die x-Werte sind als Zeitbereich zu interpretieren: y=f(t).
TLG_SETDATA_TIME_RANGE_Y	0x00000004	Die y-Werte sind als Zeitbereich zu interpretieren: x=f(t).

Kurvenformen

TLG_CURVEFORM_STEP	0x00000011	Darstellung der Variablenwerte als Treppenkurve.
TLG_CURVEFORM_POINTS	0x00000012	Darstellung der Variablenwerte als Einzelpunkte.
TLG_CURVEFORM_DIRECT	0x00000014	Variablenwerte linear interpolieren.
TLG_CURVEFORM_DIRFILL	0x00000021	
TLG_CURVEFORM_STEPFILL	0x00000022	

Datenbereichstypen

TLG_DATATYP_TIMERANGE	0x00000001	
TLG_DATATYP_USERARCHIV	0x00000002	
TLG_DATATYP_BLOCKDATA	0x00000004	

Zeitbereiche

TLG_TR_MINUTE	1	
TLG_TR_HOUR	2	
TLG_TR_DAY	3	
TLG_TR_WEEK	4	
TLG_TR_MONTH	5	
TLG_TR_YEAR	6	

Achsen

TLG_AXIS_X_TOP	0x00000001	
TLG_AXIS_X_MIDDLE	0x00000002	
TLG_AXIS_X_BOTTOM	0x00000004	
TLG_AXIS_Y_LEFT	0x00000001	
TLG_AXIS_Y_MIDDLE	0x00000002	
TLG_AXIS_Y_RIGHT	0x00000004	

Protokollanbindung

TLG_PROTFLG_TIME_RANGE	0x00000001	
TLG_PROTFLG_DATA_RANGE	0x00000002	
TLG_PROTFLG_DATA	0x00000001	
TLG_PROTFLG_TITLE	0x00000002	
TLG_PROTFLG_TIMEFIELD	0x00000004	
TLG_PROTFLG_DATACOUNT	0x00000008	
TLG_PROTFLG_DATEFIELD	0x00000010	

TAG_PROT_CURVE_FORM_STEP	0x00000001	
TAG_PROT_CURVE_FORM_POINTS	0x00000002	
TAG_PROT_CURVE_FORM_DIRECT	0x00000003	
TAG_PROT_CURVE_FORM_AREA	0x00000004	
TLG_PROTFLG_FIRST_COL	0x00000000	
TLG_PROTFLG_FIRST_COL_TIME	0x00000001	
TLG_PROTFLG_FIRST_COL_DATA	0x00000002	
TLG_PROTFLG_FIRST_COL_DATE	0x00000004	

Archivdaten lesen

TLG_DATA_FROM_POS	0x00000000	
TLG_DATA_FROM_BEGIN	0x00000001	
TLG_DATA_FROM_END	0x00000002	
TLG_DATA_FROM_ACTUAL	0x00000004	
TLG_DATA_FROM_LEFT_ABS	0x00000008	
TLG_DATA_FROM_RIGHT_ABS	0x00000010	

Schaltflächen der Runtime Symbolleiste

Schaltflächen für Kurven und Tabellen

TLG_BASIC_BTN_HELP	0x00000001	Kennung der Schaltfläche zum Aufruf der Online-Hilfe.
TLG_BASIC_BTN_DLG	0x00000002	Kennung der Schaltfläche zum Öffnen des Parametrierungsdialog für Kurven- und Tabellenfenstervorlagen.
TLG_BASIC_BTN_FIRST	0x00000004	Kennung der Schaltfläche zur Anzeige der ersten Datensätze eines Archives.
TLG_BASIC_BTN_PREV	0x00000008	Kennung der Schaltfläche zum Rückwärtsblättern im Archiv.
TLG_BASIC_BTN_NEXT	0x00000010	Kennung der Schaltfläche zum Vorwärtsblättern im Archiv.
TLG_BASIC_BTN_LAST	0x00000020	Kennung der Schaltfläche zur Anzeige der letzten Datensätze eines Archives.
TLG_BASIC_BTN_STARTSTOP	0x00000040	Kennung der Schaltfläche zum Anhalten bzw. Fortsetzen der aktualisierten Darstellung einer Kurve oder Tabelle.
TLG_BASIC_BTN_PREV_ITEM	0x00000800	Vorherige Kurve in Vordergrund bzw. Tabellenspalte zuerst.
TLG_BASIC_BTN_NEXT_ITEM	0x00001000	Nächste Kurve in Vordergrund bzw. Tabellenspalte zuerst.
TLG_BASIC_BTN_ARC_VAR_SELECT	0x00000100	Öffnet den Archiv-variablen Auswahldialog.

TLG_BASIC_BTN_ITEM_SELECT	0x00000200	Öffnet den Auswahldialog für Kurven bzw. Tabellenspalten.
TLG_BASIC_BTN_TIME_SELECT	0x00000400	Öffnet den Auswahldialog für den Darstellungszeitbereich.

Schaltflächen für Kurven

TLG_CURVE_BTN_ZOOMIN	0x01000000	Kennung der Schaltfläche zum Aktivieren der Zoomfunktion zur Ausschnittsvergrößerung. Die Größe des Zoomausschnittes kann nur über die Maus festgelegt werden!
TLG_CURVE_BTN_ZOOMOUT	0x02000000	Kennung der Schaltfläche zum Deaktivieren der Zoomfunktion. Dabei wird nicht in die Normaldarstellung umgeschaltet.
TLG_CURVE_BTN_LINEAL	0x04000000	Kennung der Schaltfläche zum Aktivieren bzw. Deaktivieren des Lineals zur Ermittlung der Koordinaten eines Messpunktes.
TLG_CURVE_BTN_1_TO_1	0x08000000	Kennung der Schaltfläche zum Umschalten in die Normaldarstellung einer Kurve.

Schaltflächen für Tabellen

TLG_TABLE_BTN_EDIT	0x00040000	Kennung der Schaltfläche zum Aktivieren bzw. Deaktivieren der Editierfunktion.
--------------------	------------	--

Felddefinitionen für Strukturen

PDE_DB_PRJ_PROJNAME_MAX LENGHT TLG_DB_PRJ_PROJNAME_MAX LENGHT	128	
PDE_DB_PRJ_COMMENT_MAX LENGHT TLG_DB_PRJ_COMMENT_MAX LENGHT	200	
PDE_DB_PRJ_USERDATE_MAX LENGHT TLG_DB_PRJ_USERDATE_MAX LENGHT	12	
PDE_DB_PRJ_USERNAME_MAX LENGHT TLG_DB_PRJ_USERNAME_MAX LENGHT	30	
PDE_DB_ARC_NAME_MAX LENGHT TLG_DB_ARC_NAME_MAX LENGHT	32	
PDE_DB_ARC_NAME_MAX LENGHT TLG_DB_ARC_NAME_MAX LENGHT	32	
TLG_DB_ARC_USER_NAME_MAX LENGHT	8	
PDE_DB_ARC_SERVERNAME_MAX LENGHT TLG_DB_ARC_SERVERNAME_MAX LENGHT	50	
PDE_DB_ARC_COMMENT_MAX LENGHT TLG_DB_ARC_COMMENT_MAX LENGHT	100	
PDE_DB_ARC_ACTIONNAME_MAX LENGHT TLG_DB_ARC_ACTIONNAME_MAX LENGHT	30	
PDE_DB_VAR_NAME_MAX LENGHT TLG_DB_VAR_NAME_MAX LENGHT	64	

TLG_DB6_VAR_NAME_MAX_LENGTH	128	
TLG_DB_MCP_VAR_MAME_MAX_LENGTH	128	
PDE_DB_VAR_COMMENT_MAX_LENGTH	100	
TLG_DB_VAR_COMMENT_MAX_LENGTH		
PDE_DB_VAR_WRITEBACK_MAX_LENGTH	64	
TLG_DB_VAR_WRITEBACK_MAX_LENGTH		
TLG_DB6_VAR_WRITEBACK_MAX_LENGTH	128	
PDE_DB_VAR_CYCLENAME_MAX_LENGTH	30	
TLG_DB_VAR_CYCLENAME_MAX_LENGTH		
PDE_DB_VAR_SIGNALTEX_MAX_LENGTH	30	
TLG_DB_VAR_SIGNALTEX_MAX_LENGTH		
PDE_DB_VAR_DLLNAME_MAX_LENGTH	30	
TLG_DB_VAR_DLLNAME_MAX_LENGTH		
PDE_DB_VAR_FUNCNAME_MAX_LENGTH	30	
TLG_DB_VAR_FUNCNAME_MAX_LENGTH		
PDE_DB_VAR_UNITDIR_MAX_LENGTH	30	
TLG_DB_VAR_UNITDIR_MAX_LENGTH		
PDE_DB_VAR_UNITSTR_MAX_LENGTH	64	
TLG_DB_VAR_UNITSTR_MAX_LENGTH		
TLG_DB6_VAR_UNITSTR_MAX_LENGTH	128	
PDE_DB_FUNCNAME_MAX_LENGTH	30	
TLG_DB_FUNCNAME_MAX_LENGTH		
PDE_DB_DLLNAME_MAX_LENGTH	30	
TLG_DB_DLLNAME_MAX_LENGTH		
PDE_DB_SCALE_MAX_LENGTH	30	
TLG_DB_SCALE_MAX_LENGTH		
PDE_DB_TIMENAME_MAX_LENGTH	30	
TLG_DB_TIMENAME_MAX_LENGTH		
PDE_DB_FIELDNAME_MAX_LENGTH	64	
TLG_DB_FIELDNAME_MAX_LENGTH		
PDE_DB_STRUCTNAME_MAX_LENGTH	30	
TLG_DB_STRUCTNAME_MAX_LENGTH		

Felddefinitionen VARIANT-Record zu Archivvariable

für TLGCSCreateTagMulti, TLGCSReadTagMulti und TLGCSModifyTagMulti

siehe Include-Datei pde_typ.h

Diese Funktionen stehen erst ab WinCC V6.2 zur Verfügung.

Definition in ENUM_TLG_TAG _FIELDINDEX	Feldindex	zugehörige Typ- Definition	Typ	Bemerkung / Ein- schränkung
ENUM_TLG_TAG _ENUMTYPE	0	VT_TLG_TAG_EN UMTYPE	VT_I4	intern festgelegt (Wert >= 8), keine Änderung möglich
ENUM_TLG_TAG _VERSION	1	VT_TLG_TAG_VE RSION	VT_I4	intern festgelegt (Wert >= 1), keine Änderung möglich
ENUM_TLG_TAG _CREATORID	2	VT_TLG_TAG_CR EATORID	VT_I4	
ENUM_TLG_TAG _NAME	3	VT_TLG_TAG_NA ME	VT_BSTR	erforderlich bei Create
ENUM_TLG_TAG _ARCHIVNAME	4	VT_TLG_TAG_AR CHIVNAME	VT_BSTR	erforderlich bei Create
ENUM_TLG_TAG _TYPE	5	VT_TLG_TAG_TY PE	VT_I4	erforderlich bei Create
ENUM_TLG_TAG _WRITEBACKNA- ME	6	VT_TLG_TAG_W RITEBACKNAME	VT_BSTR	
ENUM_TLG_TAG _COMMENT	7	VT_TLG_TAG_CO MMENT	VT_BSTR	
ENUM_TLG_TAG _ARCTYPE	8	VT_TLG_TAG_AR CTYPE	VT_I4	
ENUM_TLG_TAG _SUPPLYTYPE	9	VT_TLG_TAG_SU PPLYTYPE	VT_I4	
ENUM_TLG_TAG _LOCKED	10	VT_TLG_TAG_LO CKED	VT_I4	
ENUM_TLG_TAG _SCANTIME	11	VT_TLG_TAG_SC ANTIME	VT_BSTR	erforderlich bei Create
NUM_TLG_TAG _ARCTIME	12	VT_TLG_TAG_AR CTIME	VT_BSTR	erforderlich bei Create
ENUM_TLG_TAG _ARCFACTOR	13	VT_TLG_TAG_AR CFACTOR	VT_I4	erforderlich bei Create
ENUM_TLG_TAG _VARCOUNT- FORWARD	14	VT_TLG_TAG_VA RCOUNTFOR- WARD	VT_I4	
ENUM_TLG_TAG _VARCOUNT- BACKWARD	15	VT_TLG_TAG_VA RCOUNTBACK- WARD	VT_I4	
ENUM_TLG_TAG _SAVEBYFAULT	16	VT_TLG_TAG_SA VEBYFAULT	VT_I4	
ENUM_TLG_TAG _ARCTRIGGER	17	VT_TLG_TAG_AR CTRIGGER	VT_I4	erforderlich bei Create
ENUM_TLG_TAG _STATETEX- THIGH	18	VT_TLG_TAG_ST ATETEXHIGH	VT_BSTR	

2.8 Funktionen für Variablen und Archive (RT Professional)

Definition in ENUM_TLG_TAG_FIELDINDEX	Feldindex	zugehörige Typ-Definition	Typ	Bemerkung / Einschränkung
ENUM_TLG_TAG_STATETEXTLOW	19	VT_TLG_TAG_STATETEXTLOW	VT_BSTR	
ENUM_TLG_TAG_VARPRO	20	VT_TLG_TAG_VARPRO	VT_I4	
ENUM_TLG_TAG_FUNCVALPRO	21	VT_TLG_TAG_FUNCVALPRO	VT_BSTR	
ENUM_TLG_TAG_DLLVARPRO	22	VT_TLG_TAG_DLLVARPRO	VT_BSTR	
ENUM_TLG_TAG_FUNCSTARTEVENT	23	VT_TLG_TAG_FUNCSTARTEVENT	VT_BSTR	
ENUM_TLG_TAG_DLLSTARTEVENT	24	VT_TLG_TAG_DLLSTARTEVENT	VT_BSTR	
ENUM_TLG_TAG_FUNCSTOPEVENT	25	VT_TLG_TAG_FUNCSTOPEVENT	VT_BSTR	
ENUM_TLG_TAG_DLLSTOPEVENT	26	VT_TLG_TAG_DLLSTOPEVENT	VT_BSTR	
ENUM_TLG_TAG_FUNCDYNAMIC	27	VT_TLG_TAG_FUNCDYNAMIC	VT_BSTR	
ENUM_TLG_TAG_DLLDYNAMIC	28	VT_TLG_TAG_DLLDYNAMIC	VT_BSTR	
ENUM_TLG_TAG_UNITDIRECT	29	VT_TLG_TAG_UNITDIRECT	VT_BSTR	
ENUM_TLG_TAG_NAMEUNITSTRUCT	30	VT_TLG_TAG_NAMEUNITSTRUCT	VT_BSTR	
ENUM_TLG_TAG_UNITINDEX	31	VT_TLG_TAG_UNITINDEX	VT_I4	
ENUM_TLG_TAG_VARUPPER	32	VT_TLG_TAG_VARUPPER	VT_R8	
ENUM_TLG_TAG_VARLOWER	33	VT_TLG_TAG_VARLOWER	VT_R8	
ENUM_TLG_TAG_ARCUPPER	34	VT_TLG_TAG_ARCUPPER	VT_R8	
ENUM_TLG_TAG_ARCLOWER	35	VT_TLG_TAG_ARCLOWER	VT_R8	
ENUM_TLG_TAG_NAMESCALE	36	VT_TLG_TAG_NAMESCALE	VT_BSTR	
ENUM_TLG_TAG_NAMESOURCEARC	37	VT_TLG_TAG_NAMESOURCEARC	VT_BSTR	
ENUM_TLG_TAG_NAMESOURCEVAR	38	VT_TLG_TAG_NAMESOURCEVAR	VT_BSTR	

Definition in ENUM_TLG_TAG _FIELDINDEX	Feldindex	zugehörige Typ- Definition	Typ	Bemerkung / Ein- schränkung
ENUM_TLG_TAG _NAMERAWVAR	39	VT_TLG_TAG_NA MERAWVAR	VT_BSTR	
ENUM_TLG_TAG _CONVDLL	40	VT_TLG_TAG_CO NVDLL	VT_BSTR	
ENUM_TLG_TAG _PROCVARNAME	41	VT_TLG_TAG_PR OCVARNAME	VT_BSTR	erforderlich bei Create
ENUM_TLG_TAG _TIMEMODIFY	42	VT_TLG_TAG_TI MEMODIFY	VT_BSTR	intern festgelegt (neuer Zeitstempel bei Create und Modify), keine Änderung möglich
ENUM_TLG_TAG _ARCON- CHANGE	43	VT_TLG_TAG_AR CONCHANGE	VT_I4	
ENUM_TLG_TAG _ARCONHYSTERE- RESE	44	VT_TLG_TAG_AR CONHYSTERESE	VT_R8	
ENUM_TLG_TAG _FLAGS	45	VT_TLG_TAG_FL AGS	VT_I4	
ENUM_TLG_TAG _ALIAS	46	VT_TLG_TAG_ALI AS	VT_BSTR	
ENUM_TLG_TAG _VARSTARTE- VENT	47	VT_TLG_TAG_VA RSTARTEVENT	VT_BSTR	
ENUM_TLG_TAG _VARSTOPE- VENT	48	VT_TLG_TAG_VA RSTOPEVENT	VT_BSTR	
ENUM_TLG_TAG _PRECISION	49	VT_TLG_TAG_PR ECISION	VT_BSTR	
ENUM_TLG_TAG _TAGID	50	VT_TLG_TAG_TA GID	VT_I4	intern festgelegt, keine Änderung möglich

Flags für Archivvariablen

siehe Include-Datei pde_typ.h ENUM_TLG_TAG_FLAGVALUES ab WinCC V6.2

ENUM_TLG_TAG_FLAGVALUE_NOFLAGS	0	Default
ENUM_TLG_TAG_FLAGVALUE_LONG- TERM_DISABLE	1	
ENUM_TLG_TAG_FLAGVALUE_ARCHI- VING_ON_SEGMENTCHANGE	2	

Selektions-Flags für TLGCSReadTagMulti und TLGCSModifyTagMulti

Diese Funktionen stehen erst ab WinCC V6.2 zur Verfügung.

TLG_TAG_SELECTION_BY_NAME	1	Selektion mit jeweils Archivname und Variablenname.
---	---	---
TLG_TAG_DEFAULT_SELECTION	TLG_TAG_SELECTION_BY_NAME	Bevorzugt zu verwendende Selektions-Art.
TLG_TAG_SELECTION_FLAGS_MASK	0x0000000FL	Maskierung der Selektion in den Flags.
TLG_TAG_SELECTION_FLAGS_OFFSET	0x0L	Wert-Offset der Selektion nach Maskierung in den Flags.

2.8.2 Strukturen (RT Professional)

2.8.2.1 TLG_ARCHIV_STR (RT Professional)

Deklaration

```
typedef struct {
    TCHAR          szName[
        PDE_DB_ARC_NAME_MAX LENGHT+1];
    TCHAR          szComment[
        PDE_DB_ARC_COMMENT_MAX LENGHT+1];
    TCHAR          szServername[
        PDE_DB_ARC_SERVERNAME_MAX LENGHT +1];
    DWORD          dwTyp;
    DWORD          dwAccessRead;
    DWORD          dwAccessWrite;
    TCHAR          szArchivAction[
        PDE_DB_ARC_ACTIONNAME_MAX LENGHT+1];
    BOOL          fLocked;
    DWORD          dwRecordTyp;
    DWORD          dwFillMessage;
    DWORD          dwRecordSize;
    DWORD          dwStorage;
    TCHAR          szCircularAction[
        PDE_DB_ARC_ACTIONNAME_MAX LENGHT+1];
    TCHAR          szCompressTime[
        PDE_DB_TIMENAME_MAX LENGHT+1];
    DWORD          dwSourceProcess;
    TCHAR          szRawDatVar[
        PDE_DB_VAR_NAME_MAX LENGHT+1];
    TCHAR          szSendAct[
        PDE_DB_ARC_ACTIONNAME_MAX LENGHT+1];
    TCHAR          szRecAct[
        PDE_DB_ARC_ACTIONNAME_MAX LENGHT+1];
    DWORD          dwRecItems;
}
TLG_ARCHIV_STR;
```

Members

szName

Globaler Parameter für den Namen des Archives

szComment

Globaler Parameter für den Kommentar zum Projekt.

szServername

Globaler Parameter, für zukünftige Erweiterungen reserviert.

dwTyp

Globaler Parameter, der den Archivtyp kennzeichnet.

TLG_ARCHIV_TYP_PROCESS	Prozesswertarchiv
TLG_ARCHIV_TYP_COMPRESS	Verdichtungsarchiv

dwAccessRead

Globaler Parameter, der die Berechtigungsstufe für lesenden Zugriff (0...999) kennzeichnet.

dwAccessWrite

Globaler Parameter, der die Berechtigungsstufe für schreibenden Zugriff (0...999) kennzeichnet.

szArchivAction

Globaler Parameter mit dem Namen der Aktion zur allgemeinen Archivbearbeitung.

fLocked

Globaler Parameter, der ein Archiv als freigegeben bzw. gesperrt kennzeichnet.

dwRecordTyp

Parameter für Umlauf- und Folgearchive, der den Archivtyp kennzeichnet.

TLG_RECORD_TYP_CIRCULAR	Umlaufarchiv
TLG_RECORD_TYP_FOLLOW	Folgearchiv

dwFillMessage

Parameter für Umlauf-, Folge- und Verdichtungsarchive, der kennzeichnet, wann eine Füllstandsmeldung abgesetzt werden soll. Mit einer ODER Verknüpfung sind max. zwei Füllstandsmeldungen projektierbar.

z.B.: 100% || 50%-90%)

TLG_FILL_MESSAGE_NO	Keine Füllstandsmeldung
TLG_FILL_MESSAGE_50	Füllstandsmeldung bei 50%
TLG_FILL_MESSAGE_60	Füllstandsmeldung bei 60%
TLG_FILL_MESSAGE_70	Füllstandsmeldung bei 70%
TLG_FILL_MESSAGE_80	Füllstandsmeldung bei 80%
TLG_FILL_MESSAGE_90	Füllstandsmeldung bei 90%
TLG_FILL_MESSAGE_100	Füllstandsmeldung bei 100%

dwRecordSize

Parameter für Umlaufarchive, der die Größe des Umlaufpuffers in Datensätzen festlegt.

dwStorage

Parameter für Umlaufarchive, der den Archivierungsort kennzeichnet.

TLG_STORAGE_MEMORY	Umlaufarchiv im Hauptspeicher
TLG_STORAGE_HD	Umlaufarchiv auf Festplatte

szCircularAction

Parameter für Umlaufarchive, Name der Aktion zum Auslagern der Daten eines Umlaufarchives.

szCompressTime

Parameter für Verdichtungsarchive, Name des Zeitobjektes das den Verdichtungszeitraum spezifiziert.

dwSourceProcess

Parameter für Verdichtungsarchive, der die Bearbeitungsart des Quellarchives kennzeichnet.

PDE_COMPSRC_CALC	Berechnen
PDE_COMPSRC_CALCCOPY	Berechnen und Kopieren
PDE_COMPSRC_CALCDEL	Berechnen und löschen
PDE_COMPSRC_CALCCOPYDEL	Berechnen kopieren und löschen

szRawDatVar

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

szSendAct

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

szRecAct

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

dwRecltems

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

Bemerkung

Die Parameter der Struktur TLG_ARCHIV_STR sind abhängig vom Archivtyp (Prozessarchiv, oder verdichtetes Archiv) und der Art der Archivierung (Umlauf- oder Folgearchiv).

Benötigte Dateien

pde_def.h

API - Funktionen

TLGReadArchiv (Seite 893)	Archivparameter lesen
---------------------------	-----------------------

Siehe auch

TLGReadArchiv (Seite 893)

2.8.2.2 TLG_ARCHIVDATARAW (RT Professional)

Deklaration

```
typedef struct {
    SYSTEMTIME    stTime;
    double        doValue;
    DWORD         dwFlags;
}
    TLG_ARCHIVDATARAW;
```

Members

stTime

stTime enthält den Archivierungszeitpunkt.

doValue

doValue enthält den zum Zeitpunkt stTime vorhandenen Wert.

dwFlags

Von Tag Logging werden zu jedem Wert, der ins Archiv geschrieben wird, Flags gesetzt, die Aufschluss über den Zustand der Variablen geben.

Zur Analyse der Flags muss der Wert in seine hexadezimale Darstellung umgewandelt werden. Dabei enthält

- das linke Wort (HighWord) Flags aus dem Datenmanager
- das linke Wort (HighWord) Quality Codes, wenn PDE_RT_QUALITYCODE gesetzt ist
- das rechte Wort (LowWord) Statusflags aus dem Tag Logging

Tag Logging Statusflags (LowWord):		
PDE_RT_DAYLIGHT	0x0001	Sommerzeit, Status kann mit GetTimeZomeInformation ermittelt werden)
PDE_RT_SUBSTITUTION	0x0002	Ersatzwert
PDE_RT_TIME_BEVOR_JUMP	0x0004	Wert vor einem Zeitsprung
PDE_RT_TIME_BEHIND_JUMP	0x0008	Wert nach einem Zeitsprung
PDE_RT_TIME_OVERLAPPED	0x0010	Werte während einer Zeitüberlappung
PDE_RT_LOAD_SYSTEM	0x0020	Erstmalig nach dem Anlegen des Archiv archivierter Wert
PDE_RT_RELOAD_SYSTEM	0x0040	Erster Wert nach Aktivierung von Runtime
PDE_RT_CMPCOPY	0x0080	verdichteter Wert
PDE_RT_TIME_CHANGED	0x0100	Zeitänderung hat stattgefunden
PDE_RT_HAND	0x0200	Variablenversorgung von Hand
PDE_RT_ONCHANGEBACKUPVALUESTART	0x0400	
PDE_RT_ONCHANGEBACKUPVALUESTOP	0x0800	
PDE_RT_QUALITYCODE	0x1000	

Datenmanager Statusflags (HighWord)		
DM_VARSTATE_NOT_ESTABLISHED	0x0001	Verbindung zum Partner nicht aufgebaut
DM_VARSTATE_HANDSHAKE_ERROR	0x0002	Protokollfehler
DM_VARSTATE_HARDWARE_ERROR	0x0004	Netzwerkbaugruppe defekt
DM_VARSTATE_MAX_LIMIT	0x0008	projektierte Obergrenze überschritten
DM_VARSTATE_MIN_LIMIT	0x0010	projektierte Untergrenze unterschritten
DM_VARSTATE_MAX_RANGE	0x0020	Formatgrenze überschritten
DM_VARSTATE_MIN_RANGE	0x0040	Formatgrenze unterschritten
DM_VARSTATE_CONVERSION_ERROR	0x0080	Anzeige Wandlungsfehler, in Verbindung mit MAX/ MIN_RANGE
DM_VARSTATE_STARTUP_VALUE	0x0100	Initialisierungswert der Variablen
DM_VARSTATE_DEFAULT_VALUE	0x0200	Ersatzwert der Variablen
DM_VARSTATE_ADDRESS_ERROR	0x0400	Adressierungsfehler im Kanal
DM_VARSTATE_INVALID_KEY	0x0800	Variable nicht gefunden oder nicht vorhanden
DM_VARSTATE_ACCESS_FAULT	0x1000	Zugriff auf Variable nicht erlaubt
DM_VARSTATE_TIMEOUT	0x2000	Timeout, keine Rückmeldung vom Kanal
DM_VARSTATE_SERVERDOWN	0x4000	Verbindung zum Server unterbrochen oder Server herunter- gefahren

	WinCC Variablenstatus	Quality Code nach Profibus PA/OPC
DM_VARSTATE_SERVERDOWN (0x4000)	Server ist down	Bad, out of service, 0x1C
DM_VARSTATE_HARDWARE_ERROR (0x0004)	Netzwerkbaugruppe defekt	Bad, device failure, 0x0C
DM_VARSTATE_NOT_ESTABLISHED (0x0001)	Verbindung zum Partner nicht aufgebaut	Bad, not connected, 0x08
DM_VARSTATE_TIMEOUT (0x2000)	Timeout, keine Rueckmeldung vom Kanal	Uncertain, last usable value, 0x44
DM_VARSTATE_HANDSHAKE_ERROR (0x0002)	Protokollfehler	Bad, no communication (no usable value), 0x18
DM_VARSTATE_ADDRESS_ERROR (0x0400)	Adressierungsfehler im Kanal	Bad, configuration error, value not accep- ted, 0x04
DM_VARSTATE_INVALID_KEY (0x0800)	Variable nicht gefunden oder nicht vorhanden	Bad, configuration error, value not accep- ted, 0x04
DM_VARSTATE_ACCESS_FAULT (0x1000)	Zugriff auf Variable nicht er- laubt	Bad, configuration error, value not accep- ted, 0x04
DM_VARSTATE_MAX_RANGE (0x0020)	Formatgrenze überschritten	Uncertain, engineering unit range violation, high limit gesetzt, 0x56
DM_VARSTATE_MIN_RANGE (0x0040)	Formatgrenze unterschritten	Uncertain, engineering unit range violation, low limit gesetzt, 0x55
DM_VARSTATE_CONVERSION_ERROR (0x0080)	Anzeige Wandlungsfehler	Uncertain, engineering unit range violation, ohne limits gesetzt, 0x54
DM_VARSTATE_MAX_LIMIT (0x0008)	projektierte Obergrenze über- schritten	Abbilden auf das Limit-Bit HIGH, 0x01 verodern mit dem sonstigen Variablensta- tus, z.B. "good (cascade), ok" ergibt 0xC2

	WinCC Variablenstatus	Quality Code nach Profibus PA/OPC
DM_VARSTATE_MIN_LIMIT (0x0010)	projektierte Untergrenze unterschritten	Abbilden auf das Limit-Bit LOW, 0x02 verodern mit dem sonstigen Variablenstatus, z.B. "good (cascade), ok" ergibt 0xC1
DM_VARSTATE_STARTUP_VALUE (0x0100)	Initialisierungswert der Variablen	Uncertain, initial value, 0x4C
DM_VARSTATE_DEFAULT_VALUE (0x0200)	Ersatzwert der Variablen	Uncertain, substitute-set, 0x48

Bemerkung

Wird eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt, so ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

Setzen Sie das Flag PDE_RT_DAYLIGHT, wenn die entsprechende Zeit in die Sommerzeit fällt. Den Status des Flags können Sie mit der Systemfunktion GetTimeZoneInformation ermitteln (TIME_ZONE_ID_DAYLIGHT).

Benötigte Dateien

- pdertdef.h
- pdert.h
- dmdefs.h

API - Funktionen

TLGGetArchivDataEx (Seite 881)	Daten eines Archives auslesen
--------------------------------	-------------------------------

Siehe auch

- TLGGetArchivDataEx (Seite 881)

2.8.2.3 TLG_BACKUP_TABLE_INFO (RT Professional)

Deklaration

```
typedef struct {
    TCHAR      szArchivName[ _MAX_PATH + 1 ];
    TCHAR      szFileName[ _MAX_PATH + 1 ];
    TCHAR      szComment[ _MAX_PATH + 1 ];
    DWORD      dwFormatFlags;
    DWORD      dwJobFlags;
    DWORD      dwSize;
    SYSTEMTIME sysFrom;
    SYSTEMTIME sysTo;
    DWORD      dwUserData;
}
TLG_BACKUP_TABLE_INFO;
```

Members

szArchivName

Name des Archivs, aus dem Daten ausgelagert werden sollen.

szFileName

Name der Auslagerungsdatei mit Pfad und Extension

szComment

Text des Kommentars zur Auslagerung

dwFormatFlags

Formatspezifizierer:

TLG_BAKFMT_CSV:	CSV-Format (Comma Seperated Value)
-----------------	------------------------------------

dwJobFlags

Auftragsspezifische Kennzeichnung, möglich sind:

TLG_BACKUP_EXPORT:	Einfaches Backup ohne Löschen der Sourcedaten.
TLG_BACKUP_EVACUATE:	Auslagern mit Löschen der Sourcen

dwSize

Größe der zu exportierenden Daten.

sysFrom

Startzeitpunkt von dem aus Daten ausgelagert werden sollen.

sysTo

Endzeitpunkt bis zu dem Daten ausgelagert werden sollen.

dwUserData

applikationsspezifische Daten

Benötigte Dateien

pdertdef.h

API - Funktionen

TLG_ENUMBACKUP_ENTRIES (Seite 911)	Backups auflisten (Callback)
------------------------------------	------------------------------

Siehe auch

TLG_ENUMBACKUP_ENTRIES (Seite 911)

2.8.2.4 TLG_CURVESCALEX (RT Professional)

Deklaration

```

typedef struct {
    DWORD          dwDataTyp;
    DWORD          dwBufferSize;
    DWORD          dwRangeTyp;
    DWORD          dwAxisLocation;
    BOOL           fActualize;
    BOOL           fAutoRange;
    BOOL           fGridLinesBig;
    BOOL           fGridLinesFine;
    BOOL           fGridLinesBigVisible;
    BOOL           fGridLinesFineVisible;
    BOOL           fPercent;
    BOOL           fLimitRange;
    BOOL           fSubstitute;
    SYSTEMTIME     stFrom;
    SYSTEMTIME     stTo;
    double         doFrom;
    double         doTo;
    double         doGridBig;
    double         doGridFine;
    double         doShowDigits;
    double         doLimitUpper1;
    double         doLimitUpper2;
    double         doLimitUpper3;
    double         doLimitLower1;
    double         doLimitLower2;
    double         doLimitLower3;
    double         doDisplayRangeFrom;
    double         doDisplayRangeTo;
    double         doLimitRangeFrom;
    double         doLimitRangeTo;
    COLORREF       crColor;
    COLORREF       crColorTimeOverlapped;
    COLORREF       crColorTimeJump;
    COLORREF       crColorLimitUpper1;
    COLORREF       crColorLimitUpper2;
    COLORREF       crColorLimitUpper3;
    COLORREF       crColorLimitLower1;
    COLORREF       crColorLimitLower2;
    COLORREF       crColorLimitLower3;
    TCHAR          szSelectsSQL[ TLG_MAX_SQL_SELECT ];
    TCHAR          szText[ TLG_MAX_STD_TEXT_NAME ];
    TCHAR          szFunction[ TLG_MAX_FUNCTION_NAME ];
    TCHAR          szDLL[ TLG_MAX_DLL_NAME ];
}
TLG_CURVESCALEX;

```

Members

dwDataTyp

Über dwDataTyp wird festgelegt, welche Datenart der Kurve zugrunde liegt:

TLG_DATATYP_TIMERANGE	0x00000001	für Zeitbereich: Die Daten stammen aus einem zeitgesteuerten Archiv (z.B.: Prozesswertarchiv). Die in stFrom und stTo angegebenen Werte sind gültig. Das Flag fAutoRange hat dann keine Wirkung. ansonsten gilt: Die Daten stammen aus einem datengesteuerten Archiv (z.B.: Anwenderarchiv). Die in doFrom und doTo angegebenen Werte sind gültig.
TLG_DATATYP_USERARCHIV	0x00000002	für Kurven nicht relevant: Die Daten stammen aus einem datengesteuerten Anwenderarchiv. Die in doFrom und doTo angegebenen Werte sind gültig.
TLG_DATATYP_BLOCKDATA	0x00000004	Daten werden über TLGSetTemplateData gesetzt: Die Werte in stFrom, stTo, doFrom und doTo sind dann nicht relevant.

dwBufferSize

dwBufferSize entspricht der Anzahl der Messpunkte der Kurve und legt die Größe des Datenbuffers der Kurve fest.

dwRangeTyp

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

dwAxisLocation

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

fActualize

Wenn fActualize = TRUE ist erfolgt die Kurvendarstellung dynamisch. Andernfalls erfolgt sie statisch.

fAutoRange

Wenn fAutoRange = TRUE ist, dann ist die Autoskalierung aktiviert.

Liegt ein Zeitgesteuertes Archiv vor (dwDataTyp = TLG_DATATYP_TIMERANGE), hat fAutoRange keine Wirkung und die in stFrom und stTo angegebenen Werte sind gültig.

fGridLinesBig

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

fGridLinesFine

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

fGridLinesBigVisible

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

fGridLinesFineVisible

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

fPercent

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

fLimitRange

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

fSubstitute

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

stFrom

Startzeitpunkt des Darstellungsbereiches. Der Parameter wird nur ausgewertet, wenn `dwDataTyp = TLG_TIME_RANGE`.

stTo

Endzeitpunkt des Darstellungsbereiches. Der Parameter wird nur ausgewertet, wenn `dwDataTyp = TLG_TIME_RANGE`.

Wird eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt, so ist die Funktion `GetLocalTime` zu verwenden und nicht `GetSystemTime`. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

doFrom

Startwert des Darstellungsbereiches. Der Parameter wird nur ausgewertet, wenn `dwDataTyp = TLG_DATA_RANGE`.

doTo

Endwert des Darstellungsbereiches. Der Parameter wird nur ausgewertet, wenn in `dwDataTyp = TLG_DATA_RANGE`.

doGridBig

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doGridFine

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doShowDigits

Anzahl der Nachkommastellen

doLimitUpper1

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doLimitUpper2

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doLimitUpper3

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitLower1

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitLower2

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitLower3

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doDisplayRangeFrom

Gibt an ab welchem Wert eine Datengesteuerte Anzeige erfolgen soll.

doDisplayRangeTo

Gibt an bis zu welchem Wert eine Datengesteuerte Anzeige erfolgen soll.

doLimitRangeFrom

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitRangeTo

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

crColor

Der Windows spezifische 32-Bit-Wert crColor legt die Farbe fest, in der die Kurve dargestellt wird.

crColorTimeOverlapped

Der Parameter ist für spätere Erweiterungen reserviert.

crColorTimeJump

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper1

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper2

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper3

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower1

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower2

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower3

Der Parameter ist für spätere Erweiterungen reserviert.

szSelectSQL

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

szText;

Beschriftung der x-Achse.

szFunction

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

szDLL

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

Bemerkung

TLG_CURVESCALEX wird von der Struktur TLG_TPLITEM_CURVE (Seite 835) verwendet.

Benötigte Dateien

pdertdef.h

Siehe auch

TLG_TPLITEM_CURVE (Seite 835)

2.8.2.5 TLG_CURVESCALEY (RT Professional)

Deklaration

```

typedef struct {
    DWORD      dwDataTyp;
    DWORD      dwRangeTyp;
    DWORD      dwAxisLocation;
    DWORD      dwCurveForm;
    BOOL       fAutoRange;
    BOOL       fGridLinesBig;
    BOOL       fGridLinesFine;
    BOOL       fGridLinesBigVisible;
    BOOL       fGridLinesFineVisible;
    BOOL       fPercent;
    BOOL       fLimitRange;
    BOOL       fSubstitute;
    SYSTEMTIME stFrom;
    SYSTEMTIME stTo;
    double     doFrom;
    double     doTo;
    double     doGridBig;
    double     doGridFine;
    double     doShowDigitsF;
    double     doShowDigitsB;
    double     doLimitUpper1;
    double     doLimitUpper2;
    double     doLimitUpper3;
    double     doLimitLower1;
    double     doLimitLower2;
    double     doLimitLower3;
    double     doDisplayRangeFrom;
    double     doDisplayRangeTo;
    double     doLimitRangeFrom;
    double     doLimitRangeTo;
    COLORREF   crColor;
    COLORREF   crColorTimeOverlapped;
    COLORREF   crColorTimeJump;
    COLORREF   crColorLimitUpper1;
    COLORREF   crColorLimitUpper2;
    COLORREF   crColorLimitUpper3;
    COLORREF   crColorLimitLower1;
    COLORREF   crColorLimitLower2;
    COLORREF   crColorLimitLower3;
    TCHAR      szSelectSQL[ TLG_MAX_SQL_SELECT ];
    TCHAR      szText[ TLG_MAX_STD_TEXT_NAME ];
}
TLG_CURVESCALEY;

```

Members

dwDataType

Der Datentyp (Zeit/Anwender/Blockdaten) ist hier ohne Bedeutung

dwRangeTyp

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

dwAxisLocation

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

dwCurveForm

Über dwCurveForm wird das Aussehen der Kurve festgelegt

TLG_CURVEFORM_STEP	Darstellung der Kurve als Treppenkurve
TLG_CURVEFORM_POINTS	Darstellung der Kurve als Einzelpunkt
TLG_CURVEFORM_DIRECT	Punkte der Kurve linear verbinden

fAutoRange

fAutoRange = TRUE passt den Darstellungsbereich automatisch an.

fGridLinesBig

fgridLinesBig = TRUE aktiviert grobe Rasterlinien.

fGridLinesFine

fgridLinesFine = TRUE aktiviert feine Rasterlinien.

fGridLinesBigVisible

Wenn fgridLinesBigVisible = TRUE ist, werden die groben Rasterlinien dargestellt.

fGridLinesFineVisible

Wenn fgridLinesFineVisible = TRUE ist, werden die feinen Rasterlinien dargestellt.

fPercent

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

fLimitRange

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

fSubstitute

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

stFrom

Der Parameter ist für spätere Erweiterungen reserviert.

stTo

Der Parameter ist für spätere Erweiterungen reserviert.

doFrom

Untergrenze des Darstellungsbereiches auf der y-Achse.

doTo

Obergrenze des Darstellungsbereiches auf der y-Achse.

doGridBig

Abstand der groben Rasterlinien.

doGridFine

Abstand der feinen Rasterlinien.

doShowDigitsF

Stellen bei Anzeige von Werten (Vorkomma)

doShowDigitsB

Legt die Anzahl der Nachkommastellen

doLimitUpper1

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitUpper2

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitUpper3

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitLower1

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitLower2

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitLower3

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doDisplayRangeFrom

Legt den Wert fest, ab dem der Darstellungsbereich beginnt.

doDisplayRangeTo

Legt den Wert fest, bis zu dem der Darstellungsbereich reicht.

doLimitRangeFrom

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doLimitRangeTo

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

crColor

Der Windows spezifische 32-Bit-Wert crColor legt die Farbe fest, in der die Kurve dargestellt wird.

crColorTimeOverlapped

Der Parameter ist für spätere Erweiterungen reserviert.

crColorTimeJump

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper1

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper2

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper3

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower1

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower2

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower3

Der Parameter ist für spätere Erweiterungen reserviert.

szSelectSQL[TLG_MAX_SQL_SELECT];

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

szText[TLG_MAX_STD_TEXT_NAME];

Beschriftung der y-Achse.

Bemerkung

TLG_CURVESCALEY wird von der Struktur AUTOHOTSPOT verwendet.

Benötigte Dateien

pdertdef.h

Siehe auch

TLG_TPLITEM_CURVE (Seite 835)

2.8.2.6 TLG_GETARCHIVDATA (RT Professional)

Deklaration

```
typedef struct {  
    LPTSTR      lpszArchivName;  
    LPTSTR      lpszVarName;  
    SYSTEMTIME  stTime;  
    double      doValue;  
    DWORD       dwFlags;  
}  
TLG_GETARCHIVDATA;
```

Members

lpszArchivName

Zeiger auf den Archivnamen aus dem die Daten gelesen wurden.

lpszVarName

Zeiger auf den Variablennamen deren Werte gelesen wurden.

stTime

x-Wert des Archivwertes.

doValue

y-Wert des Archivwertes.

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

Benötigte Dateien

pdertdef.h

API - Funktionen

TLG_GETARCHIVDATA_CALLBACK (Seite 879)	Daten eines Archives auslesen (Callback)
--	--

Siehe auch

TLG_GETARCHIVDATA_CALLBACK (Seite 879)

2.8.2.7 TLG_IO_BACKUP_SELECT (RT Professional)

Deklaration

```
typedef struct {
    SYSTEMTIME    sysFrom;
    SYSTEMTIME    sysTo;
    LPTSTR        lpszSqlString;
}
TLG_IO_BACKUP_SELECT;
```

Members

sysFrom

Systemzeit des ersten zu selektierenden Datensatzes

sysTo

Systemzeit des letzten zu selektierenden Datensatzes.

lpszSqlString

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbesetzt werden.

Benötigte Dateien

pdertdef.h

API - Funktionen

TLGGetBackupSize (Seite 914)	Größe eines exportierten Datensatzes bestimmen
TLGExport (Seite 912)	Archivdaten exportieren

Siehe auch

TLGGetBackupSize (Seite 914)

TLGExport (Seite 912)

2.8.2.8 TLG_PROT_CURVE_INFOS (RT Professional)

Deklaration

```
typedef struct {
    TCHAR          szArchivName[ 128 + 1 ];
    TCHAR          szVariableName[ 128 + 1 ];
    TCHAR          szTextX[ 128 + 1 ];
    TCHAR          szTextY[ 128 + 1 ];
    SYSTEMTIME     stFrom;
    SYSTEMTIME     stTo;
    double         doFrom;
    double         doTo;
    DWORD          dwFlags;
    DWORD          dwCurveForm;
}
    TLG_PROT_CURVE_INFOS;
```

Members

szArchivName

Name des Archives aus dem die zu protokollierenden Daten stammen.

szVariableName

Name der Archivvariablen deren Werte protokolliert werden.

szTextX

Achsenbeschriftung der x-Achse.

szTextY

Achsenbeschriftung der y-Achse.

stFrom

Startzeitpunkt, von dem ab protokolliert wird. Der Parameter wird nur ausgewertet, wenn in dwFlags TLG_PROTFLG_TIME_RANGE gesetzt ist.

stTo

Endzeitpunkt, bis zu dem protokolliert wird. Der Parameter wird nur ausgewertet, wenn in dwFlags TLG_PROTFLG_TIME_RANGE gesetzt ist.

Wenn eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt wird, ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

doFrom

Startzeitpunkt, von zu dem ab die Protokollierung erfolgen soll. Der Parameter wird nur ausgewertet, wenn in dwFlags TLG_PROTFLG_DATA_RANGE gesetzt ist.

doTo

Endzeitpunkt, bis zu dem die Protokollierung erfolgen soll. Der Parameter wird nur ausgewertet, wenn in dwFlags TLG_PROTFLG_DATA_RANGE gesetzt ist.

dwFlags

Über dwFlags wird festgelegt, welche Datenart der Kurve zugrunde liegt:

TLG_PROTFLG_TIME_RANGE	Die Daten stammen aus einem zeitgesteuerten Archiv (z.B.: Prozesswertarchiv). Die in stFrom und stTo angegebenen Werte sind gültig.
TLG_PROTFLG_DATA_RANGE	Die Daten stammen aus einem datengesteuerten Archiv (z.B.: Anwenderarchiv). Die in doFrom und doTo angegebenen Werte sind gültig.

dwCurveForm

Über dwCurveForm wird das Aussehen der Kurve festgelegt

TLG_CURVEFORM_STEP	Darstellung der Kurve als Treppenkurve
TLG_CURVEFORM_POINTS	Darstellung der Kurve als Einzelpunkt
TLG_CURVEFORM_DIRECT	Punkte der Kurve linear verbinden

Bemerkung

TLG_PROT_CURVE_INFOS wird von den Strukturen TLG_TABLESCALE (Seite 828) und TLG_TPLITEM_INFO (Seite 836) verwendet.

Benötigte Dateien

pdertdef.h

API - Funktionen

TLGDrawCurvesInDC (Seite 895)	Kurven darstellen
-------------------------------	-------------------

Siehe auch

TLGDrawCurvesInDC (Seite 895)

TLG_TABLESCALE (Seite 828)

TLG_TPLITEM_INFO (Seite 836)

2.8.2.9 TLG_SCAL_STR (RT Professional)

Deklaration

```
typedef struct {  
    double    doScalVarUpper;  
    double    doScalVarLower;  
    double    doScalArcUpper;  
    double    doScalArcLower;  
    TCHAR     szStructName[PDE_DB_SCALE_MAX LENGHT];  
}  
TLG_SCAL_STR;
```

Members

doScalVarLower

Untere Grenze des Variablenwertes.

doScalVarUpper

Obere Grenze des Variablenwertes.

doScalArcLower

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

doScalArcUpper

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

szStructName

Name der Skalierungsstruktur für die Variablengrenzwerte.

Bemerkung

Die Struktur TLG_SCAL_STR wird innerhalb der Struktur TLG_VAR_STR (Seite 838) verwendet und ist für spätere Erweiterungen vorgesehen.

Benötigte Dateien

pde_def.h

Siehe auch

TLG_VAR_STR (Seite 838)

2.8.2.10 TLG_TABLE_INFO (RT Professional)

Deklaration

```
typedef struct {
    DWORD    dwArchivTyp;
    TCHAR    szArchivName[ _MAX_PATH + 1 ];
    DWORD    dwSaveTyp;
}
TLG_TABLE_INFO;
```

Members

dwArchivTyp

dwArchivTyp kennzeichnet den Archivtyp:

TLG_ARCTYP_USER	Anwenderarchiv
TLG_ARCTYP_PROCESS	Prozesswertarchiv
TLG_ARCTYP_COMPRESS	Verdichtungsarchiv

szArchivName

In welcher Form der Name des Archives bereitgestellt wird ist abhängig von den Funktionen, die TLG_TABLE_INFO verwenden.

Die Enumeration der Archive über TLGEnumArchivs und TLGEnumArchivsSel liefert den Archivnamen als Tabellennamen, d.h. in der Form "UA#ARCHIV#Archivname" für Userarchive bzw. "PDE#HD#Archivname#Variablenname", da hier für jede Variable ein Tabelleneintrag existiert. Es wird also für jede Variable enumeriert.

Die Enumeration über TLGEnumArchivsEx liefert den "reinen" Archivnamen.

dwSaveTyp

dwSaveTyp kennzeichnet die Art des Archives:

TLG_ARCTYP_CIRCULAR	Umlaufarchiv
TLG_ARCTYP_FOLLOW	Folgearchiv

Benötigte Files

pdertdef.h

API - Funktionen

TLG_ENUMTABLES (Seite 874)	Archive auflisten (Callback)
----------------------------	------------------------------

Siehe auch

TLG_ENUMTABLES (Seite 874)

2.8.2.11 TLG_TABLESCALE (RT Professional)

Deklaration

```

typedef struct {
    BOOL          fActualize;
    BOOL          fVisible;
    BOOL          fModify;
    BOOL          fCommon;
    DWORD         dwDataTyp;
    SYSTEMTIME    stFrom;
    SYSTEMTIME    stTo;
    double        doFrom;
    double        doTo;
    double        doShowDigits;
    double        doLimitUpper1;
    double        doLimitUpper2;
    double        doLimitUpper3;
    double        doLimitLower1;
    double        doLimitLower2;
    double        doLimitLower3;
    OLORREF       crColor;
    COLORREF      crColorTimeOverlapped;
    COLORREF      crColorTimeJump;
    COLORREF      crColorLimitUpper1;
    COLORREF      crColorLimitUpper2;
    COLORREF      crColorLimitUpper3;
    COLORREF      crColorLimitLower1;
    COLORREF      crColorLimitLower2;
    COLORREF      crColorLimitLower3;
    TCHAR         szSelectSQL[ TLG_MAX_SQL_SELECT ];
}
TLG_TABLESCALE;

```

Members

fActualize

TRUE	Die Kurvendarstellung erfolgt dynamisch.
FALSE	Die Kurvendarstellung erfolgt statisch.

fVisible

TRUE	Die Spalte ist sichtbar.
FALSE	Die Spalte ist nicht sichtbar.

fModify

TRUE	Die Einträge in der Spalte sind editierbar.
FALSE	Die Einträge in der Spalte sind nicht editierbar.

fCommon

TRUE	Alle Spalten der Tabelle besitzen eine gemeinsame Zeitspalte.
FALSE	Die Spalten der Tabelle besitzen keine gemeinsame Zeitspalte.

dwDataTyp

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

stFrom

Startzeitpunkt des Darstellungsbereichs.

stTo

Endzeitpunkt des Darstellungsbereichs.

Wenn eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt wird, ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

doFrom

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doTo

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doShowDigits

Anzahl der auszugebenden Nachkommastellen.

doLimitUpper1

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doLimitUpper2

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

dpoLimitUpper3

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doLimitLower1

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doLimitLower2

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

doLimitLower3

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

crColor

Der Windows spezifische 32-Bit-Wert crColor legt die in der Tabellenspalte verwendete Farbe fest.

crColorTimeOverlapped

Der Parameter ist für spätere Erweiterungen reserviert.

crColorTimeJump

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper1

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper2

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitUpper3

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower1

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower2

Der Parameter ist für spätere Erweiterungen reserviert.

crColorLimitLower3

Der Parameter ist für spätere Erweiterungen reserviert.

szSelectSQL

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbesetzt werden.

Bemerkung

TLG_TABLESCALE wird von der Struktur TLG_TEMPLATEITEM_INFO (Seite 831) verwendet.

Benötigte Dateien

pdertdef.h

Siehe auch

TLG_PROT_CURVE_INFOS (Seite 824)

TLG_TPLITEM_INFO (Seite 836)

TLG_TEMPLATEITEM_INFO (Seite 831)

TLG_TPLITEM_TABLE (Seite 837)

2.8.2.12 TLG_TEMPLATEITEM_INFO (RT Professional)

Deklaration

```
typedef struct {
    TCHAR                szTemplateName[
                        TLG_MAX_TEMPLATEITEM_NAME+1 ];
    TCHAR                szTemplateName[
                        TLG_MAX_TEMPLATE_NAME + 1 ];
    TCHAR                szArchivName[ 128 + 1 ];
    TCHAR                szVariableName[ 128 + 1 ];
    TCHAR                szDMVariableName[ 128 + 1 ];
    DWORD               dwReadAccessLevel;
    DWORD               dwWriteAccessLevel;
    DWORD               dwArchivTyp;
    TCHAR                szTimeNameRange[ 128 + 1 ];
    DWORD               dwTemplateItemTyp;
    DWORD               dwTemplateTyp;
    BOOL                fVisible;
    TLG_TPLITEM_INFO    tplInfo;
}
TLG_TEMPLATEITEM_INFO;
```

Members

szTemplateItemName

Name der Kurve bzw. Spalte

szTemplateName

Name der Kurven- bzw. Tabellenfenstervorlage

szArchivName

Name des Archivs, in dem die mit der Kurve bzw. Spalte verknüpften Archivvariable archiviert ist.

szVariableName

Name der mit der Kurve bzw. Spalte verknüpften Archivvariablen

szDMVariableName

Name der Datenmanagervariablen

dwReadAccessLevel

Bei der Enumeration von Kurven bzw. Spalten enthält dwReadAccessLevel die Benutzerberechtigungsstufe für einen lesenden Zugriff.

dwWriteAccessLevel

Bei der Enumeration von Kurven bzw. Spalten enthält dwWriteAccessLevel die Benutzerberechtigungsstufe für einen schreibenden Zugriff.

dwArchivTyp

dwArchivTyp kennzeichnet den Archivtyp:

TLG_ARCTYP_USER	Anwenderarchiv
TLG_ARCTYP_PROCESS	Prozesswertarchiv
TLG_ARCTYP_COMPRESS	Verdichtungsarchiv

szTimeNameRange

Name des Zeitobjektes, das zur Festlegung eines Zeitbereiches ab einer Startzeit verwendet werden soll.

dwTemplatItemTyp

dwTemplatItemTyp muss dem in dwTemplateTyp angegebenen Wert entsprechen:

TLG_TEMPLATEITEM_ALL	Kurven- und Tabellenvorlagen
TLG_TEMPLATEITEM_CURVE	Kurven-Vorlage
TLG_TEMPLATEITEM_TABLE	Tabellen-Vorlage

dwTemplateTyp

dwTemplateTyp kennzeichnet den Typ der Fenstervorlage

TLG_TEMPLATE_CURVE	Kurvenfenster-Vorlage
TLG_TEMPLATE_TABLE	Tabellenfenstervorlage

fVisible

TRUE	Die Spalte bzw. die Kurve ist sichtbar.
FALSE	Die Spalte bzw. die Kurve ist nicht sichtbar.

tplInfo

Struktur vom Typ TLG_TPLITEM_INFO (Seite 836) mit den Eigenschaften einer Kurven- bzw. Spaltenvorlage.

Benötigte Dateien

pdertdef.h

API - Funktionen

TLG_TABLESCALE (Seite 828)	Tabellenspalte (Eigenschaften) (Struktur)
----------------------------	---

Siehe auch

TLG_TPLITEM_INFO (Seite 836)
 TLG_TABLESCALE (Seite 828)
 TLGInsertTemplateltem (Seite 897)

2.8.2.13 TLG_TIME_STR (RT Professional)**Deklaration**

```
typedef struct {
    DWORD          dwBasis;
    DWORD          dwFactor;
    TCHAR          szTimeName[
        PDE_DB_TIMENAME_MAX_LENIGHT + 1 ];
}
    TLG_TIME_STR;
```

Members**dwBasis**

Zeitbasis. Die Zykluszeit ergibt sich aus der Multiplikation von Zeitfaktor und Zeitbasis.

TLG_TBASE_500MS	500ms
TLG_TBASE_SEC	1 Sekunde
TLG_TBASE_MIN	1 Minute
TLG_TBASE_HOUR	1 Stunde
TLG_TBASE_DAY	1 Tag

dwFactor

Zeitfaktor. Die Zykluszeit ergibt sich aus der Multiplikation von Zeitfaktor und Zeitbasis.

szTimeName

Name des Zeitobjektes.

Bemerkung

Unter Zeiten werden Erfassungs und Archivierungszyklen verstanden die frei vergeben werden können. Dabei kann ein Zeitobjekt gleichzeitig als Erfassungszyklus und Archivierungszyklus eingesetzt werden.

Benötigte Dateien

pde_def.h

API - Funktionen

TLGReadTime (Seite 908)	Zeitobjekt-Parameter ermitteln
-------------------------	--------------------------------

Siehe auch

TLGReadTime (Seite 908)

2.8.2.14 TLG_TIMEDATA (RT Professional)

Deklaration

```
typedef struct {
    TCHAR    szTimeName[ _MAX_PATH + 1 ];
    DWORD    dwTimeBase;
    DWORD    dwTimeValue;
}
TLG_TIMEDATA;
```

Members

szTimeName

Name des Zeitobjektes

dwTimeBase

Zeitbasis. Die Zykluszeit ergibt sich aus der Multiplikation von Zeitfaktor und Zeitbasis

TLG_TBASE_500MS	500ms
TLG_TBASE_SEC	1 Sekunde
TLG_TBASE_MIN	1 Minute
TLG_TBASE_HOUR	1 Stunde
TLG_TBASE_DAY	1 Tag

dwTimeValue

Zeitfaktor. Die Zykluszeit ergibt sich aus der Multiplikation von Zeitfaktor und Zeitbasis.

Benötigte Dateien

pdertdef.h

API - Funktionen

TLG_ENUMTIMES_CALLBACK (Seite 907)	Zeitobjekte auflisten (Callback)
------------------------------------	----------------------------------

Siehe auch

TLG_ENUMTIMES_CALLBACK (Seite 907)

2.8.2.15 TLG_TPLITEM_CURVE (RT Professional)**Deklaration**

```
typedef struct {  
    TLG_CURVESCALEX    csx;  
    TLG_CURVESCALEY    csy;  
}  
TLG_TPLITEM_CURVE;
```

Members**csx**

Die Struktur TLG_CURVESCALEX (Seite 813) enthält die Daten zur Skalierung der X-Achse einer Kurve.

csy

Die Struktur TLG_CURVESCALEY (Seite 818) enthält die Daten zur Skalierung der Y-Achse einer Kurve.

Beschreibung

TLG_TPLITEM_CURVE wird von der Struktur TLG_TPLITEM_INFO (Seite 836) verwendet.

Benötigte Dateien

pdertdef.h

Siehe auch

TLG_TPLITEM_INFO (Seite 836)

TLG_CURVESCALEX (Seite 813)

TLG_CURVESCALEY (Seite 818)

2.8.2.16 TLG_TPLITEM_INFO (RT Professional)

Deklaration

```
typedef union {  
    TLG_TPLITEM_CURVE    tplCurve;  
    TLG_TPLITEM_TABLE    tplTable;  
}  
TLG_TPLITEM_INFO;
```

Members

tplCurve

Die Eigenschaften einer Kurve werden aus der Struktur TLG_TPLITEM_CURVE (Seite 835) gelesen.

tplTable

Die Eigenschaften einer Spalte werden aus der Struktur TLG_TPLITEM_TABLE (Seite 837) gelesen.

Bemerkung

TLG_TPLITEM_INFO wird von der Struktur TLG_TEMPLATEITEM_INFO (Seite 831) verwendet.

Da eine Variable sowohl in einer Kurve als auch in einer Tabelle dargestellt werden kann, können beide Strukturen TLG_TPLITEM_CURVE und TLG_TPLITEM_TABLE mit Werten belegt sein.

Benötigte Dateien

pdertdef.h

Siehe auch

TLG_PROT_CURVE_INFOS (Seite 824)

TLG_TABLESCALE (Seite 828)

TLG_TEMPLATEITEM_INFO (Seite 831)

TLG_TPLITEM_CURVE (Seite 835)

TLG_TPLITEM_TABLE (Seite 837)

2.8.2.17 TLG_TPLITEM_TABLE (RT Professional)

Deklaration

```
typedef struct {  
    TLG_TABLESCALE    ts;  
}  
TLG_TPLITEM_TABLE;
```

Members

ts

TLG_TABLESCALE (Seite 828) mit den Eigenschaften einer Tabellenspalte.

Beschreibung

TLG_TPLITEM_TABLE wird von der Struktur TLG_TPLITEM_INFO (Seite 836) verwendet.

Aus organisatorischen Gründen kann TLG_TABLESCALE nicht direkt in der Struktur TLG_TPLITEM_INFO verwendet werden.

Benötigte Dateien

pdertdef.h

Siehe auch

TLG_TPLITEM_INFO (Seite 836)

TLG_TABLESCALE (Seite 828)

2.8.2.18 TLG_VAR_STR (RT Professional)

Deklaration

```

typedef struct {
    TCHAR          szVarName [PDE_DB_VAR_NAME_MAX LENGHT + 1];
    TCHAR          szProcName [PDE_DB_VAR_NAME_MAX LENGHT + 1];
    DWORD          dwVarType;
    DWORD          dwArchivStyle;
    TCHAR          szWriteBackTo [PDE_DB_VAR_WRITEBACK_MAX LENGHT + 1];
    DWORD          dwSupply;
    BOOL           fLocked;
    TCHAR          szComment [PDE_DB_VAR_COMMENT_MAX LENGHT + 1];
    TCHAR          szRecordCycle [PDE_DB_VAR_CYCLENAME_MAX LENGHT + 1];
    TCHAR          szArchivCycle [PDE_DB_VAR_CYCLENAME_MAX LENGHT + 1];
    DWORD          dwMultiple;
    DWORD          dwValueFlow;
    DWORD          dwValueFollow;
    DWORD          dwSaveByFault;
    DWORD          dwArchivTrigger;
    TCHAR          szTextHighSignal [PDE_DB_VAR_SIGNALTEX_MAX LENGHT + 1];
    TCHAR          szTextLowSignal [PDE_DB_VAR_SIGNALTEX_MAX LENGHT + 1];
    DWORD          dwValProcess;
    TCHAR          szFuncValProcess [PDE_DB_VAR_FUNCNAME_MAX LENGHT + 1];
    TCHAR          szDLLValProcess [PDE_DB_VAR_DLLNAME_MAX LENGHT + 1];
    TCHAR          szFuncStartEvent [PDE_DB_VAR_FUNCNAME_MAX LENGHT + 1];
    TCHAR          szDLLStartEvent [PDE_DB_VAR_DLLNAME_MAX LENGHT + 1];
    TCHAR          szFuncStopEvent [PDE_DB_VAR_FUNCNAME_MAX LENGHT + 1];
    TCHAR          szDLLStopEvent [PDE_DB_VAR_DLLNAME_MAX LENGHT + 1];
    TCHAR          szFuncDynamic [PDE_DB_VAR_FUNCNAME_MAX LENGHT + 1];
    TCHAR          szDLLDynamic [PDE_DB_VAR_DLLNAME_MAX LENGHT + 1];
    TCHAR          szUnitDirect [PDE_DB_VAR_UNITDIR_MAX LENGHT + 1];
    TCHAR          szUnitStruct [PDE_DB_VAR_UNITSTR_MAX LENGHT + 1];
    DWORD          dwRecItems;
    TCHAR          szSourceArchiv [PDE_DB_ARC_NAME_MAX LENGHT + 1];
    TCHAR          szSourceVarName [PDE_DB_VAR_NAME_MAX LENGHT + 1];
    TCHAR          szRawdataName [PDE_DB_VAR_NAME_MAX LENGHT + 1];
    TCHAR          szRawConvDLLName [TLG_DB_DLLNAME_MAX LENGHT + 1];
    DWORD          dwRawDataIndex;
    DWORD          dwRawDataFormat;
    TLG_SCAL_STR   ScaleStruct;
    TLG_RECORD_STR RecordStruct;
} TLG_VAR_STR;

```

Bemerkung

Die Besetzung der Parameter der Struktur TLG_VAR_STR ist abhängig von der Archivierungsart (azyklisch, zyklisch-selektiv oder zyklisch-kontinuierlich) und dem Variablentyp (binär oder analog).

Members

szVarName

Globaler Parameter für den Name der Archivvariablen bzw. der Variablengruppe.

Die Zeichen : ? " ' \ * % und Blank dürfen nicht im Archivvariablen-Namen verwendet werden, ansonsten wird der Fehler TLG_API_NAME_WRONG_CHAR geliefert.

szProcName

Globaler Parameter für den Namen der Prozessvariablen

dwVarType

Globaler Parameter, der den Typ der Archivvariablen kennzeichnet.

TLG_VAR_TYP_ANALOG	0x00010001	analoge Variable: dwArchivTrigger wird intern auf 0L gesetzt (nicht relevant)
TLG_VAR_TYP_BIN	0x00010002	binäre Variable: dwValProcess wird intern auf 0L gesetzt (nicht relevant)
TLG_VAR_TYP_COMPRESS	0x00010004	Variable für Verdichtungsarchive: dwSupply wird intern auf TLG_SUPPLY_BY_SYSTEM gesetzt dwSaveByFault wird intern auf TLG_SAVE_LAST_VALUE gesetzt
TLG_VAR_TYP_PROCESS	0x00010008	prozessgesteuerte Variable
	ungültiger Wert	Fehlermeldung: TLG_API_VARTYPE_MISMATCH

dwArchivStyle

Globaler Parameter, der die Archivierungsart kennzeichnet.

TLG_STY_ACYCLIC	0x00800001	Azyklische Archivierung
TLG_STY_CYCLIC_CON	0x00800002	zyklisch-kontinuierliche Archivierung
TLG_STY_CYCLIC_SEL	0x00800004	zyklisch-selektive Archivierung
TLG_STY_ON_CHANGE	0x00800008	Erfassung bei Änderung
	ungültiger Wert	Fehlermeldung TLG_API_ERR_ARCHIVSTYLE

szWriteBackTo

Globaler Parameter in den der Name der Archivvariable zurückgeschrieben werden soll.

dwSupply

Globaler Parameter, der die Art der Variablenversorgung kennzeichnet.

TLG_SUPPLY_BY_HAND	0L - FALSE	Handeingabe der Variablen
TLG_SUPPLY_BY_SYSTEM	1L - TRUE	Versorgung vom System (Prozesskopplung oder interne Variable)
	ungültiger Wert	Fehlermeldung TLG_API_INVALID_PARAM

Ist dwVarType = TLG_VAR_TYP_COMPRESS wird intern immer TLG_SUPPLY_BY_SYSTEM gesetzt.

fLocked

Globaler Parameter, der die Behandlung der Archivierung beim Systemstart kennzeichnet.

TRUE	gesperrt
FALSE	freigegeben

szComment

Globaler Parameter für den Kommentar zur Archivvariablen.

szRecordCycle

Name des Zeitobjektes (z.B.: "1 Minute"), das den Erfassungszyklus spezifiziert. Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Erfassung von analogen und binären Variablen.

szArchivCycle

Name des Zeitobjektes (z.B.: "1 Minute"), das den Archivierungszyklus spezifiziert. Der Archivierungszyklus ergibt sich aus der Multiplikation von dwMultiple und szArchivCycle. Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen.

dwMultiple

Kennzeichnet den Multiplikationsfaktor. Der Archivierungszyklus ergibt sich aus der Multiplikation von dwMultiple und szArchivCycle. Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen.

dwValueFlow

Kennzeichnet die Anzahl der Vorlauf-Werte. Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen.

dwValueFollow

Kennzeichnet die Anzahl der Rücklauf-Werte. Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen.

dwSaveByFault

Kennzeichnet das Speicherverhalten bei einer Störung. Der Parameter ist relevant für analoge und binäre Variablen aller Archivierungsarten.

TLG_SAVE_LAST_VALUE	1L	letzten gültigen Wert speichern
TLG_SAVE_SUBST_VALUE	2L	Ersatzwert speichern
	ungültiger Wert	Fehlermeldung TLG_API_INVALID_PARAM

Ist dwVarType = TLG_VAR_TYP_COMPRESS wird intern immer TLG_SAVE_LAST_VALUE gesetzt.

dwArchivTrigger

Kennzeichnet das Triggerverhalten der Archivierung.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von binären Variablen.

TLG_TRIG_CHANGE	1L	Archivierung bei jedem Signalwechsel
TLG_TRIG_CHANGE_01	2L	Archivierung bei Signalwechsel 0 -> 1
TLG_TRIG_CHANGE_10	3L	Archivierung bei Signalwechsel 1 -> 0
TLG_TRIG_ALLWAYS	4L	Der Wert wird in jedem Archivierungszyklus, auch ohne Signalwechsel, archiviert..
	ungültiger Wert	Fehlermeldung TLG_API_INVALID_PARAM

Ist dwVarType = TLG_VAR_TYP_ANALOG wird intern immer 0L gesetzt (nicht relevant).

szTextHighSignal

Text für den Signalzustand 1.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von binären Variablen.

szTextLowSignal

Text für den Signalzustand 0.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von binären Variablen.

dwValProcess

Kennzeichnet die Bearbeitung des abzuspeichernden Archivwertes über die erfassten Werte innerhalb des Archivierungszyklus.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen Variablen.

TLG_VAL_ACTUAL	1L	Momentanwert
TLG_VAL_AVERAGE	2L	Mittelwert
TLG_VAL_SUM	3L	Summe
TLG_VAL_MIN	4L	Minimalwert
TLG_VAL_MAX	5L	Maximalwert
TLG_VAL_DLL	6L	Funktion oder DLL
	ungültiger Wert	Fehlermeldung TLG_API_INVALID_PARAM

Ist dwVarType = TLG_VAR_TYP_BIN wird intern immer 0L gesetzt (nicht relevant).

szFuncValProcess

Name der Aktion oder der DLL-Funktion, die die Bearbeitung des abzuspeichernden Archivwertes spezifiziert.

Der Parameter ist für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und Verdichtungsvariablen und nur relevant, wenn dwValProcess = TLG_VAL_DLL.

szDLLValProcess

Name der DLL.

Der Parameter ist für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und Verdichtungsvariablen und nur relevant wenn dwValProcess = TLG_VAL_DLL.

szFuncStartEvent

Name der Aktion oder der DLL-Funktion, die das Startereignis spezifiziert.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen.

szDLLStartEvent

Name der DLL.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen.

szFuncStopEvent

Namen der Aktion oder der DLL-Funktion, die das Stoppereignis spezifiziert.

Der Parameter ist relevant für die zyklisch-selektive Archivierung von analogen und binären Variablen.

szDLLStopEvent

Name der DLL.

Der Parameter ist relevant für die zyklisch-selektive Archivierung von analogen und binären Variablen.

szFuncDynamic

Name der Aktion oder der DLL-Funktion, die die dynamische Umschaltung des Erfassungs- und/oder des Archivierungszyklus spezifiziert.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen,

szDLLDynamic

Name der DLL.

Der Parameter ist relevant für die zyklisch-selektive, zyklisch-kontinuierliche Archivierung von analogen und binären Variablen,

szUnitDirect

Kennzeichnet die Einheit der Archivvariablen.

Der Parameter ist relevant für analoge Variable aller Archivierungsarten.

szUnitStruct

Name der Struktur, die die Einheit der Archivvariablen enthält. NULL wenn die Einheit direkt in szUnitDirect projiziert wird.

Der Parameter ist relevant für analoge Variable aller Archivierungsarten,

dwRecltems

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

szSourceArchiv

Name des Quellarchivs.

Der Parameter ist relevant für die zyklisch-kontinuierliche Archivierung,

szSourceVarName

Name der Archivvariablen im Quellarchiv.

Der Parameter ist relevant für die zyklisch-kontinuierliche Archivierung,

szRawdataName

Name der Rohdatenvariablen.

Der Parameter ist relevant für prozessgesteuerte Variablen.

szRawConvDLLName

Name der Normierungs-DLL.

Der Parameter ist relevant für prozessgesteuerte Variablen,

dwRawDataIndex

Die Nummer der Rohdatenvariablen entspricht im Zusammenhang mit S7PMC der Archiv-ID der PLC.

Der Parameter ist relevant für prozessgesteuerte Variablen.

dwRawDataFormat

Format der Rohdatenvariablen.

Der Parameter ist relevant für prozessgesteuerte Variablen

ScaleStruct

Struktur TLG_SCAL_STR (Seite 826) mit den Grenzwerten der Variablen.

RecordStruct

Der Parameter ist für spätere Erweiterungen reserviert.

Benötigte Dateien

pde_def.h

API Funktionen

TLGReadVariable (Seite 864)	Variablen-Parameter ermitteln
-----------------------------	-------------------------------

Siehe auch

TLGReadVariable (Seite 864)

TLG_SCAL_STR (Seite 826)

2.8.2.19 TLG_VARIABLE_INFO (RT Professional)

Deklaration

```
typedef struct {  
    DWORD    dwVariableTyp;  
    TCHAR    szVariableName[ _MAX_PATH + 1 ];  
}  
TLG_VARIABLE_INFO;
```

Members

dwVariableTyp

dwVariableTyp kennzeichnet die Art der Variablen. Mögliche Werte sind:

TLG_VAR_TYP_ANALOG	analoge Variable
TLG_VAR_TYP_BIN	binäre Variable
TLG_VAR_TYP_COMPRESS	Variable für Verdichtungsarchive
TLG_VAR_TYP_PROCESS	prozessgesteuerte Variable

szVariableName

Name der Variablen

Benötigte Dateien

pdertdef.h

pde_typ.h

API Funktionen

TLG_ENUMVARIABLES (Seite 863)	Variablen auflisten (Callback)
-------------------------------	--------------------------------

Siehe auch

TLG_ENUMVARIABLES (Seite 863)

2.8.3 Allgemeine Funktionen (RT Professional)

2.8.3.1 TLGCSCConnect (RT Professional)

Beschreibung

Die Funktion baut eine Verbindung zur WinCC-Projektdatenbank auf.

Deklaration

```
BOOL TLGCSCConnect (
    HWND          hwndParent,
    LPCMN_ERROR   lpError );
```

Parameter

hwndParent

Fensterhandle des Vaterfensters.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung aufgebaut.

FALSE

Fehler

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLGCSCConnectEx (Seite 846)	Verbindung zur Projektdatenbank aufbauen
TLGCSDisConnect (Seite 847)	Verbindung zur Projektdatenbank aufbauen

Beispiele

Enum all acquisition and archiving times (Seite 919) "TL01.cpp"
Read parameters of time object (Seite 933) "TL01.cpp"
Enum archives (Seite 922) "TL01.cpp"
Read archive (Seite 929) "TL01.cpp"

Siehe auch

TLGCSConnectEx (Seite 846)
Enum archives (Seite 922)
TLGOpenProject (Seite 854)
Read parameters of time object (Seite 933)
Enum all acquisition and archiving times (Seite 919)
Read archive (Seite 929)
TLGCSDisConnect (Seite 847)

2.8.3.2 TLGCSConnectEx (RT Professional)

Beschreibung

Die Funktion baut eine Verbindung zur WinCC-Projektdatebank auf.

Deklaration

```
BOOL TLGCSConnectEx (  
    HWND          hwndParent,  
    DWORD         dwMode,  
    LPCMN_ERROR   lpError );
```

Parameter

hwndParent

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

dwMode

Der Parameter ist für spätere Erweiterungen reserviert.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung aufgebaut

FALSE

Fehler

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLGCSCConnect (Seite 845)	Verbindung zur Projektdatenbank aufbauen
---------------------------	--

Siehe auch

TLGCSCConnect (Seite 845)

TLGOpenProject (Seite 854)

2.8.3.3 TLGCSDisConnect (RT Professional)

Beschreibung

Die Funktion baut eine bestehende Verbindung zur WinCC-Projektdatenbank ab. Der Aufruf ist nötig, damit die DLL wieder sauber entladen werden kann.

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX...) benutzt werden, weil das auf Grund Microsoft spezifischer Mechanismen u. U. zum Hängenbleiben des Aufrufs und damit des Programms führen kann.

Deklaration

```
BOOL TLGCSDisConnect (  
    LPCMN_ERROR    lpError );
```

Parameter

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung abgebaut.

FALSE

Fehler

Benötigte Files

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLGCSCConnect (Seite 845)	Verbindung zur Projektdatenbank aufbauen
---------------------------	--

Beispiele

Enum all acquisition and archiving times (Seite 919) "TL01.cpp"

Read parameters of time object (Seite 933) "TL01.cpp"

Enum archives (Seite 922) "TL01.cpp"

Read archive (Seite 929) "TL01.cpp"

Siehe auch

TLGCSCConnect (Seite 845)

Enum all acquisition and archiving times (Seite 919)

Enum archives (Seite 922)

Read archive (Seite 929)

Read parameters of time object (Seite 933)

2.8.3.4 TLGChangeLanguage (RT Professional)

Beschreibung

Mit der Funktion kann die aktuell verwendete Datensprache umgeschaltet werden.

Deklaration

```
BOOL TLGChangeLanguage (
    DWORD          dwLanguage,
    PCMN_ERROR     lpError );
```

Parameter

dwLanguage

Code der zukünftig zu verwendenden Sprache.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Datensprache umgeschaltet

FALSE

Fehler

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertcli.lib

pdertcli.dll

2.8.3.5 TLGConnect (RT Professional)

Beschreibung

Die Funktion initialisiert das Archivsystem und baut eine Verbindung zum Tag Logging Runtime auf.

Deklaration

```
BOOL TLGConnect (
    HWND          hwndParent,
    PCMN_ERROR    lpError );
```

Parameter

hwndParent

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbesetzt werden.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung aufgebaut

FALSE

Fehler

Bemerkung

Der Aufruf erzeugt für die Kommunikation ein unsichtbares Fenster, das erst mit der Funktion TLGDisconnect gelöscht wird. Dies ist unter Umständen von Bedeutung, wenn auch Windows-Funktionen benutzt werden.

Fehlermeldungen

TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertcli.lib

pdertcli.dll

Verwandte Funktionen

TLGDisconnect (Seite 851)	Verbindung abbauen
---------------------------	--------------------

Beispiele

Enumerate archives (Seite 927) "TL02.cpp"

Siehe auch

Enumerate archives (Seite 927)

TLGDisconnect (Seite 851)

Edit curve template - Example 1 (Seite 916)

2.8.3.6 TLGDisconnect (RT Professional)

Beschreibung

Mit der Funktion wird eine bestehende Verbindung zum Tag Logging Runtime abgebaut.

Deklaration

```
BOOL TLGDisconnect (  
    PCMN_ERROR    lpError );
```

Parameter

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Verbindung abgebaut

FALSE

Fehler

Bemerkung

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX...) benutzt werden, weil das auf Grund Microsoft spezifischer Mechanismen gegebenenfalls zum Hängenbleiben des Aufrufs und damit des Programms führen kann.

Fehlermeldungen

TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertcli.lib

pdertcli.dll

Verwandte Funktionen

TLGConnect (Seite 850)	Verbindung aufbauen
------------------------	---------------------

Beispiele

Enumerate archives (Seite 927) "TL02.cpp"

Siehe auch

Enumerate archives (Seite 927)

TLGConnect (Seite 850)

Edit curve template - Example 1 (Seite 916)

Enum all acquisition and archiving times (Seite 919)

2.8.4 Projektverwaltungsfunktionen (RT Professional)

2.8.4.1 TLGCloseProject (RT Professional)

Beschreibung

Schließt das aktuelle Projekt. Alle zu diesem Projekt gehörenden Datenobjekte werden gelöscht.

Die Funktion hat keine Bedeutung mehr und liefert immer TRUE als Rückgabewert.

Deklaration

```
BOOL TLGCloseProject (
    HANDLE      hProject,
    LPCMN_ERROR lpoes );
```

Parameter

hProject

Handle eines mit TLGOpenProject geöffneten Projekts.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Projekt geschlossen

FALSE

Fehler

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Beispiele

Read parameters of time object (Seite 933) "TL01.cpp"

Enum archives (Seite 922) "TL01.cpp"

Read archive (Seite 929) "TL01.cpp"

Siehe auch

Read parameters of time object (Seite 933)

Enum archives (Seite 922)

Read archive (Seite 929)

2.8.4.2 TLGOpenProject (RT Professional)

Beschreibung

Baut eine Verbindung zur WinCC-Projektdatenbank auf.

Deklaration

```
BOOL TLGOpenProject (  
    HANDLE*      lphProject,  
    LPTSTR      lpszProjectName,  
    HWND        hwndParent,  
    LPCMN_ERROR lpmes );
```

Parameter

lphProject

Adresse eines Speicherbereichs, in dem das Handle gespeichert werden soll.

lpszProjectName

Name des zu öffnenden Projektes.

Der hier anzugebende Projektpfad kann mit einer der folgenden API-Funktionen ermittelt werden:

- DMEnumOpenedProjects
- DMGetRuntimeProject
- TLGEnumProject

Wenn ein anderer Projektpfad als das aktuell geöffnete Projekt angegeben wird, so wird der Fehler TLG_API_PROJECT_NAME_NOT_EXIST zurückgegeben. Ist kein Projekt geöffnet, wird der Fehler TLG_API_NO_PROJECT_EXIST zurückgegeben.

hwndParent

Handle des Vaterfensters.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Projekt geöffnet

FALSE

Fehler

Bemerkung

Wenn nicht zuvor mittels TLGCSCConnect eine Verbindung zur WinCC-Projektdatenbank aufgebaut wurde, wird der Verbindungsfehler TLG_API_NO_CONNECTION zurückgegeben.

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_NO_PROJECT_EXIST	kein Projekt vorhanden
TLG_API_PROJECT_NAME_NOT_EXIST	Projektname nicht gefunden

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLGCSCConnect (Seite 845)	Verbindung zur Projektdatenbank aufbauen
TLGCSCConnectEx (Seite 846)	Verbindung zur Projektdatenbank aufbauen
TLGEnumProject (Seite 856)	Projekte auflisten

Beispiele

Read parameters of time object (Seite 933) "TL01.cpp"

Enum archives (Seite 922) "TL01.cpp"

Read archive (Seite 929) "TL01.cpp"

Enum variables of an archive (Seite 925) "TL02.cpp"

Siehe auch

TLGEnumProject (Seite 856)

TLGCSCConnect (Seite 845)

TLGCSCConnectEx (Seite 846)

Read parameters of time object (Seite 933)

Enum archives (Seite 922)

Read archive (Seite 929)

Enum variables of an archive (Seite 925)

2.8.4.3 TLGEnumProject (RT Professional)

Beschreibung

Die Funktion ermittelt die Namen der geöffneten Projekte.

Deklaration

```
BOOL TLGEnumProject (
    TLG_ENUM_PROJECT_NAME_CALLBACK    lpCallbackFunc,
    PVOID                               lpUser,
    LPCMN_ERROR                        lpoes );
```

Parameter

lpCallbackFunc

Zeiger auf Ihre Callback-Funktion, die für jedes geöffnete Projekt aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Projekte aufgelistet

FALSE

Fehler

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLG_ENUM_PROJECT_NAME_CALLBACK (Seite 857)	Projekte auflisten (Callback)
--	-------------------------------

Siehe auch

TLG_ENUM_PROJECT_NAME_CALLBACK (Seite 857)

TLGOpenProject (Seite 854)

2.8.4.4 TLG_ENUM_PROJECT_NAME_CALLBACK (RT Professional)**Beschreibung**

Um die vom System aufgelisteten Projekte auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUM_PROJECT_NAME_CALLBACK bereitstellen.

Deklaration

```

BOOL ( * TLG_ENUM_PROJECT_NAME_CALLBACK) (
    LPTSTR    lpszName,
    PVOID     lpUser );

```

Parameter**lpszName**

Der Zeiger lpszName verweist auf den Namen des ersten Projektes.

IpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

Die Rückgabewerte sind abhängig von Ihrer Implementierung.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

pdecsccli.h

Verwandte Funktionen

TLGEnumProject (Seite 856)	Projekte auflisten
----------------------------	--------------------

Siehe auch

TLGEnumProject (Seite 856)

2.8.5 Funktionen zum Bearbeiten von Variablen (RT Professional)

2.8.5.1 TLGEnumVariables (RT Professional)

Beschreibung

Dier Funktion ermittelt alle Variablennamen eines Archives.

Deklaration

```
BOOL TLGEnumVariables (
    HANDLE                hProject,
    LPTSTR                lpszArchivName,
    TLG_ENUM_VARIABLE_NAME_CALLBACK lpCallbackFunc,
    PVOID                lpUser,
    LPCMN_ERROR           lpoes );
```

Parameter

hProject

Handle des Projekts, in dem das Archiv liegt.

lpszArchivName

Zeiger auf den Namen des Archivs

lpCallbackFunc

Zeiger auf Ihre Callback-Funktion, die für jeden Variableneintrag in dem Archiv aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archivvariablen aufgelistet

FALSE

Fehler

Benötigte Dateien

pdecscli.h

pdecscli.lib

pdecscli.dll

Verwandte Funktionen

TLG_ENUM_VARIABLE_NAME_CALLBACK (Seite 860)	Variablen auflisten (Callback)
--	--------------------------------

Siehe auch

TLG_ENUM_VARIABLE_NAME_CALLBACK (Seite 860)

TLGReadVariable (Seite 864)

2.8.5.2 TLG_ENUM_VARIABLE_NAME_CALLBACK (RT Professional)

Beschreibung

Um die vom System aufgelisteten Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUM_VARIABLE_NAME_CALLBACK bereitstellen.

Deklaration

```
BOOL ( * TLG_ENUM_VARIABLE_NAME_CALLBACK ) (  
    LPTSTR    lpszName,  
    PVOID     lpUser );
```

Parameter

lpszName

Der Zeiger lpszName verweist auf den Namen der ersten Variablen.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Benötigte Dateien

pdecsccli.h

Verwandte Funktionen

TLGEnumVariables (Seite 858)	Variablen auflisten
------------------------------	---------------------

Siehe auch

TLGEnumVariables (Seite 858)

2.8.5.3 TLGEnumVariablesEx (RT Professional)**Beschreibung**

Dier Funktion ermittelt alle Variablennamen eines Archives.

Deklaration

```

BOOL TLGEnumVariablesEx (
    LPCTSTR          lpszArchiveName,
    TLG_ENUMVARIABLES lpfnCallback,
    LPVOID           lpUser,
    PCMN_ERROR       lpError );

```

Parameter**lpszArchiveName**

Zeiger auf den Namen des Archivs, dessen Variablen aufgelistet werden sollen.

lpfnCallback

Zeiger auf Ihre Callback-Funktion, die für jede Variable des Archivs aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablen aufgelistet

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt

Benötigte Dateien

- pdecsccli.h
- pdecsccli.lib
- pdecsccli.dll
- pdertcli.h
- pdertcli.lib
- pdertcli.dll

Verwandte Funktionen

TLG_ENUMVARIABLES (Seite 863)	Variablen auflisten (Callback)
-------------------------------	--------------------------------

Beispiele

Enum variables of an archive (Seite 925) "TL02.cpp"

Siehe auch

TLG_ENUMVARIABLES (Seite 863)
Enum variables of an archive (Seite 925)

2.8.5.4 TLG_ENUMVARIABLES (RT Professional)**Beschreibung**

Um die vom System aufgelisteten Daten einer Variablen auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUMVARIABLES bereitstellen.

Deklaration

```
BOOL ( * TLG_ENUMVARIABLES) (  
    PTLG_VARIABLE_INFO    lpvi,  
    LPVOID                lpUser );
```

Parameter**lpvi**

Adresse einer Struktur vom Typ TLG_VARIABLE_INFO (Seite 844) mit den Daten einer Variablen.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert**TRUE**

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

pdertcli.h
pde_glob.h
pdertdef.h

Verwandte Funktionen

TLGEnumVariablesEx (Seite 861)	Variablen auflisten
--------------------------------	---------------------

Beispiele

Enum variables of an archive (Seite 925) "TL02.cpp"

Siehe auch

TLGEnumVariablesEx (Seite 861)
TLG_VARIABLE_INFO (Seite 844)
Enum variables of an archive (Seite 925)
TLGEnumArchivsEx (Seite 870)

2.8.5.5 TLGReadVariable (RT Professional)

Beschreibung

Lesen der Parameter einer Variablen.

Deklaration

```
BOOL TLGReadVariable (  
    HANDLE          hProject,  
    LPTSTR          lpszArchivName,  
    LPTSTR          lpszVariableName,  
    PTLG_VAR_STR    lpVariable,  
    LPCMN_ERROR     lpoes );
```

Parameter

hProject

Handle des Projekts, in dem die zu bearbeitende Variable liegt.

lpszArchivName

Zeiger auf den Namen des Archivs

IpszVariableName

Zeiger auf den Namen einer Variablen des Archives.

IpVariable

Adresse der Struktur TLG_VAR_STR (Seite 838) , aus der die Daten der Variablen ausgelesen werden.

Ipoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Parameter gelesen

FALSE

Fehler

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLGEnumVariables (Seite 858)	Variablen auflisten
TLGReadVariable6	Variablen-Parameter ermitteln

Beispiele

Read archive (Seite 929) "TL01.cpp"

Siehe auch

TLG_VAR_STR (Seite 838)

TLGEnumVariables (Seite 858)

Read archive (Seite 929)

2.8.6 Funktionen zum Bearbeiten von Archiven (RT Professional)

2.8.6.1 TLGEnumArchives (RT Professional)

Beschreibung

Die Funktion ermittelt die Namen aller Archive im Projekt hproject. Mit dieser Funktion können alle Arten von Archiven bearbeitet werden.

Deklaration

```
BOOL TLGEnumArchives (  
    HANDLE                hProject,  
    TLG_ENUM_ARCHIV_CALLBACK lpCallbackFunc,  
    PVOID                lpUser,  
    LPCMN_ERROR           lpoes );
```

Parameter

hProject

Handle des Projekts, in dem die Archive liegen.

lpCallbackFunc

Zeiger auf Ihre Callback-Funktion, die für jedes Archiv aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archive aufgelistet

FALSE

Fehler

Benötigte Dateien

pdeccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLGReadArchiv (Seite 893)	Archivparameter lesen
TLG_ENUM_ARCHIV_CALLBACK (Seite 867)	Archive auflisten (Callback)

Beispiele

Enum archives (Seite 922) "TL01.cpp"

Siehe auch

TLG_ENUM_ARCHIV_CALLBACK (Seite 867)

Enum archives (Seite 922)

TLGReadArchiv (Seite 893)

2.8.6.2 TLG_ENUM_ARCHIV_CALLBACK (RT Professional)

Beschreibung

Um die vom System aufgelisteten Archive auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUM_ARCHIV_CALLBACK bereitstellen.

Deklaration

```

BOOL ( * TLG_ENUM_ARCHIV_CALLBACK) (
    LPTSTR    lpszName,
    PVOID     lpUser );

```

Parameter

lpszName

Der Zeiger lpszName verweist auf den Namen des Archives.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

pdecsccli.h

Verwandte Funktionen

TLGEnumArchives (Seite 866)	Archive auflisten
---	-----------------------------------

Beispiele

[Enum archives \(Seite 922\) "TL01.cpp"](#)

Siehe auch

[TLGEnumArchives \(Seite 866\)](#)

[Enum archives \(Seite 922\)](#)

2.8.6.3 TLGEnumArchivs (RT Professional)

Beschreibung

Enumeriert die vorhandenen Archive. Im Unterschied zu TLGEnumArchivsSel können die aufzulistenden Archive nur über den Archivtyp eingeschränkt werden.

Über die Callback-Funktion werden die Informationen eines Archivs in Strukturen vom Typ TLG_TABLE_INFO bereitgestellt. Dabei wird der Archivname als Tabellename, d.h. in der Form "UA#ARCHIV#ArchivName" bzw. "PDE#HD#Archivname#Variablenname", übergeben.

Deklaration

```

BOOL TLGEnumArchivs (
    DWORD          dwArchivTyp,
    TLG_ENUMTABLES lpfnCallback,
    LPVOID         lpUser,
    PCMN_ERROR     lpError );

```

Parameter

dwArchivTyp

dwArchivTyp kennzeichnet den Archivtyp:

TLG_ARCTYP_USER	Anwenderarchiv
TLG_ARCTYP_PROCESS	Prozesswertarchiv
TLG_ARCTYP_COMPRESS	Verdichtungsarchiv
NULL	Es werden alle Archive enumeriert.

lpfnCallback

Zeiger auf Ihre Callback-Funktion, die für jedes Archiv aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archive aufgelistet

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten

Benötigte Dateien

pdertcli.h
pde_glob.h
pdertdef.h
pdertcli.lib
pdertcli.dll

Verwandte Funktionen

TLGEnumArchivsEx (Seite 870)	Archive auflisten
TLGEnumArchivsSel (Seite 872)	Archive auflisten
TLG_ENUMTABLES (Seite 874)	Archive auflisten (Callback)

Beispiele

Enumerate archives (Seite 927) "TL02.cpp"

Siehe auch

TLGEnumArchivsEx (Seite 870)
TLGEnumArchivsSel (Seite 872)
TLG_ENUMTABLES (Seite 874)
Enumerate archives (Seite 927)

2.8.6.4 TLGEnumArchivsEx (RT Professional)

Beschreibung

Enumeriert die vorhandenen Archive. Im Unterschied zu TLGEnumArchivsSel können die aufzulistenden Archive nur über den Archivtyp (Anwender-, Prozesswert-, Verdichtungsarchiv) eingeschränkt werden.

Über die Callback-Funktion werden die Informationen eines Archivs in Strukturen vom Typ TLG_TABLE_INFO bereitgestellt. Im Unterschied zu TLGEnumArchivs wird nicht der Tabellenname des Archivs, sondern der "reine" Archivname übergeben.

Deklaration

```

BOOL TLGEnumArchivsEx (
    DWORD          dwArchivTyp,
    TLG_ENUMTABLES lpfnCallback,
    LPVOID         lpUser,
    PCMN_ERROR     lpError );

```

Parameter

dwArchivTyp

dwArchivTyp kennzeichnet den Archivtyp:

TLG_ARCTYP_U_USER	Anwenderarchiv
TLG_ARCTYP_PROCESS	Prozesswertarchiv
TLG_ARCTYP_COMPRESS	Verdichtungsarchiv
NULL	Es werden alle Archive enumeriert.

lpfnCallback

Zeiger auf Ihre Callback-Funktion, die für jedes Archiv aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archive werden aufgelistet

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung

TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt

Benötigte Dateien

pdertcli.h
pde_glob.h
pdertdef.h
pdertcli.lib
pdertcli.dll

Verwandte Funktionen

TLG_ENUMTABLES (Seite 874)	Archive auflisten (Callback)
TLGEnumArchivs (Seite 868)	Archive auflisten
TLGEnumArchivsSel (Seite 872)	Archive auflisten

Siehe auch

TLGEnumArchivsSel (Seite 872)
TLGEnumArchivs (Seite 868)
TLG_ENUMTABLES (Seite 874)

2.8.6.5 TLGEnumArchivsSel (RT Professional)

Beschreibung

Die Funktion enumeriert die vorhandenen Archivtypen. Im Unterschied zu TLGEnumArchivs können die aufzulistenden Archive neben dem Archivtyp auch über die Archivierungsart (Umlaufarchiv, Folgearchiv) eingeschränkt werden.

Über die Callback-Funktion werden die Informationen eines Archivs in Strukturen vom Typ TLG_TABLE_INFO bereitgestellt. Dabei wird der Archivname als Tabellename übergeben, d.h. in der Form "UA#ARCHIV#ArchivName" bzw. "PDE#HD#Archivname#Variablenname". Es wird über alle Variablen der Archive enumeriert. Diese Form wird benötigt z.B. bei TLGBackup.

Deklaration

```

BOOL TLGEnumArchivsSel (
    DWORD          dwArchivTyp,
    DWORD          dwSaveTyp,
    TLG_ENUMTABLES lpfnCallback,
    LPVOID         lpUser,
    PCMN_ERROR     lpError );

```

Parameter

dwArchivTyp

dwArchivTyp kennzeichnet den Archivtyp:

TLG_ARCTYP_USER	Anwenderarchiv
TLG_ARCTYP_PROCESS	Prozesswertarchiv
TLG_ARCTYP_COMPRESS	Verdichtungsarchiv
NULL	Es werden alle Archive enumeriert.

dwSaveTyp

dwSaveTyp kennzeichnet die Art des Archives:

TLG_ARCTYP_ALL	Umlauf- und Folgearchive
TLG_ARCTYP_CIRCULAR	Umlaufarchive
TLG_ARCTYP_FOLLOW	Folgearchive

lpfnCallback

Zeiger auf Ihre Callback-Funktion, die für jedes Archiv aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Archive werden aufgelistet.

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige oder falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt

Benötigte Dateien

- pdertcli.h
- pde_glob.h
- pdertdef.h
- pdertcli.lib
- pdertcli.dll

Verwandte Funktionen

TLG_ENUMTABLES (Seite 874)	Archive auflisten (Callback)
AUTOHOTPOT	Archive auflisten

Siehe auch

- TLGEnumArchivsEx (Seite 870)
- TLGEnumArchivs (Seite 868)
- TLG_ENUMTABLES (Seite 874)
- TLG_TABLE_INFO (Seite 827)

2.8.6.6 TLG_ENUMTABLES (RT Professional)

Beschreibung

Um die vom System aufgelisteten Archive auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUMTABLES bereitstellen.

Deklaration

```
BOOL TLG_ENUMTABLES (
    LPTSTR lpTableName,
    PTLG_TABLE_INFO lpti,
    PVOID lpUser );
```

Parameter

IpTableName

Zeiger auf den Namen des Archives.

lpti

Informationen zur Datenbanktabelle sind in der Struktur TLG_TABLE_INFO (Seite 827) enthalten.

IpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

pdertcli.lib

pdertcli.dll

Verwandte Funktionen

TLGEnumArchivs (Seite 868)	Archive auflisten
TLGEnumArchivsSel (Seite 872)	Archive auflisten

Beispiele

Enumerate archives "TL02.cpp"

Siehe auch

TLGEnumArchivs (Seite 868)

TLGEnumArchivsSel (Seite 872)

TLG_TABLE_INFO (Seite 827)

TLGEnumArchivsEx (Seite 870)

2.8.6.7 TLGFreeMemory (RT Professional)

Beschreibung

Mit der Funktion wird ein durch TLGGetArchivDataEx allozierter Speicherbereich freigegeben.

Deklaration

```
BOOL TLGFreeMemory (  
    LPVOID lpMemory );
```

Parameter

lpMemory

Den Zeiger auf den freizugebenden Speicherbereich erhalten Sie beim Aufruf der Funktion TLGGetArchivDataEx im Parameter ppTLGData.

Hinweis

Der Zeiger muss vor dem Aufruf TLGGetArchivDataEx() mit NULL initialisiert, und vor Aufruf TLGFreeMemory() auf ungleich NULL geprüft werden.

Rückgabewert

TRUE

Speicher wird freigegeben

FALSE

Fehler

Hinweis

Der Zeiger muss vor dem Aufruf TLGGetArchivDataEx() mit NULL initialisiert, und vor Aufruf TLGFreeMemory() auf ungleich NULL geprüft werden.

Benötigte Dateien

pdertcli.h

pdertcli.lib

pdertcli.dll

Verwandte Funktionen

TLGGetArchivDataEx (Seite 881)	Daten eines Archives auslesen
--------------------------------	-------------------------------

Siehe auch

TLGGetArchivDataEx (Seite 881)

2.8.6.8 TLGGetArchivData (RT Professional)**Beschreibung**

Die Funktion liest die Daten zwischen zwei Zeitpunkten aus einem Archiv aus. Im Unterschied zu TLGGetArchivDataEx werden die Werte der Archivvariablen über eine Callback-Funktion zur Verfügung gestellt.

Mit einem Aufruf können maximal 10.000 Datensätze gelesen werden. Liegen zwischen Startzeit und Endzeit mehr Datensätze, so erhalten Sie nur die ersten 10.000.

Die folgenden Datensätze erhalten Sie, indem Sie zu der Uhrzeit des letzten Datensatzes eine Millisekunde addieren und dies als Startzeit in einem neuen Aufruf von TLGGetArchivData verwenden.

Deklaration

```
BOOL TLGGetArchivData (
    LPTSTR                lpszArchivName,
    LPTSTR                lpszVarName,
    SYSTEMTIME            stStart,
    SYSTEMTIME            stStop,
    TLG_GETARCHIVDATA_CALLBACK lpfnCallback,
    PVOID                lpUser,
    DWORD                dwFlags,
    PCMN_ERROR            lpError );
```

Parameter

lpszArchivName

Zeiger auf den Namen des Archivs, aus dem Daten gelesen werden sollen.

lpszVarName

Zeiger auf den Namen der Archivvariablen, deren Werte gelesen werden sollen.

stStart

Startzeitpunkt, von dem ab Daten gelesen werden sollen.

stStop

Endzeitpunkt, bis zu dem Daten gelesen werden sollen.

lpfnCallback

Zeiger auf Ihre Callback-Funktion, die für jeden auszulesenden Messpunkt aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archivdaten ermittelt

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NODEFAULTSERVER	Kein Standardserver projiziert
TLG_API_NOSERVER	Kein Standardserver projiziert und kein lokaler Server verfügbar

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

pdertcli.lib

pdertcli.dll

Verwandte Funktionen

TLG_GETARCHIVDATA_CALLBACK (Seite 879)	Daten eines Archives auslesen (Callback)
TLGGetArchivDataEx (Seite 881)	Daten eines Archives auslesen

Siehe auch

TLG_GETARCHIVDATA_CALLBACK (Seite 879)

TLGGetArchivDataEx (Seite 881)

2.8.6.9 TLG_GETARCHIVDATA_CALLBACK (RT Professional)**Beschreibung**

Um die vom System aufgelisteten Archivdaten auszuwerten, müssen Sie eine Callback-Funktion vom Typ TLG_GETARCHIVDATA_CALLBACK bereitstellen.

Deklaration

```
BOOL ( * TLG_GETARCHIVDATA_CALLBACK) (  
    PTLG_GETARCHIVDATA lpGAD,  
    PVOID lpUser );
```

Parameter

lpGAD

Pro Archiv reserviert das System temporär Speicher der Länge der Struktur TLG_GETARCHIVDATA. Der Zeiger lpGAD verweist auf den Beginn des ersten Elementes.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Möglichst nur Daten umkopieren. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z. B. GetMessage
 - Runtime API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

Verwandte Funktionen

TLGGetArchivData (Seite 877)	Daten eines Archives auslesen
TLG_GETARCHIVDATA (Seite 822)	Archivdaten von Prozesswertarchiven (Struktur) Archivdaten von Verdichtungsarchiven (Struktur)

Siehe auch

TLGGetArchivData (Seite 877)

TLG_GETARCHIVDATA (Seite 822)

2.8.6.10 TLGGetArchivDataEx (RT Professional)**Beschreibung**

Liest die Daten zwischen zwei Zeitpunkten aus einem Archiv aus. Im Unterschied zu TLGGetArchivData werden die Werte der Archivvariablen im Speicher hinterlegt.

Hinweis

Der zugewiesene Speicher in ppTlgData muss über TLGFreeMemory wieder freigegeben werden.

Deklaration

```

BOOL TLGGetArchivDataEx (
    LPCTSTR                lpszArchivName,
    LPCTSTR                lpszVarName,
    SYSTEMTIME*            pstStart,
    SYSTEMTIME*            pstStop,
    PTLG_ARCHIVDATARAW*   ppTlgData,
    DWORD*                 pdwNumberOfData,
    DWORD*                 pdwFlags,
    PCMN_ERROR             lpError );

```

Parameter**lpszArchivName**

Zeiger auf den Archivnamen, aus dem die Daten gelesen werden.

lpszVarName

Zeiger auf den Namen der Archivvariablen, deren Werte gelesen werden.

stStart

Startzeitpunkt, von dem ab Daten gelesen werden.

stStop

Endzeitpunkt, bis zu dem Daten gelesen werden.

Wenn eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt wird, verwenden Sie die Funktion GetLocalTime und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

ppTlgData

Adresse eines Zeiger, in dem die Adresse der Daten der Archivvariablen hinterlegt werden. Das Datenfeld wird von TLGGetArchivDataEx allokiert und enthält Strukturen vom Typ TLG_ARCHIVDATARAW (Seite 808).

Hinweis

Der zugewiesene Speicher muss über TLGFreeMemory wieder freigegeben werden.

Der Zeiger muss vor dem Aufruf TLGGetArchivDataEx() mit NULL initialisiert, und vor Aufruf TLGFreeMemory() auf ungleich NULL geprüft werden.

pdwNumberOfData

Zeiger auf Anzahl der Datensätze.

- Vor dem Aufruf: Maximale Anzahl der zu lesenden Datensätze.
- Nach dem Aufruf: Anzahl der gelesenen Datensätze mit den Werten der Archivvariablen (der Strukturen vom Typ TLG_ARCHIVDATARAW).

Maximal 10000 Datensätze können mit einem Aufruf gelesen werden. Gibt es zwischen Startzeit und Endzeit mehr Datensätze, so erhalten Sie nur die ersten 10000. Die nächsten bekommen Sie, indem Sie zur Zeit des letzten Datensatzes eine Millisekunde addieren, und dies als Startzeit in einem neuen Aufruf von TLGGetArchivDataEx verwenden.

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archivdaten ermittelt

FALSE

Fehler

Hinweis

Der durch TLGGetArchivDataEx zugewiesene Speicher in ppTlgData muss über TLGFreeMemory wieder freigegeben werden.

Der Zeiger muss vor dem Aufruf TLGGetArchivDataEx() mit NULL initialisiert, und vor Aufruf TLGFreeMemory() auf ungleich NULL geprüft werden.

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NODEFAULTSERVER	Kein Standardserver projiziert
TLG_API_NOSERVER	Kein Standardserver projiziert und kein lokaler Server verfügbar

Benötigte Dateien

pdertcli.h
 pde_glob.h
 pdertdef.h
 pdertcli.lib
 pdertcli.dll

Verwandte Funktionen

TLGGetArchivData (Seite 877)	Daten eines Archives auslesen
TLGFreeMemory (Seite 876)	Speicher freigeben

Siehe auch

TLGFreeMemory (Seite 876)
 TLGGetArchivData (Seite 877)
 TLG_ARCHIVDATARAW (Seite 808)

2.8.6.11 TLGGetClosestTime (RT Professional)**Beschreibung**

Die Funktion ermittelt die zu einem vorgegebenen Zeitpunkt benachbarten Archivierungszeitpunkte.

Deklaration

```
BOOL TLGGetClosestTime (
    LPCTSTR      lpszArchivName,
    LPCTSTR      lpszVarName,
    SYSTEMTIME*  pstTime,
    BOOL         bPrevious,
    PCMN_ERROR   lpError );
```

Parameter

lpszArchivName

Zeiger auf den Namen des Archives

lpszVarName

Zeiger auf den Namen der Variablen

pstTime

pstTime wird als Ein- und Ausgabeparameter verwendet. Beim Aufruf von TLGGetClosestTime enthält pstTime die Systemzeit zu der die benachbarten Archivierungszeiten ermittelt werden sollen. Nach dem erfolgreichen Funktionsaufruf enthält pstTime den nächstgelegenen Archivierungszeitpunkt.

bPrevious

bPrevious kennzeichnet, ob zur übergebenen Systemzeit ein vorheriger oder ein nachfolgender Archivierungszeitpunkt ermittelt werden soll.

TRUE	Ein vorheriger Archivierungszeitpunkt wird ermittelt.
FALSE	Ein nachfolgender Archivierungszeitpunkt wird ermittelt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

pstTime enthält einen benachbarten Archivierungszeitpunkt.

FALSE

Fehler oder es konnte kein benachbarter Archivierungszeitpunkt ermittelt werden.

Bemerkung

Wenn bPrevious als TRUE angegeben ist und pstTime einen vorhandenen Wert enthält, kann es sein, dass genau dieser Zeitpunkt zurückgegeben wird, und nicht der davor liegende Zeitpunkt. Wenn bPrevious als FALSE angegeben ist, wird immer der nachfolgende Zeitpunkt geliefert.

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_VARIABLE_NAME_NOT_EXIST	keine Variable mit diesem Namen gefunden
TLG_API_TIME_NAME_NOT_EXIST	kein Zeitobjekt mit diesem Namen gefunden
TLG_API_NO_TIME_EXIST	es existiert kein Zeitobjekt

Benötigte Dateien

pdertcli.h
 pde_glob.h
 pdertcli.lib
 pdertcli.dll

Verwandte Funktionen

TLGGetClosestTimeEx (Seite 885)	Archivierungszeitpunkt ermitteln
---------------------------------	----------------------------------

Siehe auch

TLGGetClosestTimeEx (Seite 885)

2.8.6.12 TLGGetClosestTimeEx (RT Professional)**Beschreibung**

Die Funktion ermittelt die zu einem vorgegebenen Zeitpunkt benachbarten Archivierungszeitpunkte.

Deklaration

```
BOOL TLGGetClosestTimeEx (  
    LPCTSTR        lpszArchivName,  
    LPCTSTR        lpszVarName,  
    SYSTEMTIME*    pstTime,  
    BOOL           bPrevious,  
    BOOL           bIgnoreInvalid,  
    PCMN_ERROR     lpError );
```

Parameter

lpszArchivName

Zeiger auf den Namen des Archives

lpszVarName

Zeiger auf den Namen der Variablen

pstTime

pstTime wird als Ein- und Ausgabeparameter verwendet. Beim Aufruf von TLGGetClosestTime enthält pstTime die Systemzeit zu der die benachbarten Archivierungszeiten ermittelt werden. Nach dem erfolgreichen Funktionsaufruf enthält pstTime den nächstgelegenen Archivierungszeitpunkt.

bPrevious

bPrevious kennzeichnet, ob zur übergebenen Systemzeit ein vorheriger oder ein nachfolgender Archivierungszeitpunkt ermittelt werden soll.

TRUE	Ein vorheriger Archivierungszeitpunkt wird ermittelt.
FALSE	Ein nachfolgender Archivierungszeitpunkt wird ermittelt.

bIgnoreInvalid

Gibt an, ob Werte mit dem Flag Verbindungsstörung bei der Zeitermittlung ignoriert werden sollen.

TRUE	Die Werte mit Verbindungsstörung werden ignoriert und übersprungen.
FALSE	Alle Werte werden zur Zeitermittlung herangezogen, womit die selbe Funktion wie bei TLGGetClosestTime() vorliegt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

pstTime enthält einen benachbarten Archivierungszeitpunkt.

FALSE

Fehler oder es konnte kein benachbarter Archivierungszeitpunkt ermittelt werden.

Bemerkung

Wenn bPrevious als TRUE angegeben ist und pstTime einen vorhandenen Wert enthält, kann es sein, dass genau dieser Zeitpunkt zurückgegeben wird, und nicht der davor liegende Zeitpunkt. Wenn bPrevious als FALSE angegeben ist, wird immer der nachfolgende Zeitpunkt geliefert.

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_VARIABLE_NAME_NOT_EXIST	keine Variable mit diesem Namen gefunden
TLG_API_TIME_NAME_NOT_EXIST	kein Zeitobjekt mit diesem Namen gefunden
TLG_API_NO_TIME_EXIST	es existiert kein Zeitobjekt

Benötigte Dateien

pdertcli.h
 pde_glob.h
 pdertcli.lib
 pdertcli.dll

Verwandte Funktionen

TLGGetClosestTime (Seite 883)	Archivierungszeitpunkt ermitteln
-------------------------------	----------------------------------

Siehe auch

TLGGetClosestTime (Seite 883)

2.8.6.13 TLGInsertArchivData (RT Professional)**Beschreibung**

Die Funktion bietet die Möglichkeit, beliebige Daten in ein bestehendes Archiv auf der Festplatte einzufügen.

Wenn Sie TlgInsertArchivData für Werte durchführen, die während der Sommerzeit erstellt wurden, dann müssen Sie die Zeitverschiebung berücksichtigen. Setzen Sie beim Aufruf dieser Werte das Flag PDE_RT_DAYLIGHT.

Ohne dieses Flag liefern lesende Aufrufe, z.B. TLGGetArchivData oder TLGGetClosestTime, die Standardzeit zurück.

Hinweis

Da keine Überprüfung der Richtigkeit der einzufügenden Daten vorgenommen wird, besteht die Gefahr der Zerstörung des Archivs.

Deklaration

```
BOOL TlgInsertArchivData (
    LPCTSTR          lpszArchivName,
    LPCTSTR          lpszVarName,
    PTLG_ARCHIVDATARAW pTlgData,
    DWORD            dwNumberOfData,
    DWORD            dwFlags,
    PCMN_ERROR       lpError );
```

Parameter

lpszArchivName

Zeiger auf den Namen des Archives

lpszVarName

Zeiger auf den Namen der Variablen

pTlgData

Zeiger auf die erste der Strukturen vom Typ TLG_ARCHIVDATARAW mit den in das Archiv einzufügenden Variablenwerten.

dwNumberOfData

Anzahl der einzufügenden Variablenwerte (der Strukturen TLG_ARCHIVDATARAW).

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden. Hier kann das Flag TLG_API_FLG_FAST_INSERT angegeben werden.

Wenn dieses Flag gesetzt ist, werden die Daten wesentlich schneller im optimierten Modus geschrieben.

Voraussetzung zur Nutzung dieser Performanceoptimierung ist jedoch, dass die Zeitstempel der geschriebenen Datensätze zeitfolgerichtig sortiert sind, und außerdem stets aktueller sind als die Datensätze, die sich bereits im Archiv befinden.

Beim Einfügen von Datensätzen älteren Datums zwischen bereits bestehende Datensätze darf dieses Flag nicht verwendet werden. Werden Werte eingefügt, die älter sind als bestehende, oder die nicht zeitfolgerichtig sortiert sind, so werden die älteren Werte ohne Fehleranzeige verworfen.

IpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variablenwerte ins Archiv eingefügt.

FALSE

Fehler

Bemerkung

Wird ein Archiv unter IpszArchivName angegeben, das im Hauptspeicher liegt, so kommt es zur Fehlermeldung TLG_API_ERR_SUPPLY.

Mit dieser Funktion können nur Archive auf der Festplatte angesprochen werden. Wenn Schreibfehler bei einzelnen Werten aufgetreten sind, wird eine genauere Fehlerbeschreibung in CMN_ERROR zurückgegeben. Aufgrund der begrenzten Länge des CMN_ERROR.szErrorText ist die Anzahl der Detail-Fehler auf max. 57 begrenzt.

dwError1	TLG_API_ERR_SUPPLY
dwError2	0
dwError3	Zeilennummer im Quellcode
dwError4	Anzahl Fehler (max. 57)
dwError5	Anzahl der zu schreibenden Daten (dwNumberOfData)
szErrorText	TLG-API: ein Fehler ist aufgetreten, line xxx; Pos:0=0xXX,1=0xXX,2=0xXX, ... (, 56=0xXX)

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NODEFAULTSERVER	Kein Standardserver projiziert
TLG_API_NOSERVER	Kein Standardserver projiziert und kein lokaler Server verfügbar

Benötigte Dateien

- pdertcli.h
- pde_glob.h
- pdertdef.h
- pdertcli.lib
- pdertcli.dll

2.8.6.14 TLGLockArchiv (RT Professional)

Beschreibung

Sperrt ein komplettes Archiv oder gibt es frei. Wenn ein Archiv gesperrt ist, werden neu anfallende Daten nicht archiviert.

Deklaration

```
BOOL TLGLockArchiv (  
    HWND          hwnd,  
    LPTSTR        lpszArchivName,  
    BOOL          fLocked,  
    PCMN_ERROR    lpError );
```

Parameter

hwnd

Fensterhandle des PDE-Runtime-Fensters.

lpszArchivName

Zeiger auf den Namen des Archivs.

fLocked

TRUE:	Archiv sperren
FALSE:	Archiv freigeben

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archiv gesperrt/freigegeben

FALSE

Fehler

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NO_ARCHIV_EXIST	kein Archiv verfügbar
TLG_API_NODEFAULTSERVER	Kein Standardserver projektiert
TLG_API_NOSERVER	Kein Standardserver projektiert und kein lokaler Server verfügbar

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertcli.lib

pdertcli.dll

2.8.6.15 TLGLockVariable (RT Professional)**Beschreibung**

Sperrt eine Variable bzw. gibt sie frei. Eine gesperrte Variable wird nicht mehr aktualisiert und nicht mehr archiviert.

Deklaration

```

BOOL TLGLockVariable (
    HWND          hwnd,
    LPTSTR        lpszArchivName,
    LPTSTR        lpszVarName,
    BOOL          fLocked,
    PCMN_ERROR    lpError );

```

Parameter**hwnd**

Fensterhandle des PDE-Runtime-Fensters.

lpszArchivName

Zeiger auf den Namen des Archivs

Zur Auswahl eines bestimmten Servers kann dem Archivnamen "ServerPräfix:." vorangestellt werden werden.

lpszVarName

Zeiger auf den Namen der Variablen

fLocked

TRUE:	Variable sperren
FALSE:	Variable freigeben

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Variable freigegeben / gesperrt

FALSE

Fehler

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NO_ARCHIV_EXIST	kein Archiv verfügbar
TLG_API_NODEFAULTSERVER	Kein Standardserver projiziert
TLG_API_NOSERVER	Kein Standardserver projiziert und kein lokaler Server verfügbar

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertcli.lib

pdertcli.dll

2.8.6.16 TLGReadArchiv (RT Professional)

Beschreibung

Lesen der Parameter eines existierenden Archives im Projekt hProject. Mit dieser Funktion können alle Arten von Archiven bearbeitet werden.

Deklaration

```
BOOL TLGReadArchiv (  
    HANDLE          hProject,  
    LPTSTR         lpszArchivName,  
    PTLG_ARCHIV_STR lpArchiv,  
    LPCMN_ERROR    lpoes );
```

Parameter

hProject

Handle des Projekts, in dem das zu bearbeitende Archiv liegt.

lpszArchivName

Name des zu bearbeitenden Archives.

lpArchiv

Adresse der Struktur TLG_ARCHIV_STR (Seite 805) , in der die Parameter des Archives abgelegt werden.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archivinhalt gelesen

FALSE

Fehler

Benötigte Dateien

pdeccli.h

pdeccli.lib

pdeccli.dll

Verwandte Funktionen

TLGEnumArchives (Seite 866)	Archive auflisten
TLGReadArchiv6	Archivparameter lesen

Beispiele

Read archive (Seite 929)"TL01.cpp"

Siehe auch

- TLGEnumArchives (Seite 866)
- Read archive (Seite 929)
- TLG_ARCHIV_STR (Seite 805)

2.8.7 Funktionen zur Kurven- und Tabellendarstellung (RT Professional)

2.8.7.1 TLGCloseWindow (RT Professional)

Beschreibung

Mit der Funktion wird ein Applikationsfenster geschlossen.

Deklaration

```
BOOL TLGCloseWindow (  
    HWND          hwnd,  
    PCMN_ERROR    lpError );
```

Parameter

hwnd

Handle auf das Vaterfenster der zu schließenden Applikation.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Applikationsfenster geschlossen

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertcli.lib

pdertcli.dll

2.8.7.2 TLGDrawCurvesInDC (RT Professional)**Beschreibung**

Mit dieser Funktion können Kurven in einem Applikationsfenster dargestellt werden.

Deklaration

```

BOOL TLGDrawCurvesInDC (
    HDC                hDC,
    PRECT              lprect,
    PTLG_PROT_CURVE_INFOS  lpci,
    DWORD              dwNumberOfCurves,
    LPCMN_ERROR        lpError );

```

Parameter**hDC**

Handle auf das Vaterfenster, in dem die Kurve ausgegeben werden soll

lprect

Zeiger auf die Windows spezifische Struktur vom Typ RECT mit den Größenangaben des Fensters.

lpci

Zeiger auf die erste Struktur vom Typ TLG_PROT_CURVE_INFOS (Seite 824) mit Informationen wie die Kurven ausgegeben werden sollen.

dwNumberOfCurves

Anzahl der in lpci übergebenen Strukturen (Anzahl der darzustellenden Kurven).

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Kurven dargestellt

FALSE

Fehler

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NOT_SUPPORTED	diese Funktion wird nicht unterstützt

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

pdertcli.lib

pdertcli.dll

Siehe auch

TLG_PROT_CURVE_INFOS (Seite 824)

2.8.7.3 TLGInsertTemplateltem (RT Professional)

Beschreibung

Die Funktion fügt zu einer bestehende Fenstervorlage einen neuen Eintrag, entweder eine Kurvenvorlage oder eine Tabellenvorlage, hinzu.

Deklaration

```
BOOL TLGInsertTemplateItem (
    LPTSTR                lpszTemplateName,
    PTLG_TEMPLATEITEM_INFO lpptii,
    LPCMN_ERROR           lpError );
```

Parameter

lpszTemplateName

Zeiger auf den Namen der Fenstervorlage zu der eine Kurven- bzw. Tabellenvorlage hinzugefügt werden soll.

lpptii

Zeiger auf eine Struktur vom Typ TLG_TEMPLATEITEM_INFO (Seite 831) mit den Daten der zu ergänzenden Kurven- bzw. Tabellenvorlage.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Fenstervorlage erweitert

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen

TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NOT_SUPPORTED	diese Funktion wird nicht unterstützt

Benötigte Dateien

- pdertcli.h
- pde_glob.h
- pdertdef.h
- pdertcli.lib
- pdertcli.dll

Beispiele

Edit curve template - Example 1 (Seite 916) "TL02.cpp"

Siehe auch

- TLG_TEMPLATEITEM_INFO (Seite 831)
- Edit curve template - Example 1 (Seite 916)

2.8.7.4 TLGPressToolBarButton (RT Professional)

Beschreibung

Über dieser Funktion können Sie die mit den Schaltflächen der Symbolleiste verknüpften Funktionen auslösen.

Deklaration

```
BOOL TLGPressToolBarButton (  
    LPTSTR      lpszTemplate,  
    DWORD      dwButtonID,  
    PCMN_ERROR  lpError );
```

Parameter

lpszTemplate
Name der Vorlage

dwButtonID

Kennziffer für die auszulösende Schaltfläche:

Konstanten für Kurven und Tabellen	
TLG_BASIC_BTN_HELP	Online-Hilfe öffnen
TLG_BASIC_BTN_DLG	Parametrierungs-dialog öffnen
TLG_BASIC_BTN_FIRST	Erste Datensätze eines Archives anzeigen.
TLG_BASIC_BTN_PREV	Rückwärtsblättern im Archiv.
TLG_BASIC_BTN_NEXT	Vorwärtsblättern im Archiv.
TLG_BASIC_BTN_LAST	Letzte Datensätze eines Archives anzeigen.
TLG_BASIC_BTN_STARTSTOP	Aktivieren bzw. Deaktivieren der aktualisierten Darstellung einer Kurve oder Tabelle.
TLG_BASIC_BTN_PREV_ITEM	Vorherige Kure in Vordergrund bzw. Tabellenspalte zuerst.
TLG_BASIC_BTN_NEXT_ITEM	Nächste Kure in Vordergrund bzw. Tabellenspalte zuerst.
TLG_BASIC_BTN_ARC_VAR_SELECT	Öffnet den Archiv-variablen Auswahldialog.
TLG_BASIC_BTN_ITEM_SELECT	Öffnet den Auswahldialog für Kurven bzw Tabellenspalten.
TLG_BASIC_BTN_TIME_SELECT	Öffnet den Auswahldialog für den Darstellungszeitbereich.

Konstanten für Tabellen	
TLG_TABLE_BTN_EDIT	Aktivieren bzw. Deaktivieren der Editierfunktion.

Konstanten für Kurven	
TLG_CURVE_BTN_ZOOMIN	Aktivieren der Zoomfunktion zur Ausschnittsvergrößerung.
TLG_CURVE_BTN_ZOOMOUT	Deaktivieren der Zoomfunktion.
TLG_CURVE_BTN_1_TO_1	Umschalten in die Normaldarstellung einer Kurve.
TLG_CURVE_BTN_LINEAL	Aktivieren bzw. Deaktivieren des Lineals

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
IDS_API_ERROR_WINDOW_NOT_FOUND	Button (Fenster) konnte nicht gefunden werden
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NO_TYP_CHANGE_ALLOWED	Typänderung ist nicht erlaubt

Benötigte Dateien

- pdertcli.h
- pde_glob.h
- pdertdef.h
- pdertcli.lib
- pdertcli.dll

2.8.7.5 TLGSetRulerWindowVisible (RT Professional)

Beschreibung

Mit dieser Funktion wird das Datenfenster der Leselinie angezeigt oder ausgeblendet.

Deklaration

```
BOOL TLGSetRulerWindowVisible (  
    LPTSTR      lpszTemplateName,  
    BOOL        bShowRulerWindow,  
    LPCMN_ERROR lpError );
```

Parameter

lpszTemplateName

Zeiger auf den Namen der Fenstervorlage.

bShowRulerWindow

TRUE:	Die Leselinie mit Datenfenster wird angezeigt
FALSE:	Die Leselinie ohne Datenfenster wird angezeigt

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Der Parameter für die Sichtbarkeit des Datenfensters der Leselinie wurde neu übernommen.

FALSE

Fehler

Bemerkung

Bei der f(x)-Kurvenanzeige und der Tabelleanzeige ist die Funktion erst wirksam, wenn das entsprechende Bild mit dem Objekt angezeigt ist.

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
IDS_API_ERROR_WINDOW_NOT_FOUND	Button (Fenster) konnte nicht gefunden werden
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung

Benötigte Dateien

pdertcli.h
 pde_glob.h
 pdertdef.h
 pdertcli.lib
 pdertcli.dll

2.8.7.6 TLGShowWindow (RT Professional)**Beschreibung**

Die Funktion beeinflusst die Darstellungsform des Fensters.

Deklaration

```

BOOL TLGShowWindow (
    HWND          hwnd,
    DWORD         dwFlags,
    PCMN_ERROR    lpError );

```

Parameter**hwnd**

Fensterhandle des Runtime-Fensters.

dwFlags

Es gelten hier die selben Flags wie bei ::ShowWindow(HWND hwnd, int nCmdShow), so wie sie in WinUser.h definiert sind (SW_SHOW, SW_HIDE, ...).

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Applikationsfenster angezeigt

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

pdertcli.lib

pdertcli.dll

WinUser.h

2.8.8 Funktionen zum Bearbeiten von Zeitsystemen (RT Professional)

2.8.8.1 TLGEnumTime (RT Professional)

Beschreibung

Enumeriert die Namen aller Erfassungs- und Archivierungszeiten eines Projektes.

Deklaration

```

BOOL TLGEnumTime (
    HANDLE                hProject,
    TLG_ENUM_TIME_NAME_CALLBACK lpCallbackFunc,
    PVOID                lpUser,
    LPCMN_ERROR          lpoes );

```

Parameter

hProject

Handle des Projekts, in dem die aufzulistenden Zeitobjekte liegen.

lpCallbackFunc

Zeiger auf Ihre Callback-Funktion, die für jedes existierende Zeitobjekt einmal aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Zeitobjekte aufgelistet

FALSE

Fehler

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Verwandte Funktionen

TLG_ENUM_TIME_NAME_CALLBACK (Seite 904)	Zeitobjekte auflisten (Callback)
---	----------------------------------

Beispiele

Enum all acquisition and archiving times (Seite 919) "TL01.cpp"

Siehe auch

TLG_ENUM_TIME_NAME_CALLBACK (Seite 904)

Enum all acquisition and archiving times (Seite 919)

2.8.8.2 TLG_ENUM_TIME_NAME_CALLBACK (RT Professional)

Beschreibung

Um die vom System aufgelisteten Zeitobjekte auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUM_TIME_NAME_CALLBACK bereitstellen.

Deklaration

```
BOOL ( * TLG_ENUM_TIME_NAME_CALLBACK) (
    LPTSTR    lpszName,
    PVOID     lpUser );
```

Parameter

lpszName

Der Zeiger lpszName verweist auf den Namen des ersten Zeitobjektes.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Benötigte Dateien

pdecsccli.h

Verwandte Funktionen

TLGEnumTime (Seite 902)	Zeitobjekte auflisten
-------------------------	-----------------------

Beispiele

Enum all acquisition and archiving times (Seite 919) "TL01.cpp"

Siehe auch

Enum all acquisition and archiving times (Seite 919)

TLGEnumTime (Seite 902)

2.8.8.3 TLGEnumTimes (RT Professional)**Beschreibung**

Enumeriert alle Erfassungs und Archivierungszeiten.

Deklaration

```

BOOL TLGEnumTimes (
    TLG_ENUMTIMES_CALLBACK    lpfnCallback,
    LPVOID                    lpUser,
    PCMN_ERROR                 lpError );

```

Parameter

lpfnCallback

Zeiger auf Ihre Callback-Funktion,

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Zeitobjekte aufgelistet.

FALSE

Fehler

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

pdertcli.lib

pdertcli.dll

Verwandte Funktionen

TLG_ENUMTIMES_CALLBACK (Seite 907)	Zeitobjekte auflisten (Callback)
------------------------------------	----------------------------------

Siehe auch

TLG_ENUMTIMES_CALLBACK (Seite 907)

2.8.8.4 TLG_ENUMTIMES_CALLBACK (RT Professional)**Beschreibung**

Um die vom System aufgelisteten Zeitobjekte auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUMTIMES_CALLBACK bereitstellen.

Deklaration

```
BOOL ( * TLG_ENUMTIMES_CALLBACK) (  
    PTLG_TIMEDATA    lpTime,  
    PVOID            lpUser );
```

Parameter**lpTime**

Pro Zeitobjekt reserviert das System temporär Speicher der Länge der Struktur TLG_TIMEDATA (Seite 834). Der Zeiger lpTime verweist auf den Beginn des ersten Elementes.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert**TRUE**

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
 - API-Funktionen aus derselben DLL
 - Enumerationen, die weitere Enumerationen aufrufen
-

Benötigte Dateien

pdertcli.h
pde_glob.h
pdertdef.h

Verwandte Funktionen

TLGEnumTimes (Seite 905)	Zeitobjekte auflisten
--------------------------	-----------------------

Siehe auch

TLGEnumTimes (Seite 905)
TLG_TIMEDATA (Seite 834)

2.8.8.5 TLGReadTime (RT Professional)

Beschreibung

Lesen der Parameter eines bereits angelegten Zeitobjekts.

Deklaration

```
BOOL TLGReadTime (  
    HANDLE          hProject,  
    LPTSTR          lpszTimeName,  
    PTLG_TIME_STR   lpTime,  
    LPCMN_ERROR     lpoes );
```

Parameter

hProject

Handle des Projekts, in dem das zu bearbeitende Zeitobjekt liegt.

lpszTimeName

Zeiger auf den Namen des Zeitobjektes

lpTime

Adresse der Struktur TLG_TIME_STR (Seite 833) , mit den Daten des Zeitobjektes.

lpoes

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Parameter des Zeitobjektes gelesen.

FALSE

Fehler

Benötigte Dateien

pdecsccli.h

pdecsccli.lib

pdecsccli.dll

Beispiele

Read parameters of time object (Seite 933) "TL01.cpp"

Siehe auch

TLG_TIME_STR (Seite 833)

Read parameters of time object (Seite 933)

2.8.9 Funktionen zum Sichern und Wiederherstellen (RT Professional)

2.8.9.1 TLGEnumBackupEntries (RT Professional)

Beschreibung

Die Funktion wird diese Funktion nicht mehr unterstützt und liefert als Rückgabewert FALSE und den Fehlercode TLG_API_NOT_SUPPORTED.

Deklaration

```
BOOL TLGEnumBackupEntries (
    LPTSTR                lpszArchivName,
    TLG_ENUMBACKUP_ENTRIES lpfnCallback,
    LPVOID                lpUser,
    PCMN_ERROR            lpError );
```

Parameter

lpzArchivName

Zeiger auf den Namen des Archivs

lpfnCallback

Zeiger auf Ihre Callback-Funktion, die für jedes existierende Backup aufgerufen wird.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Backups aufgelistet

FALSE

Fehler

Fehlermeldungen

TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NOT_SUPPORTED	diese Funktion wird nicht unterstützt

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

pdertcli.lib

pdertcli.dll

Verwandte Funktionen

TLG_ENUMBACKUP_ENTRIES (Seite 911)	Backups auflisten (Callback)
------------------------------------	------------------------------

Siehe auch

TLG_ENUMBACKUP_ENTRIES (Seite 911)

2.8.9.2 TLG_ENUMBACKUP_ENTRIES (RT Professional)**Beschreibung**

Die Funktion TLGEnumBackupEntries wird nicht mehr unterstützt. Um die vom System aufgelisteten Backups auswerten zu können, müssen Sie eine Callback-Funktion vom Typ TLG_ENUMBACKUP_ENTRIES bereitstellen.

Deklaration

```
BOOL ( * TLG_ENUMBACKUP_ENTRIES) (
    PTLG_BACKUP_TABLE_INFO  lpbti,
    PVOID                   lpUser );
```

Parameter**lpbti**

Pro Backup reserviert das System temporär Speicher der Länge der Struktur TLG_BACKUP_TABLE_INFO (Seite 811). Der Zeiger lpbti verweist auf den Beginn des ersten Elementes.

lpUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert**TRUE**

Die Enumeration wird fortgesetzt.

FALSE

Die Enumeration wird abgebrochen.

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen ausderselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Benötigte Dateien

pdertcli.h
pde_glob.h
pdertdef.h

Verwandte Funktionen

TLGEnumBackupEntries (Seite 909)	Backups auflisten
----------------------------------	-------------------

Siehe auch

TLGEnumBackupEntries (Seite 909)
TLG_BACKUP_TABLE_INFO (Seite 811)

2.8.9.3 TLGExport (RT Professional)

Beschreibung

Mit der Funktion können Teile eines Archives exportiert werden. Die Selektion der zu exportierenden Datensätze erfolgt über die Struktur TLG_IO_BACKUP_SELECT.

Deklaration

```
BOOL TLGExport (
    LPTSTR                lpszArchivName,
    LPTSTR                lpszFileName,
    PTLG_IO_BACKUP_SELECT lpibs,
    DWORD                dwJobFlags,
    DWORD                dwFormatFlags,
    PCMN_ERROR            lpError );
```

Parameter

IpszArchivName

Zeiger auf den Namen des Archivs. Der Name ist in der Form "Archivname\Variablenname" zusammengesetzt anzugeben.

Mit TLGEnumArchivsEx und TLGEnumVariablesEx können die Bestandteile des Namens ermittelt werden.

IpszFileName

Zeiger auf den Namen der Datei in die die Daten exportiert werden sollen.

lpibs

Adresse der Übergabe-Struktur TLG_IO_BACKUP_SELECT (Seite 823) für die Selektionsparameter.

dwJobFlags

Der Parameter ist für spätere Erweiterungen vorgesehen und mit 0L zu versorgen.

dwFormatFlags

Formatspezifizierer:

TLG_BAKFMT_CSV:	CSV-Format (Comma Separated Value)
-----------------	------------------------------------

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Archivdaten ausgelagert

FALSE

Fehler

Bemerkung

Die Verwendung dieser Funktion ist nur bei Folgearchiven, Umlaufarchiven sowie bei Verdichtungsarchiven sinnvoll.

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt

Benötigte Dateien

pdertcli.h
pde_glob.h
pdertdef.h
pdertcli.lib
pdertcli.dll

Siehe auch

TLG_IO_BACKUP_SELECT (Seite 823)

2.8.9.4 TLGGetBackupSize (RT Professional)

Beschreibung

Die Funktion wird nicht mehr unterstützt und liefert als Rückgabewert FALSE und den Fehlercode TLG_API_NOT_SUPPORTED.

Deklaration

```
BOOL TLGGetBackupSize (  
    LPTSTR                lpszArchivName,  
    DWORD*                lpdwSizeOfTable,  
    PTLG_IO_BACKUP_SELECT lpibs,  
    DWORD                 dwJobFlags,  
    DWORD                 dwFormatFlags,  
    PCMN_ERROR            lpError );
```

Parameter

lpszArchivName

Zeiger auf den Namen des Archivs

lpdwSizeOfTable

Adresse eines DWORD, in dem die Größe der selektierten Daten in Bytes geschrieben wird.

lpibs

Adresse der Übergabestruktur TLG_IO_BACKUP_SELECT (Seite 823) für die Selektionsparameter.

dwJobFlags

Auftragsspezifische Kennzeichnung, möglich sind:

TLG_BACKUP_EXPORT:	Einfaches Backup ohne Löschen der Sourcedaten.
TLG_BACKUP_EVACUATE:	Auslagern mit Löschen der Sourcen

dwFormatFlags

Formatspezifizierer:

TLG_BAKFMT_CSV:	CSV-Format (Comma Seperated Value)-
-----------------	-------------------------------------

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Größe der Daten ermittelt

FALSE

Fehler

Fehlermeldungen

TLG_API_ERR_SUPPLY	ein Fehler ist aufgetreten
TLG_API_INVALID_PARAM	ungültige/falsche Parameterversorgung
TLG_API_NO_INTERFACE	Zugriff auf unterlagerte COM-Schnittstellen fehlgeschlagen
TLG_API_NO_CONNECTION	keine Verbindung zu einem aktivierten Projekt hergestellt
TLG_API_NOT_SUPPORTED	diese Funktion wird nicht unterstützt

Benötigte Dateien

pdertcli.h

pde_glob.h

pdertdef.h

pdertcli.lib

pdertcli.dll

Siehe auch

TLG_IO_BACKUP_SELECT (Seite 823)

2.8.10 Beispiele (RT Professional)

2.8.10.1 Edit curve template - Example 1 (RT Professional)

Beispiel

```
// =====
// =====
// : Modul with examples to TagLogging-API
// *****
// Copyright (C) 1995/96 SIEMENS AG, AUT 913 All rights reserved
// *****
#include "stdafx.h" // if MFC classes
// #include "odkapi.h" // if console application
#include "TL02.h"
#include <string.h>
#include <stdlib.h>
#include <malloc.h>
#include <time.h>

// IMPLEMENTATION
//{{ODK_EXAMPLE}Edit curve template (example no. 1) (TLG)}
//{{FUNCTION}TLGConnect (TLG)}
//{{FUNCTION}TLGDisconnect (TLG)}
//{{FUNCTION}TLGInsertTemplateItem (TLG)}
//{{FUNCTION}(END)}
// =====
// Function: MyTLGInsertTemplateItem(void) ODK TL RT
// =====
// : Edit curve template
// : Defines a new curve with previous settings
// =====
void MyTLGInsertTemplateItem (void)
{
    #define COLOR_RED 0x000000FF

//     Farbe           sym. Name           Value (hexadezimal)   Value (dezimal)
//     schwarz         CO_BLACK           0x00000000           0
//     weiß            CO_WHITE           0x00FFFFFF           16777215
//     rot             CO_RED             0x000000FF           255
//     dunkelrot       CO_DKRED           0x00000080           128
//     grün            CO_GREEN           0x0000FF00           65280
//     dunkelgrün      CO_DKGREEN         0x00008000           32768
//     blau            CO_BLUE           0x00FF0000           16711680
//     dunkelblau      CO_DKBLUE          0x00800000           8388608
//     gelb            CO_YELLOW          0x0000FFFF           65535
//     dunkelgelb      CO_DKYELLOW        0x00008080           32896
//     zyan            CO_CYAN            0x00FFFF00           1048320
//     dunkelzyan      CO_DKCYAN          0x00808000           8421376
//     magenta         CO_MAGENTA         0x00FF00FF           16711935
//     dunkelmagenta   CO_DKMAGENTA       0x00800080           8388736
//     hellgrau        CO_LTGRAY          0x00C0C0C0           12632256
//     dunkelgrau      CO_DKGRAY          0x00808080           8421504
```

```

// scripts
BOOL ret = FALSE;
CMN_ERROR Error;
TCHAR szText[255];
HWND hwndParent = NULL;
TLG_TEMPLATEITEM_INFO TemplateItem;
memset(&Error,0,sizeof(CMN_ERROR));
ret = TLGConnect(hwndParent, &Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGInsertTemplateItem: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
}
else
{
    memset(&TemplateItem, 0, sizeof(TLG_TEMPLATEITEM_INFO));
    _tcsncpy_s(TemplateItem.szArchivName, _countof(TemplateItem.szArchivName),
_T("PLSMWA"), _TRUNCATE); // archive name
    _tcsncpy_s(TemplateItem.szTemplateName, _countof(TemplateItem.szTemplateName),
_T("KV_PID1"), _TRUNCATE); // template for curve window
    _tcsncpy_s(TemplateItem.szVariableName, _countof(TemplateItem.szVariableName),
_T("VPID1_X"), _TRUNCATE); // DM variable name
    _tcsncpy_s(TemplateItem.szDMVariableName, _countof(TemplateItem.szDMVariableName),
_T("VPID1_X"), _TRUNCATE);
    _tcsncpy_s(TemplateItem.szTemplateItemName,
_countof(TemplateItem.szTemplateItemName), _T("KPID3_X"), _TRUNCATE); // curve name
    TemplateItem.dwTemplateType = TLG_TEMPLATE_CURVE;
    TemplateItem.dwTemplateItemType = TLG_TEMPLATEITEM_CURVE;
    TemplateItem.dwReadAccessLevel = 1;
    TemplateItem.dwWriteAccessLevel = 1;
    TemplateItem.dwArchivTyp = TLG_ARCTYP_PROCESS; // _USER, _PROCESS, _COMPRESS
    _tcsncpy_s(TemplateItem.szTimeNameRange, _countof(TemplateItem.szTimeNameRange),
_T("500 ms"), _TRUNCATE);
    TemplateItem.fVisible = TRUE;
    TemplateItem.tplInfo.tplCurve.csx.dwDataTyp = TLG_DATATYP_TIMERANGE;
    TemplateItem.tplInfo.tplCurve.csx.fAutoRange = TRUE;
    TemplateItem.tplInfo.tplCurve.csx.fActualize = TRUE;
    TemplateItem.tplInfo.tplCurve.csx.dwBufferSize = 10;
    TemplateItem.tplInfo.tplCurve.csy.dwDataTyp = TLG_DATATYP_BLOCKDATA;
    TemplateItem.tplInfo.tplCurve.csy.fAutoRange = TRUE;
    TemplateItem.tplInfo.tplCurve.csy.crColor = COLOR_RED;
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = TLGInsertTemplateItem(TemplateItem.szTemplateName, &TemplateItem, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGInsertTemplateItem: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("InsertItem:
TemplateItemName= %s "), TemplateItem.szTemplateItemName);
    }
}

```

```
    }
    ODKTrace(szText);
    // Disconnect
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = TLGDisconnect(&Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGDisconnect:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

[TLGInsertTemplateltem \(Seite 897\)](#)

[TLGConnect \(Seite 850\)](#)

[TLGDisconnect \(Seite 851\)](#)

2.8.10.2 Enum all acquisition and archiving times (RT Professional)

Beispiel

```

/ =====
// =====
// : Modul with examples to TagLogging-API
// *****
// Copyright (C) 1995/96 SIEMENS AG, AUT 913 All rights reserved
// *****
#include "stdafx.h" // if MFC classes
#include "stdlib.h"
#include "TL01.h"
#include "dm01.h"
// IMPLEMENTATION
// =====
// 4. Callback
// =====
//{{ODK_EXAMPLE}Enum all acquisition and archiving times (TLG)}
//{{FUNCTION}TLGCSCSConnect (TLG)}
//{{FUNCTION}TLGCSDisConnect (TLG)}
//{{FUNCTION}TLGEnumTime (TLG)}
//{{FUNCTION}TLG_ENUM_TIME_NAME_CALLBACK (TLG)}
//{{FUNCTION}(END)}
// =====
// Function: TLGEnumTime(void) ODK TL CS
// =====
// : Enum all acquisition and archiving times
// :
// =====
BOOL MyTLGEnumTimeNameCallback (LPTSTR szTime, PVOID lpUser)
{
    TCHAR szText[255];
    sprintf(szText, "...Enum Time %s", szTime);
    ODKTrace(szText);
    //printf("%s\r\n", szText);
    return TRUE;
}

void MyTLGEnumTime(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    TCHAR szText[255];
    HANDLE hProject;
    CMN_ERROR Error;
    PVOID pUser = NULL;
    sprintf(szText, "TLGEnumTime");
    ODKTrace(szText);
    //printf("%s\r\n", szText);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGCSCSConnect(NULL, &Error);
    if (FALSE == ret)

```

```

{
    sprintf(szText, "Error in TLGCSConnect: E1= 0x%08lx ; E2= 0x%08lx ; %s",
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
    //printf("%s\r\n", szText);
}
else
{
    sprintf(szText, " TLGCSConnect");
    ODKTrace(szText);
    //printf("%s\r\n", szText);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGOpenProject(&hProject, /*PROJ_PATH*/g_szProjectFile, NULL, &Error);
    if (FALSE == ret)
    {
        sprintf(szText, "Error in TLGOpenProject: E1= 0x%08lx ; E2= 0x%08lx ; %s",
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
        //printf("%s\r\n", szText);
    }
    else //TLGOpenProject OK
    {
        sprintf(szText, " TLGOpenProject");
        ODKTrace(szText);
        //printf("%s\r\n", szText);
        // read projected archiving times
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = TLGEnumTime(hProject, MyTLGEnumTimeNameCallback, pUser, &Error);
        if (FALSE == ret)
        {
            sprintf(szText, "Error in TLGEnumTime: E1= 0x%08lx ; E2= 0x%08lx ; %s",
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            sprintf(szText, " TLGEnumTime");
        }
        ODKTrace(szText);
        //printf("%s\r\n", szText);
        //TLGCloseProject();
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = TLGCloseProject(hProject, &Error);
        if (FALSE == ret)
        {
            sprintf(szText, "Error in TLGCloseProject: E1= 0x%08lx ; E2= 0x%08lx ; %s",
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            sprintf(szText, " TLGCloseProject");
        }
        ODKTrace(szText);
        //printf("%s\r\n", szText);
    }
    // Disconnect
    memset(&Error, 0, sizeof(CMN_ERROR));
}

```

```
ret = TLGCSDisConnect(&Error);
if (FALSE == ret)
{
    sprintf(szText, "Error in TLGCSDisConnect: E1= 0x%08lx ; E2= 0x%08lx ; %s",
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    sprintf(szText, " TLGCSDisConnect");
}
ODKTrace(szText);
//printf("%s\r\n",szText);
}
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

TLGCSCConnect (Seite 845)

TLG_ENUM_TIME_NAME_CALLBACK (Seite 904)

TLGEnumTime (Seite 902)

TLGDisconnect (Seite 851)

TLGCSDisConnect (Seite 847)

2.8.10.3 Enum archives (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enum archives (TLG)}
//{{FUNCTION}TLGCSConnect (TLG)}
//{{FUNCTION}TLGCSDisConnect (TLG)}
//{{FUNCTION}TLGOpenProject (TLG)}
//{{FUNCTION}TLGCloseProject (TLG)}
//{{FUNCTION}TLGEnumArchives (TLG)}
//{{FUNCTION}TLG_ENUM_ARCHIV_CALLBACK (TLG)}
//{{FUNCTION}(END)}
// =====
// Function: TLGEnumArchives(void) ODK TL CS
// =====
// : Enum archives
// :
// =====
BOOL MyTLGEnumArchivCallback(LPTSTR szArchives, PVOID lpUser)
{
    lpUser;
    TCHAR szText[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...Enum Archive=%s "),szArchives);
    ODKTrace(szText);
    return TRUE;
}

void MyTLGEnumArchives(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    CMN_ERROR Error;
    HANDLE hProject;
    PVOID pUser = NULL;
    TCHAR szText[255];
    TCHAR szArchivName[255];
    _tcsncpy_s(szArchivName, _countof(szArchivName), _T("PLSMWA"), _TRUNCATE);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    //TLGConnect
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = TLGCSConnect(NULL, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCSConnect:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        //TLGOpenProject
        memset(&Error,0,sizeof(CMN_ERROR));
        ret = TLGOpenProject(&hProject, /*PROJ_PATH*/g_szProjectFile, NULL, &Error);
        if (FALSE == ret)

```

```

    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGOpenProject: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        // TLGEnumArchives
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = TLGEnumArchives(hProject, MyTLGEnumArchivCallback, pUser, &Error);
        if (FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGEnumArchives: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
        }
        else
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  TLGEnumArchives"));
        }
        ODKTrace(szText);
    }
    //CloseProject
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGCloseProject(hProject, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCloseProject:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  TLGCloseProject"));
    }
    ODKTrace(szText);
}
// Disconnect
memset(&Error, 0, sizeof(CMN_ERROR));
ret = TLGCSDisConnect(&Error);
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCSDisConnect: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  TLGCSDisConnect"));
}
    ODKTrace(szText);
}
//{{ODK_EXAMPLE} (END) }

```

Siehe auch

TLGCSConnect (Seite 845)

TLGOpenProject (Seite 854)

TLGCloseProject (Seite 853)

TLGEnumArchives (Seite 866)

TLG_ENUM_ARCHIV_CALLBACK (Seite 867)

TLGCSDisConnect (Seite 847)

2.8.10.4 Enum variables of an archive (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enum variables of a archive (TLG)}
//{{FUNCTION}TLG_ENUM_PROJECT_NAME_CALLBACK (TLG)}
//{{FUNCTION}TLGEnumVariablesEx (TLG)}
//{{FUNCTION}TLG_ENUMVARIABLES (TLG)}
//{{FUNCTION}(END)}
// =====
// Function: MyTLGEnumVariablesEx(void) ODK TL CS
// =====
// : Edit archive variables
// :
// =====
BOOL MyTLGEnumVariablesExCallback(PTLG_VARIABLE_INFO lpvi, LPVOID lpUser)
{
    lpUser;
    TCHAR szText[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Enum Variables %d %s "), lpvi-
>dwVariableTyp, lpvi->szVariableName);
    ODKTrace(szText);
    return TRUE;
}

void MyTLGEnumVariablesEx(void)
{
    BOOL ret = FALSE;
    TCHAR szText[255];
    CMN_ERROR Error;
    HANDLE hProject = NULL;
    TCHAR szArchivName[255];
    HWND hwndParent = NULL;
    VOID* pUser = NULL;
    _tcsncpy_s(szArchivName, _countof(szArchivName), _T("PLSMWA"), _TRUNCATE);
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGConnect(hwndParent, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGConnect: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TLGConnect"));
        ODKTrace(szText);

        // Info
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = TLGEnumVariablesEx (szArchivName, MyTLGEnumVariablesExCallback,
            pUser, &Error );
    }
}

```

```
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGEnumVariablesEx: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TLGEnumVariablesEx: OK"));
    }
    ODKTrace(szText);

    memset(&Error,0,sizeof(CMN_ERROR));
    ret = TLGDisconnect(&Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGDisconnect:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
}
//{{ODK_EXAMPLE}(END)}
```

Siehe auch

- TLGOpenProject (Seite 854)
- TLGEnumVariablesEx (Seite 861)
- TLG_ENUMVARIABLES (Seite 863)

2.8.10.5 Enumerate archives (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Enumerate archives (TLG)}
//{{FUNCTION}TLGConnect (TLG)}
//{{FUNCTION}TLGDisconnect (TLG)}
//{{FUNCTION}TLGEnumArchivs (TLG)}
//{{FUNCTION}TLG_ENUMTABLES (TLG)}
//{{FUNCTION}(END)}
// =====
// Function: MyTLGEnumArchivs(void) ODK TL RT
// =====
// : Enumerate archives
// =====
BOOL MyTLGEnumTablesCallback(LPTSTR lpTableName, PTLG_TABLE_INFO lpti, PVOID lpUser)
{
    lpUser;
    TCHAR szText[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TableName=%s"), lpTableName);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" ...ArchivTyp=%d"), lpti->dwArchivTyp);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" ...ArchivName=%s"), lpti->szArchivName);
    ODKTrace(szText);
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" ...dwSaveTyp=%d"), lpti->dwSaveTyp);
    ODKTrace(szText);
    return TRUE;
}

void MyTLGEnumArchivs(void)
{
    BOOL ret = FALSE;
    //DWORD StartTime = 0;
    //DWORD StopTime = 0;
    void* lpUser = NULL;
    CMN_ERROR Error;
    TCHAR szText[255];
    HWND hwndParent = NULL;
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGConnect(hwndParent, &Error);
    if(FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGConnect: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TLGConnect"));
    }
}

```

```
ODKTrace(szText);

memset(&Error,0,sizeof(CMN_ERROR));
ret = TLGEnumArchivs(TLG_ARCTYP_PROCESS, MyTLGEnumTablesCallback, lpUser, &Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGEnumArchivs:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TLGEnumArchivs: OK"));
}
ODKTrace(szText);
memset(&Error,0,sizeof(CMN_ERROR));
ret = TLGDisconnect(&Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGDisconnect:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    ODKTrace(szText);
}
}
}
//{{ODK_EXAMPLE} (END) }
```

Siehe auch

- TLGConnect (Seite 850)
- TLGDisconnect (Seite 851)
- TLGEnumArchivs (Seite 868)

2.8.10.6 Read archive (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Read_archive (TLG)}
//{{FUNCTION}TLGCSConnect (TLG)}
//{{FUNCTION}TLGCSDisConnect (TLG)}
//{{FUNCTION}TLGOpenProject (TLG)}
//{{FUNCTION}TLGCloseProject (TLG)}
//{{FUNCTION}TLGReadArchiv (TLG)}
//{{FUNCTION}TLGReadVariable (TLG)}
//{{FUNCTION}(END)}
// =====
// Function: TLGReadArchiv(void) ODK TL CS
// =====
// : Read archive
// :
// =====
void MyTLGReadArchiv(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    CMN_ERROR Error;
    HANDLE hProject;
    //PVOID pUser = NULL;
    TCHAR szText[255];
    TCHAR szArchivName[255];
    TCHAR szVarName[255];
    TLG_ARCHIV_STR TLGArchivStruct;
    TLG_VAR_STR TLGVarStruct;
    _tcsncpy_s(szArchivName, _countof(szArchivName), _T("PLSMWA"), _TRUNCATE);
    _tcsncpy_s(szVarName, _countof(szVarName), _T("X"), _TRUNCATE);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGCSConnect(NULL, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCSConnect: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        memset(&Error, 0, sizeof(CMN_ERROR));
        ret = TLGOpenProject(&hProject, /*PROJ_PATH*/g_szProjectFile, NULL, &Error);
        if (FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGOpenProject:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
            ODKTrace(szText);
        }
        else
    }
}

```

```

{
    // read
    memset(&TLGArchivStruct,0,sizeof(TLG_ARCHIV_STR));
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = TLGReadArchiv(hProject, szArchivName, &TLGArchivStruct, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGReadArchiv:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TLGReadArchiv"));
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  ArcName=%s Comments=
%s dwTyp=0x%04x dwRecSize=%d"),
            TLGArchivStruct.szName,
            TLGArchivStruct.szComment,
            TLGArchivStruct.dwTyp,
            TLGArchivStruct.dwRecordSize);
        ODKTrace(szText);
    }
    memset(&TLGVarStruct, 0, sizeof(TLG_VAR_STR));
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = TLGReadVariable(hProject, szArchivName, szVarName, &TLGVarStruct, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGReadVariable: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TLGReadVariable"));
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  szVarName=%s"),
            TLGVarStruct.szVarName);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  szProcName=%s"),
            TLGVarStruct.szProcName);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  dwVarType=0x%04X"),
            TLGVarStruct.dwVarType);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  dwArchivStyle=0x
%04X"), TLGVarStruct.dwArchivStyle);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  szWriteBackTo=%s"),
            TLGVarStruct.szWriteBackTo);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  dwSupply=0x%04X"),
            TLGVarStruct.dwSupply);
        ODKTrace(szText);
    }
}

```

```

        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   fLocked=%d"),
TLGVarStruct.fLocked);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   szComment=%s"),
TLGVarStruct.szComment);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   szRecordCycle=%s"),
TLGVarStruct.szRecordCycle);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   szArchivCycle=%s"),
TLGVarStruct.szArchivCycle);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   dwMultiple=0x%04X"),
TLGVarStruct.dwMultiple);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   dwValueFlow=0x
%04X"), TLGVarStruct.dwValueFlow);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   dwValueFollow=0x
%04X"), TLGVarStruct.dwValueFollow);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   dwSaveByFault=0x
%04X"), TLGVarStruct.dwSaveByFault);
        ODKTrace(szText);
    }
}
//CloseProject
memset(&Error,0,sizeof(CMN_ERROR));
ret = TLGCloseProject(hProject,&Error);
if(FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCloseProject:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   TLGCloseProject"));
}
ODKTrace(szText);
}
// Disconnect
memset(&Error,0,sizeof(CMN_ERROR));
ret = TLGCSDisConnect(&Error);
if (FALSE == ret)
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCSDisConnect: E1=
0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
}
else
{
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("   TLGCSDisConnect"));
}
ODKTrace(szText);
}

```

```
// {{ODK_EXAMPLE}} (END) }
```

Siehe auch

- TLGOpenProject (Seite 854)
- TLGCloseProject (Seite 853)
- TLGCSConnect (Seite 845)
- TLGReadVariable (Seite 864)
- TLGReadArchiv (Seite 893)
- TLGCSDisConnect (Seite 847)

2.8.10.7 Read parameters of time object (RT Professional)

Beispiel

```

//{{ODK_EXAMPLE}Read parameters of time object (TLG)}
//{{FUNCTION}TLGCSConnect (TLG)}
//{{FUNCTION}TLGCSDisConnect (TLG)}
//{{FUNCTION}TLGReadTime (TLG)}
//{{FUNCTION}TLGOpenProject (TLG)}
//{{FUNCTION}TLGCloseProject (TLG)}
//{{FUNCTION}(END)}
// =====
// Function: TLGReadTime(void) ODK TL CS
// =====
// : Reads the parameters of an already existing time object.
// :
// =====
void MyTLGReadTime(void)
{
    // #define PROJ_PATH "C:\\siemens\\odk\\samples\\projects\\demo\\odk.mcp"
    BOOL ret = FALSE;
    TCHAR szText[255];
    HANDLE hProject;
    CMN_ERROR Error;
    TLG_TIME_STR TLGTimeStruct;
    TCHAR szTime[255];
    _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("TLGReadTime"));
    ODKTrace(szText);
    MyDMEnumOpenedProjects(); // open the DM and set the g_szProjectFile
    memset(&Error,0,sizeof(CMN_ERROR));
    ret = TLGCSConnect(NULL, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCSConnect: E1= 0x
%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        memset(&Error,0,sizeof(CMN_ERROR));
        ret = TLGOpenProject(&hProject, /*PROJ_PATH*/g_szProjectFile, NULL, &Error);
        if (FALSE == ret)
        {
            _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGOpenProject:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
                Error.dwError1, Error.dwError2, Error.szErrorText);
            ODKTrace(szText);
        }
        else
        {
            // read properties of selected time
            memset(&Error,0,sizeof(CMN_ERROR));
            memset(&TLGTimeStruct, 0, sizeof(TLG_TIME_STR));

```

```

    _tcsncpy_s(szTime, _countof(szTime), _T("500 ms"), _TRUNCATE);
    ret = TLGReadTime(hProject, szTime, &TLGTimeStruct, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGReadTime:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
        ODKTrace(szText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...szTimeName = %s
"), TLGTimeStruct.szTimeName);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...dwBasis = %10d
Millisec"), TLGTimeStruct.dwBasis);
        ODKTrace(szText);
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("...dwFactor = %10d
"), TLGTimeStruct.dwFactor);
        ODKTrace(szText);
    }
    // save changes permanent
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGSaveProject(hProject, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGSaveProject: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" TLGSaveProject"));
    }
    ODKTrace(szText);
    //TLGCloseProject();
    memset(&Error, 0, sizeof(CMN_ERROR));
    ret = TLGCloseProject(hProject, &Error);
    if (FALSE == ret)
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in
TLGCloseProject: E1= 0x%08lx ; E2= 0x%08lx ; %s"),
            Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T(" TLGCloseProject"));
    }
    ODKTrace(szText);
}
// Disconnect
memset(&Error, 0, sizeof(CMN_ERROR));
ret = TLGCSDisConnect(&Error);
if (FALSE == ret)
{

```

```

        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("Error in TLGCSDisConnect:
E1= 0x%08lx ; E2= 0x%08lx ; %s"),
        Error.dwError1, Error.dwError2, Error.szErrorText);
    }
    else
    {
        _sntprintf_s(szText, _countof(szText), _TRUNCATE, _T("  TLGCSDisConnect"));
    }
    ODKTrace(szText);
}
}
//{{ODK_EXAMPLE}(END)}

```

Siehe auch

[TLGOpenProject \(Seite 854\)](#)
[TLGCloseProject \(Seite 853\)](#)
[TLGCSCConnect \(Seite 845\)](#)
[TLGReadTime \(Seite 908\)](#)
[TLGCSDisConnect \(Seite 847\)](#)

2.9 Funktionen für Rezepturen (RT Professional)**2.9.1 Grundlagen (RT Professional)****2.9.1.1 Übersicht über die Funktionen (RT Professional)****Übersicht**

uaSetLocalEvents (Seite 947)	Setzt die lokalen Ereignisse
uaArchiveClose (Seite 958)	Verbindung zu aktuellem Anwenderarchiv lösen
uaArchiveDelete (Seite 959)	Datensätze im Anwenderarchiv löschen
uaArchiveExport (Seite 960)	Anwenderarchiv exportieren
uaArchiveGetCount (Seite 972)	lesen der Anzahl der Datensätze
uaArchiveGetFieldLength (Seite 980)	lesen der aktuellen Feldlänge
uaArchiveGetFieldName (Seite 981)	lesen des aktuellen Feldnamens
uaArchiveGetFields (Seite 983)	Anzahl der Felder ermitteln
uaArchiveGetFieldType (Seite 984)	lesen des Typs des aktuellen Feldes
uaArchiveGetFieldValueDate (Seite 985)	Datum und Uhrzeit aus aktuellem Datenfeld lesen
uaArchiveGetFieldValueDouble (Seite 986)	Double Wert aus aktuellem Datenfeld lesen
uaArchiveGetFieldValueLong (Seite 987)	Long Wert aus aktuellem Datenfeld lesen

2.9 Funktionen für Rezepturen (RT Professional)

uaArchiveGetFieldValueString (Seite 988)	String aus aktuellem Datenfeld lesen
uaArchiveGetFilter (Seite 973)	Filtereinstellung lesen
uaArchiveGetID (Seite 962)	ID des aktuellen Datenfeldes lesen
uaArchiveGetName (Seite 963)	Liest Name des Archivs
uaArchiveGetSort (Seite 964)	Sortierung des aktuellen Datenfeldes lesen
uaArchiveImport (Seite 965)	Anwenderarchiv importieren
uaArchiveInsert (Seite 974)	Datensatz ins Anwenderarchiv einfügen
uaArchiveMoveFirst (Seite 975)	auf ersten Datensatz positionieren
uaArchiveMoveLast (Seite 976)	auf letzten Datensatz positionieren
uaArchiveMoveNext (Seite 977)	auf nächsten Datensatz positionieren
uaArchiveMovePrevious (Seite 978)	auf vorherigen Datensatz positionieren
uaArchiveOpen (Seite 967)	Verbindung zu aktuellem Anwenderarchiv erstellen
uaArchiveReadTagValuesByName (Seite 1002)	Variablenwerte anhand des Namens lesen
uaArchiveReadTagValues (Seite 1001)	Variablenwerte lesen
uaArchiveRequery (Seite 996)	Neuaufsetzen auf aktuelles Anwenderarchiv
uaArchiveSetFieldValueDate (Seite 990)	Datum und Uhrzeit in aktuelles Datenfeld schreiben
uaArchiveSetFieldValueDouble (Seite 991)	Double Wert in aktuelles Datenfeld schreiben
uaArchiveSetFieldValueLong (Seite 992)	Long Wert in aktuelles Datenfeld schreiben
uaArchiveSetFieldValueString (Seite 993)	String in aktuelles Datenfeld schreiben
uaArchiveSetFilter (Seite 997)	Filter setzen
uaArchiveSetSort (Seite 999)	Sortierkriterium einstellen
uaArchiveUpdate (Seite 968)	Datensatz ins Anwenderarchiv aktualisieren
uaArchiveWriteTagValuesByName (Seite 1005)	Datensatz anhand des Namens in Variablen schreiben
uaArchiveWriteTagValues (Seite 1003)	Datensatz in Variablen schreiben
uaConnect (Seite 948)	Verbindung zu einem User Archive aufbauen (Runtime)
uaDisconnect (Seite 949)	Verbindung zu Anwenderarchiv wieder lösen
uaGetArchive (Seite 969)	Archivkonfiguration lesen
uaGetField (Seite 995)	Feldkonfiguration lesen
uaGetLastError (Seite 943)	letzten Fehler Anwenderarchiv abfragen
AUTOHOTSPOT	zuletzt aufgetretenen Fehlers der COM-Funktionen abfragen
uaGetNumArchives (Seite 970)	Find Anzahl von konfigurierten Anwenderarchiven
uaGetNumFields (Seite 979)	Anzahl der Felder ermitteln
uaIsActive (Seite 945)	Abfrage ob ein Archiv in Runtime geöffnet ist
uaOpenArchives (Seite 971)	Abfrage wie viele User Archive in Runtime geöffnet sind
uaOpenViews (Seite 1000)	Abfrage wie viele User Archive Ansichten in Runtime geöffnet sind
uaQueryArchiveByName (Seite 952)	Verbindung zu Anwenderarchiv anhand Name setzen (Runtime)
uaQueryArchive (Seite 950)	Verbindung zu Anwenderarchiv setzen (Runtime)
uaQueryConfiguration (Seite 954)	Verbindung zur Konfiguration abfragen

uaReleaseArchive (Seite 955)	Verbindung zu Anwenderarchiv wieder lösen (Runtime)
uaReleaseConfiguration (Seite 957)	Verbindung zur Konfiguration wieder lösen

Siehe auch

uaUsers (Seite 946)

2.9.1.2 Aufbau von Rezepturen (RT Professional)**Einleitung**

<i>dwKeyField</i>	0	1	2			n
	Record-number	Field 1	Field 2			Field n
	1					
	2					
	3					
	m					

<i>hRecord</i>					
----------------	--	--	--	--	--

Anwenderarchive sind einzelne Tabellen in der Datenbank, die mit Ausnahme der ersten Spalte einen völlig freien Aufbau besitzen. Die Datentypen sind auf char-, double-, int- und DateTime- Felder beschränkt.

Die Get- und Set- Field-Funktion arbeiten auf einen internen hRecord-Datensatz, der mit Insert und Update ins Archiv eingeschrieben werden kann.

Wird das User Archive direkt über die Datenbank (ODBC/ SQL) verändert, wird kein Online-Ableich ausgeführt.

Konfiguration der User Archives

Der erste Schritt ist die Konfiguration von Anwenderarchiven. Diese Konfiguration kann mit dem User Archives-Editor, mit den User Archives-Scriptfunktionen oder mit den User Archives-API-Funktionen erfolgen. Dabei entsprechen die Scriptfunktionen gleichlautend den API-Funktionen.

Konfiguration mit den User Archives-API-Funktionen

Die Funktion UAQueryConfiguration liefert einen Handle (UAHCONFIG) für die Konfigurationsfunktionen. Mit diesem Handle können die Konfigurationsfunktionen UASetArchive, UAAddArchive, UASetField, UAAddField etc aufgerufen werden. Die Funktion "UAREleaseConfiguration" beendet die Konfiguration der User Archives.

Diese Funktionen sind in Multiclient-Projekten nicht verfügbar.

Verbindung zu User Archives aufbauen

Als nächstes muss die Standardfunktion UAConnect aufgerufen werden, um eine Verbindung zur Komponente User Archives aufzubauen. UAConnect erstellt den Handle UAHCONNECT, mit dem Archive (keine Views) geöffnet und geschlossen werden können.

Auskunftsfunktionen im Runtime-Betrieb

Die Funktion UAIsActive zeigt an, ob ein Archiv aktiv in Runtime geöffnet ist. UAUsers gibt die Anzahl der angemeldeten Benutzer, UAOpenArchives die Anzahl geöffneter Archive und UAOpenViews die Anzahl geöffneter Ansichten (Views) zurück.

Öffnen der Runtime-Funktionen

Für den Runtime-Betrieb wird ein projektiertes Anwenderarchiv vorausgesetzt. Mit den Funktionen UAQueryArchive und UAQueryArchiveByName wird ein Handle für die Runtime-Funktionen geliefert. Nach Öffnen des Archivs mit der Funktion UAArchiveOpen können die User Archives Runtime-Funktionen verwendet werden.

Funktionen für den Runtime-Betrieb

Mit den Funktionen UAArchiveMoveNext, UAArchiveMovePrevious, UAArchiveMoveFirst und UAArchiveMoveLast wird der Positionszeiger verschoben. Über den Handle hArchive besteht eine eindeutige Zuordnung zu einem Datensatz des Anwenderarchivs. Diese Zuordnung erlaubt auch eine indirekte Adressierung, wie dies zum Beispiel für Bildschirmmasken benötigt wird.

Die Funktion UAArchiveUpdate speichert den temporären Datensatz im Archiv und überschreibt dort den Datensatz, auf dem der Positionszeiger aktuell steht. Dieser Datensatz muss vorher über die Funktionen UAArchiveMoveNext, UAArchiveMovePrevious, UAArchiveMoveFirst oder UAArchiveMoveLast gelesen werden.

Verbindung zu User Archives abbauen

Mit der Funktion UAArchiveClose wird ein Anwenderarchiv geschlossen. Die Funktion UAREleaseArchive löst die Verbindung zum aktuellen Archiv und die Funktion UADisconnect löst schließlich die Verbindung zur User Archives-Komponente.

2.9.1.3 Abhängigkeiten Aufrufreihenfolge API-Funktionen (RT Professional)

Abhängigkeiten der Aufrufreihenfolge der Runtime-Funktionen

Die User Archives-Funktion UAConnect erstellt den Handle UAHCONNECT. Dieser Handle ist die Voraussetzung für das Öffnen und Schließen von Archiven (keine Views). Es muss also zuerst die Funktion UAConnect aufgerufen werden, um den Handle UAHCONNECT zu erhalten. Mit diesem Handle können dann die untenstehenden API-Funktionen für das Öffnen und Schließen von Archiven (keine Views) aufgerufen werden. Zum Beenden der Konfiguration muss schließlich UADisconnect aufgerufen werden.

Die Funktionen UAQueryArchive und UAQueryArchiveByName erstellen den Handle UAHARCHIVE. Dieser Handle ist Voraussetzung für die User Archives API-Funktion UAArchiveOpen, welche das Archiv für den Runtime-Betrieb öffnet.

Handles für Runtime-Archiv-Funktionen

UAConnect Handle UAHCONNECT
--->

benötigt von UAIsActive

UAUsers

UAArchiveOpen

UAArchiveViews

UAQueryArchive ---> Handle UAHARCHIVE

UAQueryArchiveByName ---> Handle UAHARCHIVE

benötigt von:

UAArchiveOpen ---
>

Voraussetzung für UAArchiveDelete

UAArchiveExport

UAArchiveGetCount

UAArchiveGetFieldLength

UAArchiveGetFields

UAArchiveGetFieldType

UAArchiveGetFieldValueDate

UAArchiveGetFieldValueDouble

UAArchiveGetFieldValueFloat

UAArchiveGetFieldValueLong

UAArchiveGetFieldValueString

UAArchiveGetFieldName

UAArchiveGetFilter

UAArchiveGetID

UAArchiveGetName

UAArchiveGetSort
 UAArchiveImport
 UAArchiveInsert
 UAArchiveMoveFirst
 UAArchiveMoveLast
 UAArchiveMoveNext
 UAArchiveMovePrevious
 UAArchiveReadTagValues
 UAArchiveReadTagValuesByName
 UAArchiveRequery
 UAArchiveSetFieldValueDate
 UAArchiveSetFieldValueDouble
 UAArchiveSetFieldValueFloat
 UAArchiveSetFieldValueLong
 UAArchiveSetFieldValueString
 UAArchiveSetFilter
 UAArchiveSetSort
 UAArchiveUpdate
 UAArchiveWriteTagValues
 UAArchiveWriteTagValuesByName
 <--- UAArchiveClose

<--- UAReleaseArchive

<--- UADisconnect

2.9.1.4 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können mit der API-Funktion UAGetLastError abgefragt werden.

UA_ERROR_SUCCESS	0	Kein Fehler, alles OK.
UA_ERROR_GENERIC	1	Fehler in Konfigurationsfunktion.
UA_ERROR_CONNECT_FAILED	100	Keine Runtime-Verbindung aufgebaut (nicht Runtime).
UA_ERROR_OPEN_FAILED	101	Das Archiv kann nicht geöffnet werden.
UA_ERROR_CLOSE_FAILED	102	Das Archiv kann nicht geschlossen werden, oder Handle ungültig.
UA_ERROR_REQUERY_FAILED	103	Das Wiederaufsetzen auf Archiv mit z. B. neuem Filter / neuer Sortierung konnte nicht durchgeführt werden.

UA_ERROR_MOVE_FAILED	104	Die neue Positionierung First, Last, Next oder Previous konnte nicht durchgeführt werden.
UA_ERROR_INSERT_FAILED	105	Es konnte kein Datensatz an der angegebenen Position eingefügt werden (Position prüfen).
UA_ERROR_UPDATE_FAILED	106	Der angegebene Datensatz konnte nicht upgedatet werden (gesperrt, oder kein Datensatz an dieser Position vorhanden).
UA_ERROR_DELETE_FAILED	107	Der / die Datensätze konnten nicht (alle) gelöscht werden.
UA_ERROR_IMPORT_FAILED	108	Der Import eines Archivs ist misslungen (Format, Größe überprüfen).
UA_ERROR_EXPORT_FAILED	109	Der Export des Anwenderarchivs ist nicht geglückt (Platz prüfen).
UA_ERROR_READ_FAILED	110	Das Lesen aus Variablen konnte nicht durchgeführt werden.
UA_ERROR_WRITE_FAILED	111	Das Schreiben in Variablen konnte nicht durchgeführt werden.
UA_ERROR_GET_FAILED	112	Das Lesen der Archiv- / Feld-Beschreibung, bzw. Datensatz Werten konnte nicht durchgeführt werden.
UA_ERROR_SET_FAILED	113	Das Schreiben der Archiv- / Feld-Beschreibung, bzw. Datensatz Werten konnte nicht durchgeführt werden.
UA_ERROR_INVALID_NAME	200	Kein gültiger Name.
UA_ERROR_INVALID_TYPE	201	Falschen Typ angegeben
UA_ERROR_INVALID_NUMRECS	202	Anzahl der Datensätze kann so nicht angenommen werden
UA_ERROR_INVALID_COMMTYPE	203	Falsche Anbindung None / Raw / Direct gewählt
UA_ERROR_INVALID_LENGTH	204	Die angegebene (String)-Länge wird so nicht angenommen
UA_ERROR_INVALID_PRECISION	205	Die angegebene Anzahl von Nachkommastellen wird so nicht angenommen.
UA_ERROR_NULL_POINTER	1000	Es wurde ein NULL-Pointer übergeben
UA_ERROR_INVALID_POINTER	1001	Der angegebene Pointer ist ungültig
UA_ERROR_INVALID_HANDLE	1002	Das angegebene Handle ist ungültig
UA_ERROR_INVALID_INDEX	1003	Der angegebene Index ist ungültig
UA_ERROR_SERVER_UNKNOWN	1004	Der angegebene, verwendete Server ist ungültig

2.9.1.5 Konstanten (RT Professional)

Konfigurations Konstanten:

Archiv-Typen

UA_ARCHIVETYPE_UNKNOWN	0	Unbekannter Archivtyp
UA_ARCHIVETYPE_UNLIMITED	1	Umlaufarchiv
UA_ARCHIVETYPE_LIMITED	2	Archiv mit fester Datensatz Anzahl

Archiv-Flags

UA_ARCHIVEFLAG_ACCESS	0x00000001	Eine zusätzliche Spalte mit Zeit des letzten Zugriffs wird mit erstellt
UA_ARCHIVEFLAG_USER	0x00000002	Eine zusätzliche Spalte mit Benutzername wird mit erstellt

Kommunikations-Typen

UA_COMMTYPE_NONE	1	Keine Variablenanbindung
UA_COMMTYPE_RAW	2	Kommunikation über Rohdaten Variable
UA_COMMTYPE_DIRECT	3	Kommunikation über DM Variable

Feld-Typen

UA_FIELDTYPE_INTEGER	1	Feldtyp INT
UA_FIELDTYPE_DOUBLE	2	Feldtyp double
UA_FIELDTYPE_FLOAT	3	Feldtyp float
UA_FIELDTYPE_STRING	6	Feldtyp String / ByteArray Nullterminiert
UA_FIELDTYPE_DATETIME	8	Feldtyp für Datum und Uhrzeit

Feld-Flags

UA_FIELDFLAG_UNIQUE	0x00000001	Feld muss einen eindeutigen Wert besitzen
UA_FIELDFLAG_NOTNULL	0x00000002	Feld muss einen Wert besitzen
UA_FIELDFLAG_INDEX	0x00000010	Feld soll durch Index unterstützt werden

Maximale Längen

UA_MAXLEN_NAME	20	Maximale Länge des Archiv- / Feld-Namens
UA_MAXLEN_ALIAS	50	Maximale Länge des 2. Zusatznamens / Kommentars
UA_MAXLEN_PLCID	8	Maximale Länge des Kommunikationsgeräte-Namens
UA_MAXLEN_DMVARNAME	128	Maximale Länge des Datenmanager-Variablen-Namens
UA_MAXLEN_VALUE	255	Maximale Länge für String-Variable

Parameter Konstanten:

Aktionen

UA_ACTION_GENERIC	0	Konfiguration
UA_ACTION_INSERT	1	Einfügen
UA_ACTION_UPDATE	2	Abgleichen / Überschreiben
UA_ACTION_DELETE	3	Löschen

Move-Werte

UA_MOVE_FIRST	0	Auf 1. Datensatz positionieren
UA_MOVE_NEXT	1	Auf nächsten Datensatz positionieren
UA_MOVE_PREVIOUS	-1	Auf vorhergehenden Datensatz positionieren
UA_MOVE_LAST	0x80000000	Auf letzten Datensatz positionieren

Import-, Export-Dateitypen

UA_FILETYPE_DEFAULT	0	Voreingestellt ist der CSV-Dateityp
UA_FILETYPE_CSV	1	(Comma-Seperated-Value) Dateityp

2.9.2 Allgemeine Funktionen (RT Professional)

2.9.2.1 uaGetLastError (RT Professional)

Beschreibung

Viele Funktionen geben einen booleschen Wert zurück. TRUE bedeutet eine fehlerfreie Verarbeitung der Funktion. Wenn der Wert FALSE zurückgegeben wird, kann mit der Funktion uaGetLastError der Fehler der zuletzt verwendeten Funktion ausgelesen werden.

uaGetLastError liefert immer den zuletzt aufgetretenen Fehler zurück. Wenn Sie genau wissen, bei welcher Funktion ein Fehler aufgetreten ist müssen Sie nach jedem Aufruf einer Funktion die Rückgabe auswerten und im Fehlerfall uaGetLastError aufrufen.

Beispiel für Funktionen mit Rückgabewert

```
if ( uaArchiveGetFieldValueLong ( hArchive, Index, &IntValue ) == TRUE )
    printf( "Field Value = %u\n", IntValue );
else
    printf("Error calling uaArchiveGetFieldValueLong: %d / %08lx\n", uaGetLastError(),
uaGetLastHResult());
```

Beispiel für Funktionen ohne Rückgabewert

```

uaArchiveGetFilter(hArchive, pszFilter, cMaxLen);
    INT nUAError = uaGetLastError ( );
    if ( UA_ERROR_SUCCESS != nUAError)
    {
        printf( "Filter = [%s]\n", pszFilter );
    }
    else
    {
        rintf("Error calling uaArchiveGetFilter: %d, hr=0x%08lx\n", nUAError,
uaGetLastHResult());
    }
INT uaGetLastError()

```

Rückgabewert

Fehlerstatus der zuletzt ausgeführten Funktion. Die Fehler-Konstanten und die Predefines sind in CCUACAPI.H festgelegt.

UA_ERROR_SUCCESS	Kein Fehler, alles OK.
UA_ERROR_GENERIC	Fehler in Konfigurationsfunktion.
UA_ERROR_CONNECT_FAILED	Keine Runtime-Verbindung aufgebaut.
UA_ERROR_OPEN_FAILED	Das Archiv kann nicht geöffnet werden.
UA_ERROR_CLOSE_FAILED	Das Archiv kann nicht geschlossen werden, oder Handle ungültig.
UA_ERROR_REQUERY_FAILED	Das Wiederaufsetzen auf Archiv mit z. B. neuem Filter / neuer Sortierung konnte nicht durchgeführt werden.
UA_ERROR_MOVE_FAILED	Die neue Positionierung First, Last, Next oder Previous konnte nicht durchgeführt werden.
UA_ERROR_INSERT_FAILED	Der Datensatz konnte nicht an der angegebenen Position eingefügt werden. Prüfen Sie die Position.
UA_ERROR_UPDATE_FAILED	Der angegebene Datensatz konnte nicht aktualisiert werden. Der Datensatz ist gesperrt oder an dieser Position ist kein Datensatz vorhanden.
UA_ERROR_DELETE_FAILED	Der / die Datensätze konnten nicht (alle) gelöscht werden.
UA_ERROR_IMPORT_FAILED	Der Import eines Archivs war nicht erfolgreich. Überprüfen Sie das Format und die Größe des Archivs.
UA_ERROR_EXPORT_FAILED	Der Export des Anwenderarchivs war nicht erfolgreich. Prüfen Sie den Größe des Speicherplatzes.
UA_ERROR_READ_FAILED	Das Lesen aus Variablen wurde nicht durchgeführt.
UA_ERROR_WRITE_FAILED	Das Schreiben in Variablen wurde nicht durchgeführt.
UA_ERROR_GET_FAILED	Das Lesen der Archiv- / Feld-Beschreibung, bzw. Datensatz Werten wurde nicht durchgeführt.
UA_ERROR_SET_FAILED	Das Schreiben der Archiv- / Feld-Beschreibung, bzw. Datensatz Werten wurde nicht durchgeführt..
UA_ERROR_INVALID_NAME	Kein gültiger Name.
UA_ERROR_INVALID_TYPE	Kein gültiger Typ
UA_ERROR_INVALID_NUMRECS	Anzahl der Datensätze ist ungültig.

UA_ERROR_INVALID_COMMTYPE	Falsche Anbindung None / Raw / Direct gewählt.
UA_ERROR_INVALID_LENGTH	Die angegebene (String)-Länge ist ungültig..
UA_ERROR_INVALID_PRECISION	Die angegebene Anzahl von Nachkommastellen ist ungültig
UA_ERROR_NULL_POINTER	Es wurde ein NULL-Pointer übergeben.
UA_ERROR_INVALID_POINTER	Der angegebene Pointer ist ungültig.
UA_ERROR_INVALID_HANDLE	Das angegebene Handle ist ungültig.
UA_ERROR_INVALID_INDEX	Der angegebene Index ist ungültig
UA_ERROR_SERVER_UNKNOWN	Der angegebene, verwendete Server ist unbekannt.

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

2.9.2.2 uaIsActive (RT Professional)

Beschreibung

Prüft, ob eine Rezeptur in einem aktivierten Runtime verwendet wird.

Deklaration

```
BOOL uaIsActive (
    UAHCONNECT hConnect )
```

Parameter

hConnect

Handle zur Rezeptur in Runtime. Dieser Handle wird mit uaConnect eingerichtet.

Rückgabewert

TRUE

Rezeptur wird in aktiviertem Runtime verwendet.

FALSE

Rezeptur wird nicht in aktiviertem Runtime verwendet.

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
-----------------------	-------------------

Siehe auch

uaConnect (Seite 948)

2.9.2.3 uaUsers (RT Professional)

Beschreibung

Liefert die Anzahl aller Benutzer, die über uaConnect mit der Rezeptur verbunden sind. Hierbei werden neben den Aufrufen durch den Anwender, z.B. durch Skripte, auch WinCC-interne Aufrufe berücksichtigt.

Deklaration

```
LONG uaUsers (  
    UAHCONNECT hConnect )
```

Parameter

hConnect

Handle zur Rezeptur in Runtime. Dieser Handle wird mit uaConnect eingerichtet.

Rückgabewert

Anzahl der aktiven Verbindungen

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
-----------------------	-------------------

Siehe auch

uaConnect (Seite 948)

2.9.2.4 uaSetLocalEvents (RT Professional)

Beschreibung

Setzt die lokalen Ereignisse.

Deklaration

```
void uaSetLocalEvents (  
    UAHCONNECT hConnect  
    BOOL bLocalEvents )
```

Parameter

hConnect

Handle zur Rezeptur in Runtime. Dieser Handle wird mit uaConnect eingerichtet.

bLocalEvents

Lokales Ereignis

Rückgabewert

Kein Rückgabewert

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
-----------------------	-------------------

Siehe auch

uaConnect (Seite 948)

2.9.3 Funktionen zum Verbindungsaufbau (RT Professional)

2.9.3.1 uaConnect (RT Professional)

Beschreibung

Baut in Runtime die Verbindung zu einer Rezeptur auf.

Deklaration

```
BOOL uaConnect (
    UAHCONNECT* phConnect )
```

Parameter

phConnect

Zeiger auf den Handle zur Rezeptur.

Rückgabewert

TRUE

Verbindung zur Rezeptur aufgebaut

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Bemerkungen

Der gelieferte Handle wird für die Funktionen uaDisconnect, ualsActive, uaUsers, uaOpenViews, uaOpenArchives, uaQueryArchive und uaQueryArchiveByName benötigt.

Verwandte Funktionen

uaDisconnect (Seite 949)	Verbindung abbauen
--------------------------	--------------------

Siehe auch

uaIsActive (Seite 945)
uaUsers (Seite 946)
uaSetLocalEvents (Seite 947)
uaDisconnect (Seite 949)
uaQueryArchive (Seite 950)
uaQueryArchiveByName (Seite 952)
uaOpenArchives (Seite 971)
uaOpenViews (Seite 1000)
Übersicht über die Funktionen (Seite 935)

2.9.3.2 uaDisconnect (RT Professional)

Beschreibung

Wenn in Runtime eine Verbindung zu einer Rezeptur besteht, wird die Verbindung abgebaut.

Deklaration

```
BOOL uaDisconnect (  
    UAHCONNECT hConnect )
```

Parameter

hConnect

Handle zur Rezeptur in Runtime. Dieser Handle wird mit uaConnect eingerichtet.

Rückgabewert

TRUE

Verbindung abgebaut.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
-----------------------	-------------------

Siehe auch

uaConnect (Seite 948)
Übersicht über die Funktionen (Seite 935)

2.9.3.3 uaQueryArchive (RT Professional)

Beschreibung

Baut die Verbindung zu einer Rezeptur auf. uaQueryArchive erstellt den Handle phArchive.

Deklaration

```
BOOL uaQueryArchive (  
    UAHCONNECT          hConnect,  
    LONG                lArchive,  
    UAHARCHIVE*        phArchive )
```

Parameter

hConnect

Handle zur Rezeptur in Runtime. Dieser Handle wird mit uaConnect eingerichtet.

lArchive

ID des Archivs, das verbunden werden soll.

phArchive

Zeiger auf den Handle zur verbundenen Rezeptur.

Rückgabewert

TRUE

Verbindung aufgebaut.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Bemerkungen

Steht nur ein Archiv-Index zur Verfügung, kann die ID des Archivs mit der Konfigurationsfunktion `uaGetArchive` ermittelt werden.

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
---------------------------------------	-------------------

Siehe auch

[uaConnect \(Seite 948\)](#)
[uaReleaseArchive \(Seite 955\)](#)
[uaArchiveClose \(Seite 958\)](#)
[uaArchiveDelete \(Seite 959\)](#)
[uaArchiveExport \(Seite 960\)](#)
[uaArchiveGetID \(Seite 962\)](#)
[uaArchiveGetName \(Seite 963\)](#)
[uaArchiveGetSort \(Seite 964\)](#)
[uaArchiveImport \(Seite 965\)](#)
[uaArchiveOpen \(Seite 967\)](#)
[uaArchiveUpdate \(Seite 968\)](#)
[uaArchiveGetCount \(Seite 972\)](#)
[uaArchiveGetFilter \(Seite 973\)](#)
[uaArchiveInsert \(Seite 974\)](#)
[uaArchiveMoveFirst \(Seite 975\)](#)
[uaArchiveMoveLast \(Seite 976\)](#)
[uaArchiveMoveNext \(Seite 977\)](#)
[uaArchiveMovePrevious \(Seite 978\)](#)
[uaArchiveGetFieldLength \(Seite 980\)](#)

- uaArchiveGetFields (Seite 983)
- uaArchiveGetFieldType (Seite 984)
- uaArchiveGetFieldName (Seite 981)
- uaArchiveGetFieldValueDate (Seite 985)
- uaArchiveGetFieldValueDouble (Seite 986)
- uaArchiveGetFieldValueLong (Seite 987)
- uaArchiveSetFieldValueDate (Seite 990)
- uaArchiveSetFieldValueDouble (Seite 991)
- uaArchiveSetFieldValueLong (Seite 992)
- uaArchiveSetFieldValueString (Seite 993)
- uaArchiveRequery (Seite 996)
- uaArchiveSetFilter (Seite 997)
- uaArchiveSetSort (Seite 999)
- uaArchiveReadTagValues (Seite 1001)
- uaArchiveReadTagValuesByName (Seite 1002)
- uaArchiveWriteTagValues (Seite 1003)
- uaArchiveWriteTagValuesByName (Seite 1005)
- Übersicht über die Funktionen (Seite 935)

2.9.3.4 uaQueryArchiveByName (RT Professional)

Beschreibung

Baut die Verbindung zu einer Rezeptur über den Namen auf. uaQueryArchiveByName erstellt den Handle phArchive.

Deklaration

```
BOOL uaQueryArchiveByName (
    UAHCONNECT      hConnect,
    LPCSTR          pszName,
    UAHARCHIVE*     phArchive )
```

Parameter

hConnect

Handle zur Rezeptur in Runtime. Dieser Handle wird mit uaConnect eingerichtet.

pszName

Name der Rezeptur.

phArchive

Zeiger auf den Handle zur verbundenen Rezeptur.

Rückgabewert**TRUE**

Verbindung aufgebaut.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Bemerkungen

Steht nur ein Archiv-Index zur Verfügung, kann der Name des Archivs mit der Konfigurationsfunktion `uaGetArchive` ermittelt werden.

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
---------------------------------------	-------------------

Siehe auch

[uaConnect \(Seite 948\)](#)

[uaReleaseArchive \(Seite 955\)](#)

[uaArchiveClose \(Seite 958\)](#)

[uaArchiveDelete \(Seite 959\)](#)

[uaArchiveExport \(Seite 960\)](#)

[uaArchiveGetID \(Seite 962\)](#)

[uaArchiveGetName \(Seite 963\)](#)

[uaArchiveGetSort \(Seite 964\)](#)

[uaArchiveImport \(Seite 965\)](#)

[uaArchiveOpen \(Seite 967\)](#)

- uaArchiveUpdate (Seite 968)
- uaArchiveGetCount (Seite 972)
- uaArchiveGetFilter (Seite 973)
- uaArchiveInsert (Seite 974)
- uaArchiveMoveFirst (Seite 975)
- uaArchiveMoveLast (Seite 976)
- uaArchiveMoveNext (Seite 977)
- uaArchiveMovePrevious (Seite 978)
- uaArchiveGetFieldLength (Seite 980)
- uaArchiveGetFields (Seite 983)
- uaArchiveGetFieldType (Seite 984)
- uaArchiveGetFieldName (Seite 981)
- uaArchiveGetFieldValueDate (Seite 985)
- uaArchiveGetFieldValueDouble (Seite 986)
- uaArchiveGetFieldValueLong (Seite 987)
- uaArchiveSetFieldValueDate (Seite 990)
- uaArchiveSetFieldValueDouble (Seite 991)
- uaArchiveSetFieldValueLong (Seite 992)
- uaArchiveSetFieldValueString (Seite 993)
- uaArchiveRequery (Seite 996)
- uaArchiveSetFilter (Seite 997)
- uaArchiveSetSort (Seite 999)
- uaArchiveReadTagValues (Seite 1001)
- uaArchiveReadTagValuesByName (Seite 1002)
- uaArchiveWriteTagValues (Seite 1003)
- uaArchiveWriteTagValuesByName (Seite 1005)
- Übersicht über die Funktionen (Seite 935)

2.9.3.5 uaQueryConfiguration (RT Professional)

Beschreibung

Baut die Verbindung zu einer Rezeptur für die Konfiguration auf.

Deklaration

```
BOOL uaQueryConfiguration (  
    UAHCONFIG*          phConfig )
```

Parameter

phConfig

Zeiger auf den Handel der Rezeptur.

Rückgabewert

TRUE

Verbindung aufgebaut.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Siehe auch

uaReleaseConfiguration (Seite 957)

uaGetArchive (Seite 969)

uaGetNumArchives (Seite 970)

uaGetNumFields (Seite 979)

uaGetField (Seite 995)

Übersicht über die Funktionen (Seite 935)

2.9.3.6 uaReleaseArchive (RT Professional)

Beschreibung

Baut die Verbindung zur aktuell verbundenen Rezeptur ab.

Deklaration

```
BOOL uaReleaseArchive (  
    UAHARCHIVE      hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Verbindung abgebaut.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Bemerkung

Beim erfolgreichen Abbau der Verbindung müssen Sie den Handle hArchive auf NULL setzen. Bei einer weiteren Verwendung des nicht mehr gültige Handles wird dann der Fehler UA_ERROR_INVALID_HANDLE generiert. Eine unnötige Speicherbelastung wird vermieden.

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.3.7 uaReleaseConfiguration (RT Professional)

Beschreibung

Baut die Verbindung nach der Konfiguration der Rezeptur ab.

Deklaration

```
BOOL uaReleaseConfiguration (
    UAHCONFIG      hConfig,
    BOOL          bSave )
```

Parameter

hConfig

Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

bSave

Legt fest, dass die Änderungen an der Konfiguration vor dem Abbau der Verbindung gesichert werden.

TRUE	Änderungen sichern
FALSE	Änderungen verwerfen

Hinweis

bSave = TRUE dürfen Sie nur verwenden, wenn die Rezeptur nicht in einem aktiviertem WinCC Runtime verwendet wird! Mit der Funktion ualsActive prüfen Sie, ob Runtime aktiv ist.

Rückgabewert

TRUE

Verbindung abgebaut.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryConfiguration (Seite 954)	Handle einrichten
----------------------------------	-------------------

Siehe auch

uaQueryConfiguration (Seite 954)

Übersicht über die Funktionen (Seite 935)

2.9.4 Funktionen zum Bearbeiten von Rezepturen (RT Professional)

2.9.4.1 uaArchiveClose (RT Professional)

Beschreibung

Schließt eine geöffnete Rezeptur.

Deklaration

```
BOOL uaArchiveClose (  
    UAHARCHIVE      hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Rezeptur geschlossen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten
uaArchiveOpen (Seite 967)	Rezeptur öffnen

Siehe auch

uaQueryArchive (Seite 950)
 uaQueryArchiveByName (Seite 952)
 uaArchiveOpen (Seite 967)
 Übersicht über die Funktionen (Seite 935)

2.9.4.2 uaArchiveDelete (RT Professional)

Beschreibung

Löscht Daten aus einer Rezeptur. Die projizierte Rezeptur bleibt erhalten.

Deklaration

```

BOOL uaArchiveDelete (
    UAHARCHIVE    hArchive,
    LPCSTR        pszWhere )
  
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszWhere

String mit der SQL-Selektion der zu löschenden Datensätze. Der String entspricht der SQL-Anweisung DELETE FROM <archive> WHERE pszWhere.

Hinweis

Wenn pszWhere leer ist, wird das gesamte Anwenderarchiv gelöscht.

Rückgabewert

TRUE

Daten gelöscht.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.4.3 uaArchiveExport (RT Professional)

Beschreibung

Exportiert die Daten einer Rezeptur in eine CSV-Datei.

Deklaration

```
BOOL uaArchiveExport (
    UAHARCHIVE    hArchive,
    LPCSTR        pszDestination,
    LONG          lType,
    LONG          lOptions )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszDestination

Name der Zielfeile. Beim Aufruf der Funktion auf einem Client bezieht sich die Pfadangabe auf den Server.

IType

Datenformat der Zielfeile. Zwei Formate stehen zur Verfügung:

UA_FILETYPE_DEFAULT = 0	Das voreingestellte Dateiformat CSV
UA_FILETYPE_CSV = 1	CSV-Dateiformat

IOptions

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

Rückgabewert

TRUE

Daten exportiert.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten
uaArchiveImport (Seite 965)	Daten importieren

Siehe auch

uaArchiveImport (Seite 965)

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)
Übersicht über die Funktionen (Seite 935)

2.9.4.4 uaArchiveGetID (RT Professional)

Beschreibung

Liest die ID einer Rezeptur

Returnwert

"ID" des Anwenderarchivs

Deklaration

```
LONG uaArchiveGetID (  
    UAHARCHIVE      hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

ID der Rezeptur

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
uaQueryArchiveByName (Seite 952)
Übersicht über die Funktionen (Seite 935)

2.9.4.5 uaArchiveGetName (RT Professional)**Beschreibung**

Liest den Namen einer Rezeptur.

Deklaration

```
VOID uaArchiveGetName (  
    UAHARCHIVE      hArchive,  
    LPSTR           pszName,  
    LONG            cMaxLen )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszName

Zeiger auf den Puffer für den Namen der Rezeptur.

cMaxLen

Maximale Länge.

Rückgabewert

Kein Rückgabewert.

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Beispiel

```
char tank [40];  
uaArchiveGetName( hArchive, tank, 39 );
```

Siehe auch

- uaQueryArchive (Seite 950)
- uaQueryArchiveByName (Seite 952)
- Übersicht über die Funktionen (Seite 935)

2.9.4.6 uaArchiveGetSort (RT Professional)

Beschreibung

Liest das Sortierkriterium der Rezeptur.

Deklaration

```
VOID uaArchiveGetSort (  
    UAHARCHIVE    hArchive,  
    LPSTR         pszSort,  
    LONG          cMaxLen )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszSort

Sortierkriterium als SQL-Anweisung.

cMaxLen

Maximale Länge.

Rückgabewert

Kein Rückgabewert

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.4.7 uaArchiveImport (RT Professional)**Beschreibung**

Importiert Daten aus einer CSV-Datei in eine Rezeptur. Die Struktur der Rezeptur muss identisch sein mit dem importierten CSV-Archiv.

Deklaration

```

BOOL uaArchiveImport (
    UAHARCHIVE      hArchive,
    LPCSTR          pszSource,
    LONG            lType,
    LONG            lOptions )

```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszSource

Name der Datei mit den zu importierenden Daten.

IType

Datenformat der Quelldatei. Zwei Formate stehen zur Verfügung:

UA_FILETYPE_DEFAULT = 0	Das voreingestellte Dateiformat CSV
UA_FILETYPE_CSV = 1	CSV-Dateiformat

IOptions

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

Rückgabewert

TRUE

Daten importiert.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten
uaArchiveExport (Seite 960)	Daten exportieren

Siehe auch

uaArchiveExport (Seite 960)

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.4.8 uaArchiveOpen (RT Professional)

Beschreibung

Öffnet eine Rezeptur. Der Aufruf von uaArchiveOpen ist für Lese- oder Schreiboperationen in der Rezeptur notwendig. Zum Beispiel vor Aufruf der Funktionen uaArchiveMoveNext, uaArchiveDelete oder uaArchiveUpdate.

Deklaration

```
BOOL uaArchiveOpen (
    UAHARCHIVE    hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Rezeptur geöffnet.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten
uaArchiveClose (Seite 958)	Rezeptur schließen

Siehe auch

- uaArchiveClose (Seite 958)
- uaQueryArchive (Seite 950)
- uaQueryArchiveByName (Seite 952)
- Übersicht über die Funktionen (Seite 935)

2.9.4.9 uaArchiveUpdate (RT Professional)

Beschreibung

Aktualisiert eine geöffnete Rezeptur. Alle geänderten Rezepturdaten werden in die Datenbank übernommen. Die Struktur der Rezeptur bleibt unverändert.

Deklaration

```
BOOL uaArchiveUpdate (  
U    AHARCHIVE hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Rezepturdaten aktualisiert.

FALSE

Bei einer Konsistenzverletzung tritt der Fehler Update_failed = 106 auf. Eine derartige Konsistenzverletzung liegt z. B. vor, wenn für ein Feld die Eigenschaft "Feld muss einen Wert besitzen gesetzt ist, aber im Feld kein Wert vorhanden ist.

Benötigte Dateien

- ccuacapi.h
- ccuacapi.lib
- ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
 uaQueryArchiveByName (Seite 952)
 Übersicht über die Funktionen (Seite 935)

2.9.4.10 uaGetArchive (RT Professional)

Beschreibung

Liest die Konfiguration einer Rezeptur.

Deklaration

```

BOOL uaGetArchive (
    UAHCONFIG          hConfig,
    Long               lArchive,
    UACONFIGARCHIVE*  pArchive )
  
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

lArchive

Index der Rezeptur. Wert: 0... uaGetNumArchives() -1

pArchive

Zeiger auf Puffer für die Konfigurationsdaten.

Rückgabewert

TRUE

Zugriff erfolgreich.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaQueryConfiguration (Seite 954)	Handle einrichten
----------------------------------	-------------------

Siehe auch

uaQueryConfiguration (Seite 954)
Übersicht über die Funktionen (Seite 935)

2.9.4.11 uaGetNumArchives (RT Professional)

Beschreibung

Liest die Anzahl der momentan konfigurierten Rezepturen.

Deklaration

```
LONG uaGetNumArchives (  
    UAHCONFIG          hConfig )
```

Parameter

hConfig
Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

Rückgabewert

Anzahl der konfigurierten Rezepturen. Im Fehlerfall wird -1 zurückgeliefert.

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaQueryConfiguration (Seite 954)	Handle einrichten
----------------------------------	-------------------

Siehe auch

uaQueryConfiguration (Seite 954)
Übersicht über die Funktionen (Seite 935)

2.9.4.12 uaOpenArchives (RT Professional)**Beschreibung**

Ermittelt die Anzahl der in Runtime geöffneten Rezepturen.

Deklaration

```
LONG uaOpenArchives (
    UAHCONNECT hConnect )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaConnect eingerichtet.

Rückgabewert

Anzahl der geöffneten Rezepturen.

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
-----------------------	-------------------

Siehe auch

uaConnect (Seite 948)

Übersicht über die Funktionen (Seite 935)

2.9.5 Funktionen zum Bearbeiten von Rezepturelementen (RT Professional)

2.9.5.1 uaArchiveGetCount (RT Professional)

Beschreibung

Liest die Anzahl der Datensätze.

Returnwert

Anzahl der Datensätze. Bei "0" ist das Archiv leer oder ein Fehler ist aufgetreten. Die Abfrage mit "uaGetLastError" ist erforderlich.

Deklaration

```
LONG uaArchiveGetCount(  
    UAHARCHIVE      hArchive,  
    LONG*           plCount )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

plCount

Zeiger auf eine Variable, in der die Anzahl der Datensätze abgelegt wird.

Rückgabewert

Anzahl der Datensätze. Bei 0 ist das Archiv leer oder ein Fehler ist aufgetreten. Die Abfrage mit uaGetLastError ist erforderlich.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h
 ccuacapi.lib
 ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
 uaQueryArchiveByName (Seite 952)
 Übersicht über die Funktionen (Seite 935)

2.9.5.2 uaArchiveGetFilter (RT Professional)**Beschreibung**

Liest die Selektionskriterien des aktuellen Datensatzes.

Deklaration

```
VOID uaArchiveGetFilter (
    UAHARCHIVE      hArchive,
    LPSTR           pszFilter,
    LONG            cMaxLen )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszFilter

Selektionskriterium als SQL-Anweisung.

cMaxLen

Maximale Länge.

Rückgabewert

Kein Rückgabewert.

Benötigte Dateien

- ccuacapi.h
- ccuacapi.lib
- ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

- uaQueryArchive (Seite 950)
- uaQueryArchiveByName (Seite 952)
- Übersicht über die Funktionen (Seite 935)

2.9.5.3 uaArchiveInsert (RT Professional)

Beschreibung

Fügt den lokalen Datensatzpuffer in die Rezeptur ein. Damit sinnvolle Daten im neuen Datensatz stehen, müssen Sie die Felder des lokalen Datensatzpuffers vor dem Aufruf von uaArchiveInsert mit den Funktionen uaArchiveSetFieldValue... beschreiben.

Das Feld "ID" müssen Sie mit der Funktion uaArchiveSetFieldValueLong mit einer im Archiv noch nicht vorhandenen ID oder mit 0 beschreiben."

Deklaration

```
BOOL uaArchiveInsert (
    UAHARCHIVE    hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert**TRUE**

Datensatz eingefügt.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.5.4 uaArchiveMoveFirst (RT Professional)**Beschreibung**

Geht zum ersten Datensatz.

Deklaration

```

BOOL uaArchiveMoveFirst (
    UAHARCHIVE      hArchive )

```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Sprung erfolgreich.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.5.5 uaArchiveMoveLast (RT Professional)

Beschreibung

Geht zum letzten Datensatz.

Deklaration

```
BOOL uaArchiveMoveLast (
    UAHARCHIVE hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert**TRUE**

Sprung erfolgreich.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.5.6 uaArchiveMoveNext (RT Professional)**Beschreibung**

Geht zum nächsten Datensatz.

Deklaration

```

BOOL uaArchiveMoveNext (
    UAHARCHIVE      hArchive )

```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

TRUE

Sprung erfolgreich.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.5.7 uaArchiveMovePrevious (RT Professional)

Beschreibung

Geht zum vorherigen Datensatz.

Deklaration

```
BOOL uaArchiveMovePrevious (  
    UAHARCHIVE      hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert**TRUE**

Sprung erfolgreich.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.5.8 uaGetNumFields (RT Professional)**Beschreibung**

Liefert die Anzahl der projizierten Felder. Die Felder "ID", "Letzter Benutzer" und "Letzter Zugriff" werden nicht mitgezählt. In den Konfigurationsaufrufen werden die Indizes mit "0 ... uaGetNumFields() -1" angegeben.

Deklaration

```
LONG uaGetNumFields (
    UAHCONFIG      honfig,
    long           lArchive )
```

Parameter

hConfig

Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

lArchive

Index der Rezeptur. Wert: 0... uaGetNumArchives() -1

Rückgabewert

Anzahl der projektierten Felder. Im Fehlerfall wird -1 zurückgeliefert.

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryConfiguration (Seite 954)	Handle einrichten
--	-------------------

Siehe auch

[uaQueryConfiguration \(Seite 954\)](#)

[Übersicht über die Funktionen \(Seite 935\)](#)

2.9.6 Funktionen zum Bearbeiten von Feldern (RT Professional)

2.9.6.1 uaArchiveGetFieldLength (RT Professional)

Beschreibung

Liest die Länge eines Feldes im aktuellen Datensatz.

Deklaration

```
LONG uaArchiveGetFieldLength(  
    UAHARCHIVE    hArchive,  
    LONG          lField )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

IField

Die Feldnummer, wobei iField = 1 das erste projizierte Feld adressiert. Mit IField = 0 wird das Feld "ID" adressiert.

Rückgabewert

Länge des aktuellen Feldes.

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.6.2 uaArchiveGetFieldName (RT Professional)**Beschreibung**

Liest den Namen eines Feldes im aktuellen Datensatz.

Deklaration

```
VOID uaArchiveGetFieldName (  
    UAHARCHIVE    hArchive,  
    LONG          lField,  
    LPCSTR        pszName,  
    LONG          cMaxLen )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

pszName

Feldname

cMaxLen

Maximale Länge

Rückgabewert

TRUE

Name gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

AUTOHOTSPOT	Handle einrichten
uaQueryArchive (Seite 950)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
 uaQueryArchiveByName (Seite 952)
 Übersicht über die Funktionen (Seite 935)

2.9.6.3 uaArchiveGetFields (RT Professional)**Beschreibung**

Liest die Anzahl der projektorientierten Datenfelder. Die Felder "ID", "Letzter Benutzer" und "Letzter Zugriff" werden mitgezählt. In den Aufrufen in Runtime werden die Indizes der projektorientierten Felder mit 1 ... N angegeben. Das Feld "ID" hat den Index "0". Die Felder "Letzter Benutzer" und "Letzter Zugriff" werden am Ende der projektorientierten Felder angehängt.

Deklaration

```
LONG uaArchiveGetFields (
    UAHARCHIVE          hArchive )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert

Anzahl der projektorientierten Felder.

Benötigte Dateien

ccuacapi.h
 ccuacapi.lib
 ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

- uaQueryArchive (Seite 950)
- uaQueryArchiveByName (Seite 952)
- Übersicht über die Funktionen (Seite 935)

2.9.6.4 uaArchiveGetFieldType (RT Professional)

Beschreibung

Liest den Typ eines Feldes im aktuellen Datensatz.

Deklaration

```
LONG uaArchiveGetFieldType (
    UAHARCHIVE    hArchive,
    LONG          lField )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

Rückgabewert

Typ des aktuellen Felds. Die symbolischen Bezeichnungen für die Feldtypen sind:

UA_FIELDTYPE_INTEGER	
UA_FIELDTYPE_FLOAT	
UA_FIELDTYPE_DOUBLE	
UA_FIELDTYPE_STRING	
UA_FIELDTYPE_DATETIME	

Benötigte Dateien

- ccuacapi.h
- ccuacapi.lib
- ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
uaQueryArchiveByName (Seite 952)
Übersicht über die Funktionen (Seite 935)

2.9.6.5 uaArchiveGetFieldValueDate (RT Professional)

Beschreibung

Liest Datum und Uhrzeit eines Feldes im aktuellen Datensatz.

Deklaration

```

BOOL uaArchiveGetFieldValueDate (
    UAHARCHIVE      hArchive,
    LONG            lField,
    LPSYSTEMTIME    pstDateTime )

```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

pstDateTime

Zeiger auf eine Variable vom Typ SYSTEMTIME.

Rückgabewert

TRUE

Datum und Uhrzeit gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.6.6 uaArchiveGetFieldValueDouble (RT Professional)

Beschreibung

Liest einen Wert vom Datentyp Double aus einem Felds im aktuellen Datensatz.

Deklaration

```
BOOL uaArchiveGetFieldValueDouble (
    UAHARCHIVE    hArchive,
    LONG          lField,
    double*       pdValue )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

pdValue

Zeiger auf die Variable für den Wert des Feldes.

Rückgabewert**TRUE**

Wert gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.6.7 uaArchiveGetFieldValueLong (RT Professional)**Beschreibung**

Liest einen Wert vom Datentyp Long aus einem Felds im aktuellen Datensatz.

Deklaration

```

BOOL uaArchiveGetFieldValueLong (
    UAHARCHIVE      hArchive,
    LONG             lField,
    LONG*            pdValue )

```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

IField

Die Feldnummer, wobei iField = 1 das erste projizierte Feld adressiert. Mit IField = 0 wird das Feld "ID" adressiert.

pdValue

Zeiger auf die Variable für den Wert des Feldes.

Rückgabewert

TRUE

Wert gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.6.8 uaArchiveGetFieldValueString (RT Professional)

Beschreibung

Liest einen Wert vom Datentyp String aus einem Feld im aktuellen Datensatz.

Deklaration

```

BOOL uaArchiveGetFieldValueString (
    UAHARCHIVE      hArchive,
    LONG            lField,
    LPSTR           pszString,
    LONG            cMaxLen )

```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

pszString

Zeiger auf die Variable für den Wert des Feldes.

pdValue

Maximale Länge des Strings.

Rückgabewert

TRUE

Wert gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

- uaQueryArchive (Seite 950)
- uaQueryArchiveByName (Seite 952)
- Übersicht über die Funktionen (Seite 935)

2.9.6.9 uaArchiveSetFieldValueDate (RT Professional)

Beschreibung

Schreibt Datum und Uhrzeit in ein Feld im aktuellen Datensatz.

Deklaration

```
BOOL uaArchiveSetFieldValueDate (  
    UAHARCHIVE      hArchive,  
    LONG            lField,  
    LPSYSTEMTIME    pstDateTime )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

pstDateTime

Datum und Uhrzeit.

Rückgabewert

TRUE

Wert geschrieben.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.6.10 uaArchiveSetFieldValueDouble (RT Professional)**Beschreibung**

Schreibt einen Wert vom Datentyp Double in ein Feld im aktuellen Datensatz.

Deklaration

```

BOOL uaArchiveSetFieldValueDouble (
    UAHARCHIVE    hArchive,
    LONG          lField,
    double         dValue )

```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

dValue

Zu schreibender Wert.

Rückgabewert

TRUE

Wert geschrieben.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.6.11 uaArchiveSetFieldValueLong (RT Professional)

Beschreibung

Schreibt einen Wert vom Datentyp Long in ein Feld im aktuellen Datensatz.

Deklaration

```
BOOL uaArchiveSetFieldValueLong (
    UAHARCHIVE    hArchive,
    LONG          lField,
    LONG          dValue )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit `uaQueryArchive` oder `uaQueryArchiveByName` eingerichtet.

IField

Die Feldnummer, wobei `iField = 1` das erste projizierte Feld adressiert. Mit `IField = 0` wird das Feld "ID" adressiert.

dValue

Zu schreibender Wert.

Rückgabewert**TRUE**

Wert geschrieben.

FALSE

Fehler

Benötigte Dateien

`ccuacapi.h`

`ccuacapi.lib`

`ccuacapi.dll`

Verwandte Funktionen

<code>uaQueryArchive</code> (Seite 950)	Handle einrichten
<code>uaQueryArchiveByName</code> (Seite 952)	Handle einrichten

Siehe auch

`uaQueryArchive` (Seite 950)

`uaQueryArchiveByName` (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.6.12 uaArchiveSetFieldValueString (RT Professional)**Beschreibung**

Schreibt einen Wert vom Datentyp String in ein Feld im aktuellen Datensatz.

Deklaration

```
BOOL uaArchiveSetFieldValueString (
    UAHARCHIVE    hArchive,
    LONG          lField,
    LPCSTR        pszString )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

pszString

Zu schreibende Zeichenkette.

Rückgabewert

TRUE

Wert geschrieben.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
uaQueryArchiveByName (Seite 952)
Übersicht über die Funktionen (Seite 935)

2.9.6.13 uaGetField (RT Professional)**Beschreibung**

Liest die Konfiguration eines Feldes.

Deklaration

```
BOOL uaGetField (
    UAHCONFIG          hConfig,
    Long               lArchive,
    Long               lField,
    UACONFIGFIELD*    pField )
```

Parameter**hConfig**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryConfiguration eingerichtet.

lArchive

Index der Rezeptur. Wert: 0... uaGetNumArchives() -1

lField

Die Feldnummer, wobei lField = 1 das erste projizierte Feld adressiert. Mit lField = 0 wird das Feld "ID" adressiert.

pField

Zeiger auf den Puffer für die Konfigurationsdaten.

Rückgabewert**TRUE**

Konfiguration gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaQueryConfiguration (Seite 954)	Handle einrichten
----------------------------------	-------------------

Siehe auch

uaQueryConfiguration (Seite 954)
Übersicht über die Funktionen (Seite 935)

2.9.7 Funktionen zum Filtern und Sortieren (RT Professional)

2.9.7.1 uaArchiveRequery (RT Professional)

Beschreibung

Nach dem Aufruf von uaArchiveSetFilter und uaArchiveSetSort müssen Sie die Rezeptur mit uaArchiveRequery neu laden.

Hinweis

Die Funktionen uaArchiveSetSort und uaArchiveSetFilter können Sie verwenden, ohne die Rezeptur mit uaArchiveOpen zu öffnen. In diesem Fall müssen Sie uaArchiveRequery nicht aufrufen.

Deklaration

```
BOOL uaArchiveRequery(  
    UAHARCHIVE hArchive )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

Rückgabewert**TRUE**

Rezeptur neu geladen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Bemerkung

Rufen Sie `uaArchiveRequery` auch auf, wenn Sie in Runtime Werte in die Rezepturanzeige eingegeben haben.

Verwandte Funktionen

<code>uaQueryArchive</code> (Seite 950)	Handle einrichten
<code>uaQueryArchiveByName</code> (Seite 952)	Handle einrichten

Siehe auch

`uaQueryArchive` (Seite 950)

`uaQueryArchiveByName` (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.7.2 uaArchiveSetFilter (RT Professional)**Beschreibung**

Setzt die Selektionskriterien für die Rezeptur. Sie können die Funktion auch aufrufen, ohne die Rezeptur mit `uaArchiveOpen` zu öffnen.

Wenn Sie das Anwenderarchiv mit `uaArchiveOpen` geöffnet haben, müssen Sie nach dem Filtern die Rezeptur mit `uaArchiveRequery` neu laden.

Deklaration

```
VOID uaArchiveSetFilter (
    UAHARCHIVE    hArchive,
    LPSTR         pszFilter )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszFilter

Selektionskriterium als SQL-Kriterium.

Rückgabewert

TRUE

Selektionskriterien geändert.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

Übersicht über die Funktionen (Seite 935)

2.9.7.3 uaArchiveSetSort (RT Professional)

Beschreibung

Setzt die Sortierung einer Rezeptur. Sie können die Funktion auch aufrufen, ohne die Rezeptur mit uaArchiveOpen zu öffnen.

Wenn Sie die Rezeptur mit uaArchiveOpen geöffnet haben, müssen Sie nach dem Sortieren die Rezeptur mit uaArchiveRequery neu laden.

Deklaration

```
BOOL uaArchiveSetSort (
    UAHARCHIVE      hArchive,
    LPSTR           pszSort )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszSort

Sortierkriterium als SQL-Anweisung.

Rückgabewert

TRUE

Sortieren erfolgreich.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

- uaQueryArchive (Seite 950)
- uaQueryArchiveByName (Seite 952)
- Übersicht über die Funktionen (Seite 935)

2.9.8 Funktionen zum Bearbeiten von Rezeptursichten (RT Professional)

2.9.8.1 uaOpenViews (RT Professional)

Beschreibung

Liefert die Anzahl der in Runtime geöffneten Sichten.

Deklaration

```
LONG uaOpenViews (
    UAHCONNECT hConnect )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaConnect eingerichtet.

Rückgabewert

Anzahl der aktuell geöffneten Sichten.

Benötigte Dateien

- ccuacapi.h
- ccuacapi.lib
- ccuacapi.dll

Verwandte Funktionen

uaConnect (Seite 948)	Handle einrichten
-----------------------	-------------------

Siehe auch

uaConnect (Seite 948)

Übersicht über die Funktionen (Seite 935)

2.9.9 Funktionen zum Bearbeiten von Variablen (RT Professional)**2.9.9.1 uaArchiveReadTagValues (RT Professional)****Beschreibung**

Liest die aktuellen Werte aus den Feldvariablen.

Deklaration

```
BOOL uaArchiveReadTagValues (
    UAHARCHIVE    hArchive,
    LONG*         pnFields,
    LONG          cFields,
    LONG          lOptions )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pnFields

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

cFields

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

lOptions

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

Rückgabewert**TRUE**

Daten gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaArchiveWriteTagValues (Seite 1003)	Werte schreiben
uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

uaArchiveWriteTagValues (Seite 1003)

Übersicht über die Funktionen (Seite 935)

2.9.9.2 uaArchiveReadTagValuesByName (RT Professional)

Beschreibung

Liest die aktuellen Werte aus den Variablen.

Deklaration

```
BOOL uaArchiveReadTagValuesByName (
    UAHARCHIVE    hArchive,
    LPCSTR        pszFields,
    LONG          lOptions )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszFields

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

IOptions

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

Rückgabewert**TRUE**

Daten gelesen.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib

ccuacapi.dll

Verwandte Funktionen

uaArchiveWriteTagValuesByName (Seite 1005)	Werte lesen
uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)

uaQueryArchiveByName (Seite 952)

uaArchiveWriteTagValuesByName (Seite 1005)

Übersicht über die Funktionen (Seite 935)

2.9.9.3 uaArchiveWriteTagValues (RT Professional)**Beschreibung**

Schreibt die Werte des aktuellen Datensatzes in die Variable.

Deklaration

```
BOOL uaArchiveWriteTagValues (
    UAHARCHIVE      hArchive,
    LONG*           pnFields,
    LONG            cFields,
    LONG            lOptions )
```

Parameter

hArchive

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pnFields

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

cFields

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

lOptions

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

Rückgabewert

TRUE

Daten geschrieben.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h
ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaArchiveReadTagValues (Seite 1001)	Werte lesen
uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
uaQueryArchiveByName (Seite 952)
uaArchiveReadTagValues (Seite 1001)
Übersicht über die Funktionen (Seite 935)

2.9.9.4 uaArchiveWriteTagValuesByName (RT Professional)**Beschreibung**

Schreibt die Werte des aktuellen Datensatzes in die Variable.

Deklaration

```
BOOL uaArchiveWriteTagValuesByName (  
    UAHARCHIVE      hArchive,  
    LPCSTR          pszFields,  
    LONG            lOptions )
```

Parameter**hArchive**

Handle zur Rezeptur. Dieser Handle wird mit uaQueryArchive oder uaQueryArchiveByName eingerichtet.

pszFields

Der Parameter ist für spätere Erweiterungen reserviert und muss mit NULL vorbelegt werden.

lOptions

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbelegt werden.

Rückgabewert**TRUE**

Daten geschrieben.

FALSE

Fehler

Benötigte Dateien

ccuacapi.h

ccuacapi.lib
ccuacapi.dll

Verwandte Funktionen

uaArchiveReadTagValuesByName (Seite 1002)	Werte lesen
uaQueryArchive (Seite 950)	Handle einrichten
uaQueryArchiveByName (Seite 952)	Handle einrichten

Siehe auch

uaQueryArchive (Seite 950)
uaQueryArchiveByName (Seite 952)
uaArchiveReadTagValuesByName (Seite 1002)
Übersicht über die Funktionen (Seite 935)

2.10 Funktionen für Meldungen (RT Professional)

2.10.1 Grundlagen (RT Professional)

2.10.1.1 Übersicht über die Funktionen (RT Professional)

Übersicht

MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Sende- und Empfangsservice (Callback)
MSRTActivateMProtPlus (Seite 1059)	Protokollservice anhalten/fortsetzen
MSRTCheckWinFilterPlus (Seite 1095)	Meldungsfilterbedingungen überprüfen
MSRTCreateMsgInstancePlus (Seite 1066)	Meldung erzeugen
MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)	Meldung erzeugen
AUTOHOTSPOT	Kommentartext parametrieren
AUTOHOTSPOT	Meldungsfilter parametrieren
MSRTDialogMsgLockPlus (Seite 1068)	Meldesperren parametrieren
MSRTEnumArchivDataPlus (Seite 1047)	Meldungen eines Archives auflisten
MSRTEnumBackupListPlus (Seite 1108)	Auslagerungsliste Einträge auflisten
MSRTEnumGroupMsgPlus (Seite 1089)	Meldungsgruppe Einzelmeldungen auflisten
MSRTEnumProtDataPlus (Seite 1060)	Protokolldaten auflisten
MSRTExportPlus (Seite 1109)	Meldungen aus Folgearchiven exportieren

MSRTGetClassInfoPlus (Seite 1073)	Meldeklassen-Information abfragen
MSRTGetCommentInstancePlus (Seite 1102)	Kommentartext abfragen
MSRTGetFilterDataPlus (Seite 1094)	Meldungsfilterbedingungen abfragen
MSRTGetInfotextPlus (Seite 1105)	Infotext abfragen
MSRTGetLastMsgWithCommentPlus (Seite 1075)	Letzte Archivmeldung abfragen
MSRTGetMsgActualPlus (Seite 1077)	Anzahl von Meldungen ermitteln
MSRTGetMsgCSDDataPlus (Seite 1078)	Meldung-Projektierungsdaten abfragen
MSRTGetMsgPriorityPlus (Seite 1080)	Priorität der Meldung abfragen
MSRTGetMsgQuitPlus (Seite 1081)	Anzahl quittierpflichtiger Meldungen abfragen
MSRTGetMsgTextPlus (Seite 1083)	Meldungstextblock abfragen
MSRTGetSelectedMsgPlus (Seite 1084)	Meldungsnummer ermitteln
MSRTLoopInAlarmPlus (Seite 1088)	Loop in Alarm
MSRTMsgWinCommandPlus (Seite 1100)	Bedienfunktion in Meldefenster ausführen
MSRTPrintMProtPlus (Seite 1062)	Druckt Meldefolgeprotokoll anstehender Meldungen
MSRTQuitGroupPlus (Seite 1092)	Meldungsgruppe quittieren
MSRTQuitHornPlus (Seite 1054)	Melder quittieren
MSRTResetMsgPlus (Seite 1086)	Meldung quittieren
MSRTSetCommentInstancePlus (Seite 1103)	Kommentartext festlegen
MSRTSetInfotextPlus (Seite 1106)	Infotext festlegen
MSRTSetMsgFilterPlus (Seite 1097)	Meldungsfilter setzen
MSRTSetMsgWinFilterPlus (Seite 1098)	Meldungsfilter für ein Meldefenster setzen
MSRTStartMsgServicePlus (Seite 1049)	Sende- und Empfangsservice starten
MSRTStopMsgServicePlus (Seite 1053)	Sende- und Empfangsservice stoppen

Siehe auch

MSRTEnumMsgRTDataPlus (Seite 1072)

MSRTLCKGroupPlus (Seite 1091)

2.10.1.2 Übersicht über die Strukturen (RT Professional)**Übersicht**

MSG_BACKUP_STRUCT_PLUS (Seite 1044)	Auslagerungsliste
MSG_CLASS_STRUCT_PLUS (Seite 1034)	Meldungsklasse
MSG_COMMENT_INSTANCE_STRUCT_PLUS (Seite 1041)	Kommentar
MSG_CSDATA_STRUCT_PLUS (Seite 1037)	Einzelmeldung
MSG_FILTER_STRUCT_PLUS (Seite 1027)	Meldungsfilter
MSG_INFOTEXT_STRUCT_PLUS (Seite 1043)	Infotext
MSG_RTCREATE_STRUCT_PLUS (Seite 1022)	Erzeugen einer Meldung im Runtime-System

2.10 Funktionen für Meldungen (RT Professional)

MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024)	Einzelmeldung
MSG_RTDATA_INSTANCE_STRUCT_PLUS (Seite 1031)	Einzelmeldung
MSG_RTGROUPENUM_STRUCT_PLUS (Seite 1039)	Einzelmeldungen einer Meldungsgruppe
MSG_RTGROUPSET_STRUCT_PLUS (Seite 1040)	Meldungsgruppendaten
MSG_RTLOCK_STRUCT_PLUS (Seite 1045)	Meldungssperre
MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046)	Textbegleitwerte

2.10.1.3 Fehlermeldungen (RT Professional)

Übersicht

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zurückgegeben werden:

Alarm Logging CS

MSG_ERR_CS_CONNECT	0x00001001	Keine Verbindung zum Alarm Logging CS oder zum Projekt.
MSG_ERR_CS_PROJECT	0x00001002	Falsches oder ungültiges Projekt
MSG_ERR_CS_PROJECTNAME	0x00001003	Falscher oder ungültiger Projektname
MSG_ERR_CS_PARAM	0x00001004	Falscher Parameter; dwError2 enthält die Nummer des falschen Parameters.
MSG_ERR_CS_NODATA	0x00001005	Keine Daten gefunden
MSG_ERR_CS_STRUCTFIELD	0x00001006	Fehler in einem Struktur-Feld; dwError2 enthält die Nummer des Struktur-Feldes, in dem sich der falsche Wert befindet.
MSG_ERR_CS_DBCONNECT	0x00001011	Kein DBConnect möglich
MSG_ERR_CS_DBCDELETE	0x00001012	Fehler beim Löschen (keine Daten?)
MSG_ERR_CS_DBOPEN	0x00001013	Tabelle mit Einzelmeldungen konnte nicht geöffnet werden
MSG_ERR_CS_SOURCES	0x00001014	kein Speicher mehr ?
MSG_ERR_CS_CREATOR_ID	0x00001015	CreatorID ist WinCC-CreatorID!
MSG_ERR_CS_NOTSUPPORTED	0x00001016	Funktion wird nicht unterstützt
MSG_ERR_UNKNOWN_ERROR	0x00001020	Ein nicht näher spezifizierter Fehler ist aufgetreten (ab WinCC V6.0)
MSG_ERR_API_TERMINATED	0x00001021	Die Bearbeitung wurde vorzeitig abgebrochen (ab WinCC V6.0)
MSG_ERR_NO_MEMORY	0x00001022	Zuwenig Hauptspeicher (ab WinCC V6.0)
MSG_ERR_DB_FAIL	0x00001023	Datenbank inkonsistent (ab WinCC V6.0)

MSG_ERR_I_WRITE	0x00001024	Daten konnten nicht in die Datenbank eingetragen werden. dwError2 und dwError3 enthalten weitere Information. (ab WinCC V6.0)
MSG_ERR_CS_EXIST	0x00001025	Element existiert bereits (ab WinCC V6.0)
MSG_ERR_CS_NO_EXIST	0x00001026	Element existiert nicht (ab WinCC V6.0)
MSG_ERR_HANDLE	0x00000090	Handle ungültig

Alarm Logging RT

MSG_ERR_RT_NOTCONNECTED	0x00000001	Verbindung zum Alarm Logging RT nicht aufgebaut
MSG_ERR_RT_NOINIT	0x00000002	Alarm Logging RT nicht initialisiert
MSG_ERR_RT_SERVICEMAX	0x00000003	Maximale Anzahl Services erreicht
MSG_ERR_MSG_NOEXIST	0x00000010	Meldung existiert nicht
MSG_ERR_MSG_NOTFOUND	0x00000011	Meldung nicht gefunden
MSG_ERR_MSG_NOQUIT	0x00000012	Meldung hat keine Quittierpflicht
MSG_ERR_MSG_ALREADYQUIT	0x00000013	Meldung schon quittiert
MSG_ERR_MSG_NOCLASS	0x00000014	Klasseninformation nicht vorhanden
MSG_ERR_MSG_STATE	0x00000015	Meldungsstatus unbekannt
MSG_ERR_MSG_LOCKED	0x00000016	Meldung gesperrt
MSG_ERR_MSG_DATETIME	0x00000017	Meldung hat ungültiges Datum, oder ungültige Uhrzeit
MSG_ERR_WIN_NAMENOTFOUND	0x00000020	Templatename nicht gefunden
MSG_ERR_WIN_CREATE	0x00000021	Fehler beim Erzeugen des Meldefensters
MSG_ERR_WIN_DATA	0x00000022	Meldefensterdaten ungültig
MSG_ERR_VAR_BITEXCEEDS_	0x00000030	Bit für Variable ungültig
MSG_ERR_VAR_NOCHANGE_	0x00000031	Keine Variablenänderung
MSG_ERR_VAR_PROCESS_USED_	0x00000032	Ereignis - Variable ist bereits belegt
MSG_ERR_VAR_QUIT_USED_	0x00000033	Quittier - Variable ist bereits belegt
MSG_ERR_VAR_STATE_USED_	0x00000034	Zustands - Variable ist bereits belegt
MSG_ERR_VAR_VARIANTCONV_	0x00000035	Fehler beim Konvertieren des Variant
MSG_ERR_VAR_USED_TYPDIF_	0x00000036	Ereignis - Variable belegt, unterschiedliche Typen(Quit, Create, ...)
MSG_ERR_HANDLE	0x00000090	Handle ungültig
MSG_ERR_FILTER	0x00000091	Filterbedingung ungültig
MSG_ERR_API_PARAM	0x00000095	Parameter ungültig
MSG_ERR_API_SERVICE	0x00000099	dwServiceID ungültig
MSG_ERR_API_EXCEPTION	0x0000009a	Eine Ausnahmesituation im Code ist aufgetreten
MSG_ERR_API_DATETIME	0x0000009b	Datum-Uhrzeit-Stempel ungültig
MSG_ERR_API_SERVICEMAX	0x0000009c	Maximale Anzahl Services erreicht
MSG_ERR_API_NODATA	0x0000009d	Keine Daten
MSG_ERR_PROT_ID	0x00000111	Ungültige Protokollkennung
MSG_ERR_PROT_BLOCKS	0x00000110	Keine Blöcke im Protokoll
MSG_ERR_PROT_ACTIVE	0x00000112	Protokollservice bereits aktiv

2.10 Funktionen für Meldungen (RT Professional)

MSG_ERR_NODEFAULTSERVER	0x00000401	Kein Standardserver projiziert
MSG_ERR_NOLOCALSERVER	0x00000402	Kein lokaler Server verfügbar
MSG_ERR_NOSERVER	0x00000403	Kein Standardserver projiziert und kein lokaler Server verfügbar
MSG_ERR_NOMC	0x00000404	Kein Multiclient Projekt (wird hier nicht verwendet)
MSG_ERR_NOMCDEFAULTSERVER	0x00000405	Kein Multiclient Projekt und "@default" angegeben (wird hier nicht verwendet)

2.10.1.4 Konstanten (RT Professional)

Allgemeine Festlegungen

MSG_MAX_CLASS	(Wert: 18)	Anzahl Meldeklassen
MSG_MAX_CLASSTYPE	(Wert: 16)	Anzahl Meldearten in Meldeklasse
MSG_MAX_PVALUE	(Wert: 10)	Anzahl Prozesswerte
MSG_MAX_TB	(Wert: 10)	Anzahl Anwendertexte
MSG_MAX_HITLIST	(Wert: 7)	Anzahl Hitlisten-Elemente
MSG_MAX_TB_CONTENT	(Wert: 255)	Anzahl Zeichen pro Anwendertext
MSG_MAX_TEXTLEN	(Wert: 64)	Anzahl Zeichen für alle Texte
MSG_MAX_TEXTVALUE	(Wert: 32)	Anzahl Zeichen Textbegleitwerte
MSG_MAX_TEXTVALUE256	(Wert: 256)	Anzahl Zeichen Textbegleitwerte
MSG_MAX_STATE	(Wert: 4)	Anzahl Zustandstexte
MSG_MAX_LOCKITEMS	(Wert: 16)	Anzahl Meldungen für eine Sperre
MSG_MAX_LEVEL	(Wert: 6)	Anzahl Knoten für Sammelmeldungen
MSG_MAX_MSG_COUNT	(Wert: 150000) vor V6.2: (Wert: 50000)	Maximale Anzahl Einzelmeldungen im Projekt
MSG_MAX_GROUPITEMS	(Wert: 1000)	Anzahl Meldungen für eine Gruppe bei Enum
MSG_MAX_USERNAME	(Wert: 16)	maximale Länge eines Benutzernamens
MSG_MAX_APPLNAME	(Wert: 32)	maximale Länge eines Applikationsnamens
MSG_MAX_INSTANCE	(Wert: 512)	Instanzname
MSG_MAX_ARCHIV_QUEUE	(Wert: 1500)	Max. Anzahl pufferbarer Archivverträge

Meldezustände

MSG_STATE_COME	(Wert: 0x00000001)	Meldung gekommen
MSG_STATE_GO	(Wert: 0x00000002)	Meldung gegangen
MSG_STATE_QUIT	(Wert: 0x00000003)	Meldung quittiert
MSG_STATE_LOCK	(Wert: 0x00000004)	Meldung gesperrt
MSG_STATE_UNLOCK	(Wert: 0x00000005)	Meldung Freigabe
MSG_STATE_HIDE	(Wert: 0x0000000A)	Meldung ausgeblendet
MSG_STATE_UNHIDE	(Wert: 0x0000000B)	Meldung eingeblendet
MSG_STATE_HIDE_MANUAL	(Wert: 0x0000000C)	Meldung manuell ausgeblendet
MSG_STATE_UNHIDE_MANUAL	(Wert: 0x0000000D)	Meldung manuell eingeblendet
MSG_STATE_QUIT_SYSTEM	(Wert: 0x00000010)	Meldung quittiert durch System
MSG_STATE_QUIT_EMERGENCY	(Wert: 0x00000011)	Not-Quittierung
MSG_STATE_QUIT_HORN	(Wert: 0x00000012)	Hupenquittierung
MSG_STATE_COMEGO	(Wert: 0x00000013)	Meldung gekommen und gegangen, nur Anzeige in Meldeliste
MSG_STATE_RESET	(Wert: 0x00020000)	Bit für Meldungs-Reset
MSG_STATE_SUMTIME	(Wert: 0x00040000)	Bit für Sommerzeit aktiv
MSG_STATE_INSTANCE	(Wert: 0x00080000)	Bit für Instanzmeldung (n Meldungen einer Nr.)
MSG_STATE_TIMEINVALID	(Wert: 0x00100000)	Bit für ungültigen Datum/Uhrzeitstempel Oder-Verknüpfung mit (MSG_STATE_COME, MSG_STATE_GO, ...)
MSG_STATE_HIDDEN	(Wert: 0x02000000)	Bit für aktuell ausgeblendete Meldung, verfügbar ab V6.2

Flags einer Meldung in der Datenbanktabelle

MSG_FLAG_SUMTIME	(Wert: 0x00000001)	Sommerzeit aktiv
MSG_FLAG_COMMENT	(Wert: 0x00000002)	Meldung hat Kommentar
MSG_FLAG_ARCHIV	(Wert: 0x00000004)	Archivieren
MSG_FLAG_PROTOCOL	(Wert: 0x00000008)	Protokollieren
MSG_FLAG_TEXTVALUES	(Wert: 0x00000010)	Meldung hat Textbegleitwerte
MSG_FLAG_TIMEINVALID	(Wert: 0x00000020)	Bit für ungültigen Datum/Uhrzeitstempel
MSG_FLAG_INSTANCE	(Wert: 0x00000040)	Kennzeichnung von Instanzmeldungen
MSG_FLAG_HIDDEN	(Wert: 0x00000080)	Ausgeblendete Meldung, verfügbar ab V6.2
MSG_FLAG_TEXT10_EXT	(Wert: 0x00000100)	Erweiterte Kanal-Diagnose, verfügbar ab V7.0
MSG_FLAG_INSTANCE10	(Wert: 0x08000000)	Instanzmeldung mit Prozesswert10, verfügbar ab V5.0 SP2

MSG_FLAG_TEXTREF1	(Wert: 0x10000000)	Textreferenz1, verfügbar ab V5.0 SP2
MSG_FLAG_TEXTREF2	(Wert: 0x20000000)	Textreferenz2, verfügbar ab V5.0 SP2

Flags einer Meldung in der Online-Liste

Erst ab WinCC V6.0 SP4 in m_global.h definiert.

MSG_RT_FLAG_COME	(Wert: 0x00000001)	Meldung gekommen
MSG_RT_FLAG_GO	(Wert: 0x00000002)	Meldung gegangen
MSG_RT_FLAG_QUIT_COME	(Wert: 0x00000004)	Quittierung gekommen
MSG_RT_FLAG_QUIT_GO	(Wert: 0x00000008)	Quittierung gegangen
MSG_RT_FLAG_NOQUIT	(Wert: 0x00000010)	Keine Quittierpflicht
MSG_RT_FLAG_QUIT_NOGO	(Wert: 0x00000020)	TM II Quittierung
MSG_RT_FLAG_FLASH	(Wert: 0x00000040)	Blinkend
MSG_RT_FLAG_FIRSTFLASH	(Wert: 0x00000080)	Nur erste Meldung darf blinken
MSG_RT_FLAG_NOLIST	(Wert: 0x00000100)	Meldungen werden nicht in die Meldeliste aufgenommen
MSG_RT_FLAG_UNIQUE_USERNAME	(Wert: 0x00000200)	Meldungen mit eindeutigem Username
MSG_RT_FLAG_ONL_COM_COMMENT	(Wert: 0x00000400)	In Meldeliste nur Kommentar der gekommenen Meldung anzeigen
MSG_RT_FLAG_HIDDEN	(Wert: 0x00000800)	Meldung ausgeblendet
MSG_RT_FLAG_TEXT10_EXT	(Wert: 0x00001000)	Erweiterte Kanal-Diagnose, verfügbar ab V7.0
MSG_RT_FLAG_DELETE	(Wert: 0x00010000)	Meldung ist löscherbar
MSG_RT_FLAG_SUMTIME	(Wert: 0x00020000)	Sommerzeit-Flag
MSG_RT_FLAG_SQUIT	(Wert: 0x00040000)	Meldung hat Sammelquittierung
MSG_RT_FLAG_QUITNR	(Wert: 0x00080000)	Meldungsnummer - Quit
MSG_RT_FLAG_MSGLIST	(Wert: 0x00100000)	Meldung von Meldeliste
MSG_RT_FLAG_COMMENT	(Wert: 0x00200000)	Meldung hat Kommentar
MSG_RT_FLAG_ARCHIV	(Wert: 0x00400000)	Meldung wird archiviert
MSG_RT_FLAG_PROTOCOL	(Wert: 0x00800000)	Meldung wird protokolliert
MSG_RT_FLAG_INSTANCE	(Wert: 0x01000000)	Flag für Instanzmeldung
MSG_RT_FLAG_TIMEINVALID	(Wert: 0x02000000)	Flag für ungültigen Datum/Uhrzeit-Stempel
MSG_RT_FLAG_LOCKGROUP	(Wert: 0x04000000)	Flag für Gruppensperre
MSG_RT_FLAG_INSTANCE10	(Wert: 0x08000000)	Bit für neue Instanzmeldung mit Prozesswert 10
MSG_RT_FLAG_TEXTREF1	(Wert: 0x10000000)	Bit für Textreferenz 1
MSG_RT_FLAG_TEXTREF2	(Wert: 0x20000000)	Bit für Textreferenz 2
MSG_RT_FLAG_QUITCOUNTER	(Wert: 0x40000000)	Bit für Quittierung über Counter
MSG_RT_FLAG_QUITGROUP	(Wert: 0x80000000)	Bit für Quittierung über Counter

Flags für Einzelmeldungen

MSG_PARAMS_SQUIT	(Wert: 0x00000001)	Meldung ist einzelquittierpflichtig.
MSG_PARAMS_HORN	(Wert: 0x00000002)	Meldung steuert Hupe an.
MSG_PARAMS_ARCHIV	(Wert: 0x00000004)	Meldung soll archiviert werden.
MSG_PARAMS_PROTOCOL	(Wert: 0x00000008)	Meldung soll protokollieren werden.
MSG_PARAMS_VARFLANK	(Wert: 0x00000010)	Meldung wird bei fallender Flanke der Variablen erzeugt.
MSG_PARAMS_LOCKED	(Wert: 0x00000020)	Meldung ist bei Hochlauf gesperrt.
MSG_PARAMS_INFOTEXT	(Wert: 0x00000040)	Meldung besitzt einen Infotext.
MSG_PARAMS_LOOPIN-ALARM	(Wert: 0x00000080)	Meldung besitzt Loop In Alarm.
MSG_PARAMS_APFUNC	(Wert: 0x00000100)	Meldung steuert globale AP-Funktion an.
MSG_PARAMS_NORMDLL	(Wert: 0x00000200)	Meldung besitzt eine Normierungs-DLL.
MSG_PARAMS_TEXT10_EXT	(Wert: 0x00000400)	erweiterte Kanal-Diagnose, verfügbar ab V7.0.

Zustandsautomaten

MSG_QUIT_QUIT_COME	(Wert: 0x00000004)	Quittierung gekommen
MSG_QUIT_QUIT_GO	(Wert: 0x00000008)	Quittierung gegangen
MSG_QUIT_NOQUIT	(Wert: 0x00000010)	Keine Quittierpflicht
MSG_QUIT_QUIT_NOGO	(Wert: 0x00000020)	TM II Quittierung (ohne gehende Meldung)
MSG_QUIT_FLASH	(Wert: 0x00000040)	Blinkend
MSG_QUIT_FIRSTFLASH	(Wert: 0x00000080)	Nur erste Meldung darf blinken
MSG_QUIT_NOLIST	(Wert: 0x00000100)	Meldungen werden nicht in die Meldeliste aufgenommen
MSG_QUIT_UNIQUE_USER	(Wert: 0x00000200)	Meldungen mit eindeutigem Username
MSG_QUIT_ONL_COM_COME	(Wert: 0x00000400)	In Meldeliste nur Kommentar der gekommenen Meldung anzeigen

Sammelmeldungen

MSG_COLLECT_ALL	(Wert: 0x00000001)	Alle Meldungen
MSG_COLLECT_QUIT	(Wert: 0x00000002)	Alle unquittierten Meldungen
MSG_COLLECT_CLASS	(Wert: 0x00000003)	Alle Meldungen einer Meldeklasse

2.10 Funktionen für Meldungen (RT Professional)

MSG_COLLECT_TYPE	(Wert: 0x00000004)	Alle Meldungen einer Meldeart
MSG_COLLECT_USER	(Wert: 0x00000005)	Benutzerdefinierte Sammelmeldungen

Einzelmeldungen

MSG_SINGLE_OPERATION	(Wert: 0x00BEDBED)	Bedienmeldung
----------------------	--------------------	---------------

Meldeklassen

MSG_CLASS_SYSTEM_QUIT	(Wert: 0x00000011)	Systemmeldeklasse mit Quit
MSG_CLASS_SYSTEM	(Wert: 0x00000012)	Systemmeldeklasse ohne Quit

Meldeblöcke

Im HIWORD steht der Haupttyp der Meldeblöcke. Das LOWORD enthält die Untertypen der Meldeblöcke.

MSG_BLOCK_SYSTEM	(Wert: 0x00010000)	Systemblöcke
MSG_BLOCK_DATE	(Wert: 0x00010001)	Datum
MSG_BLOCK_TIME	(Wert: 0x00010002)	Uhrzeit
MSG_BLOCK_TIMEDIFF	(Wert: 0x00010003)	Meldungsdauer
MSG_BLOCK_SUMWIN	(Wert: 0x00010004)	Sommerzeit/Winterzeit
MSG_BLOCK_STATE	(Wert: 0x00010005)	Meldungsstatus
MSG_BLOCK_QUIT	(Wert: 0x00010006)	Quittierstatus
MSG_BLOCK_NUMBER	(Wert: 0x00010007)	Meldungsnummer
MSG_BLOCK_CLASS	(Wert: 0x00010008)	Meldungsklasse
MSG_BLOCK_TYP	(Wert: 0x00010009)	Meldungsart
MSG_BLOCK_AGCPU	(Wert: 0x0001000A)	AG-/CPU-Nummer
MSG_BLOCK_DMVAR	(Wert: 0x0001000B)	Meldungsvariable
MSG_BLOCK_LIMIT	(Wert: 0x0001000C)	Grenzwert-Überschreitung
MSG_BLOCK_ARCHIV	(Wert: 0x0001000D)	Archivkennung
MSG_BLOCK_PROTOCOL	(Wert: 0x0001000E)	Protokollkennung
MSG_BLOCK_COMMENT	(Wert: 0x0001000F)	Kommentar
MSG_BLOCK_INFO	(Wert: 0x00010010)	Infotext
MSG_BLOCK_LOOPINALARM	(Wert: 0x00010011)	Loop In Alarm / AP-Funktion
MSG_BLOCK_COMPUTERNAME	(Wert: 0x00010012)	Rechnername
MSG_BLOCK_USERNAME	(Wert: 0x00010013)	Anwendername
MSG_BLOCK_PRIORITY	(Wert: 0x00010014)	Priorität

MSG_BLOCK_TEXT	(Wert: 0x00020000)	Die max. 16 Anwender-Textblöcke können über die Konstanten MSG_BLOCK_TEXT1 ... MSG_BLOCK_TEXT16 mit den Werten 0x00020001 ... 0x00020010 angesprochen werden.
MSG_BLOCK_VALUE	(Wert: 0x00030000)	Die max. 16 Prozesswertblöcke können über die Konstanten MSG_BLOCK_VALUE1 ... MSG_BLOCK_VALUE16 mit den Werten 0x00030001 ... 0x00030010 angesprochen werden.
MSG_MAX_SYSBL		Anzahl der Systemmeldeblöcke = MSG_BLOCK_LOOPIN-ALARM -MSG_BLOCK_SYSTEM
MSG_MAX_SYSBL_V6		Anzahl der Systemblöcke ab V6.0 = MSG_BLOCK_PRIORITY - MSG_BLOCK_SYSTEM
MSG_MAX_BLOCKS		Gesamtanzahl der Meldeblöcke = (MSG_MAX_TB + MSG_MAX_PVALUE + MSG_MAX_SYSBL)
MSG_MAX_BLOCKS_V6		Gesamtanzahl der Meldeblöcke ab V6.0 = (MSG_MAX_TB + MSG_MAX_PVALUE + MSG_MAX_SYSBL_V6)

Datumsformat

DATE_FORMAT_YEAR_2	(Wert: 0x00000001)	Jahresangabe zweistellig
DATE_FORMAT_YEAR_4	(Wert: 0x00000002)	Jahresangabe vierstellig
DATE_FORMAT_DMY	(Wert: 0x00000004)	Reihenfolge: Tag, Monat, Jahr
DATE_FORMAT_MDY	(Wert: 0x00000008)	Reihenfolge: Monat, Tag, Jahr
DATE_FORMAT_YMD	(Wert: 0x00000010)	Reihenfolge: Jahr, Monat, Tag

Uhrzeit-Format

TIME_FORMAT_HOUR	(Wert: 0x00000001)	Stunde
TIME_FORMAT_MIN	(Wert: 0x00000002)	Minute
TIME_FORMAT_SEC	(Wert: 0x00000004)	Sekunde
TIME_FORMAT_MSEC	(Wert: 0x00000008)	Millisekunde
TIME_FORMAT_24HOUR	(Wert: 0x00000010)	24-Stunden-Format
TIME_FORMAT_DAY	(Wert: 0x00000020)	Stundenanzeige

Meldenummer - Format

Im HiWord steht die Formatierungsvorschrift (mit führenden Nullen), das LoWord enthält die Anzahl der Stellen.

MSGNR_FORMAT_NOLEADING	(Wert: 0x00000000)	
MSGNR_FORMAT_LEADINGZERO	(Wert: 0x00010000)	
MSG_BLOCK_FORMAT_RIGHT	(Wert: 0x00020000)	rechtsbündige Ausgabe
MSG_BLOCK_FORMAT_CENTER	(Wert: 0x00040000)	zentrierte Ausgabe
MSG_BLOCK_FORMAT_FLASH	(Wert: 0x80000000)	Blinken des Meldeblocks

Elemente der Statuszeile

MSG_SB_HELP	(Wert: 0x00000001)	Hilfezeile
MSG_SB_DATE	(Wert: 0x00000002)	Datum
MSG_SB_TIME	(Wert: 0x00000004)	Uhrzeit
MSG_SB_COUNTALL	(Wert: 0x00000008)	Meldungen gesamt
MSG_SB_COUNTSEL	(Wert: 0x00000010)	Meldungen aktuell
MSG_SB_COUNTQUIT	(Wert: 0x00000020)	Meldungen Quit
MSG_SB_SELECT	(Wert: 0x00000040)	Selektion aktiv
MSG_SB_LOCK	(Wert: 0x00000080)	Sperre aktiv
MSG_SB_HIDE	(Wert: 0x00000100)	Ausblenden von Meldungen aktiv
MSG_SB_WARNING	(Wert: 0x00000200)	Ausblend-Information
MSG_MAX_SBITEMS	(Wert: 10)	Maximale Anzahl der Elemente in der Statuszeile.

Elemente der Symbolleiste

MSG_TB_MSGWIN	(Wert: 0x00000001)	Schaltfläche "Meldefenster" anzeigen.
MSG_TB_ARC_S	(Wert: 0x00000002)	Schaltfläche "Umlaufarchiv" anzeigen.
MSG_TB_ARC_L	(Wert: 0x00000004)	Schaltfläche "Folgearchiv" anzeigen.
MSG_TB_QH	(Wert: 0x00000008)	Schaltfläche "Hupenquittierung" anzeigen.
MSG_TB_QM	(Wert: 0x00000010)	Schaltfläche "Einzelquittierung" anzeigen.
MSG_TB_QS	(Wert: 0x00000020)	Schaltfläche "Sammelquittierung" anzeigen.
MSG_TB_SCROLL	(Wert: 0x00000040)	Schaltfläche "Scrollfunktion Ein / Aus" anzeigen.

2.10 Funktionen für Meldungen (RT Professional)

MSG_TB_SELECT	(Wert: 0x00000080)	Schaltfläche "Selektion - Dialog" anzeigen.
MSG_TB_LOCK	(Wert: 0x00000100)	Schaltfläche "Sperrdialog" anzeigen.
MSG_TB_PROTOCOL	(Wert: 0x00000200)	Schaltfläche "Protokolldialog" anzeigen.
MSG_TB_CITYRUF	(Wert: 0x00000400)	Schaltfläche "City - Ruf" anzeigen.
MSG_TB_RESET	(Wert: 0x00000800)	Schaltfläche "Reset - Knopf" anzeigen.
MSG_TB_MSGFIRST	(Wert: 0x00001000)	Schaltfläche "Anfang der Liste" anzeigen.
MSG_TB_MSGLAST	(Wert: 0x00002000)	Schaltfläche "Ende der Liste" anzeigen.
MSG_TB_MSGNEXT	(Wert: 0x00004000)	Schaltfläche "Nächste Meldung" anzeigen.
MSG_TB_MSGPREV	(Wert: 0x00008000)	Schaltfläche "Vorherige Meldung" anzeigen.
MSG_TB_INFO	(Wert: 0x00010000)	Schaltfläche "Infotexte" anzeigen.
MSG_TB_COMMENT	(Wert: 0x00020000)	Schaltfläche "Kommentar eingeben" anzeigen.
MSG_TB_LOOPINALARM	(Wert: 0x00040000)	Schaltfläche "Loop In Alarm" anzeigen.
MSG_TB_LANGUAGESET	(Wert: 0x00080000)	Schaltfläche "Sprachumschaltung" anzeigen
MSG_TB_PRINT	(Wert: 0x00100000)	Schaltfläche "Drucken" anzeigen.
MSG_TB_LOCKWIN	(Wert: 0x00200000)	Schaltfläche "Sperrfenster" anzeigen
MSG_TB_UNLOCKMSG	(Wert: 0x00400000)	Schaltfläche "Meldung freigeben" anzeigen
MSG_TB_LOCK_UNLOCK_MSG	(Wert: 0x00400000)	Schaltfläche "Meldung sperren oder freigeben" anzeigen
MSG_TB_SORTDLG	(Wert: 0x00800000)	Schaltfläche "Sortierdialog" anzeigen
MSG_TB_TIMEBASE	(Wert: 0x01000000)	Schaltfläche "Dialog für Zeitbasis" anzeigen
MSG_TB_HITLIST	(Wert: 0x02000000)	Schaltfläche "Hitliste" anzeigen
MSG_TB_HIDELIST	(Wert: 0x04000000)	Schaltfläche "Liste ausgeblendeter Meldungen" anzeigen
MSG_TB_UNHIDEMSG	(Wert: 0x08000000)	Schaltfläche "Meldung einblenden" anzeigen
MSG_TB_HIDE_UNHIDE_MSG	(Wert: 0x08000000)	Schaltfläche "Meldung ausblenden und einblenden" anzeigen
MSG_TB_HIDEDLG	(Wert: 0x10000000)	Schaltfläche "Meldung ausblenden und einblenden Dialog" anzeigen

2.10 Funktionen für Meldungen (RT Professional)

MSG_MAX_TBITEMS	(Wert: 24)	Maximale Anzahl der Bedienelemente in der Symbolleiste.
MSG_MAX_TBITEMS_V6	(Wert: 25)	Maximale Anzahl, verfügbar ab V6
MSG_MAX_TBITEMS_V6_1	(Wert: 26)	Maximale Anzahl, verfügbar ab V6.1
MSG_MAX_TBITEMS_V6_2	(Wert: 30)	Maximale Anzahl, verfügbar ab V6.2
MSG_ALIGN_TOP	(Wert: 1)	Ausrichtung oben
MSG_ALIGN_BOTTOM	(Wert: 2)	Ausrichtung unten
MSG_ALIGN_LEFT	(Wert: 3)	Ausrichtung links
MSG_ALIGN_RIGHT	(Wert: 4)	Ausrichtung rechts

Fensteroptionen

MSG_MWWIN_SWITCHENAB- LE	(Wert: 0x00000001)	Umschaltung Prozessarchiv
MSG_MWWIN_PROCESS	(Wert: 0x00000002)	Prozessmeldefenster
MSG_MWWIN_ARCHIV_CIR- CLE	(Wert: 0x00000004)	Umlaufarchiv
MSG_MWWIN_ARCHIV_LONG	(Wert: 0x00000008)	Folgearchiv
MSG_MWWIN_GRIDLINE- HORZ	(Wert: 0x00000010)	Rasterlinie horizontal
MSG_MWWIN_GRIDLINEVERT	(Wert: 0x00000020)	Rasterlinie vertikal
MSG_MWWIN_LINETITLE	(Wert: 0x00000040)	Zeilenbeschriftung
MSG_MWWIN_LINEHEIGHT	(Wert: 0x00000080)	Zeilenhöhe änderbar
MSG_MWWIN_LINEHEIGHT- FONT	(Wert: 0x00000100)	Font an Zeilenhöhe anpassen
MSG_MWWIN_COLUMNTITLE	(Wert: 0x00000200)	Spaltenbeschriftung
MSG_MWWIN_COLUMNTI- TLE2	(Wert: 0x00000400)	2 Titelzeilen
MSG_MWWIN_COLUMNTI- TLE3	(Wert: 0x00000800)	3 Titelzeilen
MSG_MWWIN_COLUMN- WIDTH	(Wert: 0x00001000)	Spaltenbreite änderbar
MSG_MWWIN_COLUMNMOVE	(Wert: 0x00002000)	Spalten dürfen bewegt werden
MSG_MWWIN_CELLSELECT	(Wert: 0x00004000)	Selektion nur Zelle (Default-Einstellung: Zeile)
MSG_MWWIN_LINESELECT	(Wert: 0x00008000)	Zeilenselektion
MSG_MWWIN_AUTOSCROL- LOFF	(Wert: 0x00010000)	Auto-Scroll Aus (Default-Einstellung: Ein)
MSG_MWWIN_TITLECUT	(Wert: 0x00020000)	Titel wird bei Bedarf durch Punkte gekürzt
MSG_MWWIN_CELLCUT	(Wert: 0x00040000)	Zelleninhalt bei Bedarf durch Punkte gekürzt
MSG_MWWIN_TITLEBAR	(Wert: 0x00080000)	Titelzeile des Fensters anzeigen (+ Rahmen)

MSG_MWWIN_SIZEABLE	(Wert: 0x00100000)	Größe des Fensters veränderbar
MSG_MWWIN_TOOLTIP	(Wert: 0x00200000)	Tooltipanzeige speziell für Meldetexte
MSG_MWWIN_LOOP_IN_ALARM_DC	(Wert: 0x00400000)	Ein Klick für "Loop in Alarm"
MSG_MWWIN_SELECTFOCUSRECT	(Wert: 0x00800000)	Selektion markiert mit Rahmen statt Inversion
MSG_MWWIN_OPMSG_HIDE	(Wert: 0x01000000)	Bedienmeldung bei Ausblenden
MSG_MWWIN_OPMSGUNHIDE	(Wert: 0x02000000)	Bedienmeldung bei Einblenden
MSG_MWWIN_INSTANCE_COLWIDTH	(Wert: 0x08000000)	Instanzspezifische Spaltenbreite
MSG_MWWIN_INSTANCEBLOCKINFO	(Wert: 0x10000000)	Instanzspezifische Meldeblock-Einstellungen
MSG_MWWIN_OPMSG_LOCK	(Wert: 0x20000000)	Bedienmeldung bei Sperren
MSG_MWWIN_OPMSG_UNLOCK	(Wert: 0x40000000)	Bedienmeldung bei Freigeben
MSG_MWWIN_OPMSG_ACK	(Wert: 0x80000000)	Bedienmeldung bei Quittieren

Filterbedingungen

MSG_FILTER_DATE_FROM	(Wert: 0x00000001)	Datum von
MSG_FILTER_DATE_TO	(Wert: 0x00000002)	Datum bis
MSG_FILTER_TIME_FROM	(Wert: 0x00000004)	Uhrzeit von
MSG_FILTER_TIME_TO	(Wert: 0x00000008)	Uhrzeit bis
MSG_FILTER_NR_FROM	(Wert: 0x00000010)	Meldungsnummer von
MSG_FILTER_NR_TO	(Wert: 0x00000020)	Meldungsnummer bis
MSG_FILTER_CLASS	(Wert: 0x00000040)	Meldeklassen
MSG_FILTER_STATE	(Wert: 0x00000080)	Meldungsstatus
MSG_FILTER_AG_FROM	(Wert: 0x00000100)	AG-Nummer von
MSG_FILTER_AG_TO	(Wert: 0x00000200)	AG-Nummer bis
MSG_FILTER_AGSUB_FROM	(Wert: 0x00000400)	AG-Subnummer von
MSG_FILTER_AGSUB_TO	(Wert: 0x00000800)	AG-Subnummer bis
MSG_FILTER_ARC	(Wert: 0x00001000)	Archiv/Prot.-Kennung
MSG_FILTER_TEXT	(Wert: 0x00002000)	Meldetexte
MSG_FILTER_PVALUE	(Wert: 0x00004000)	Prozesswerte
MSG_FILTER_COUNTER_FROM	(Wert: 0x00008000)	Interner Meldungsähler von
MSG_FILTER_COUNTER_TO	(Wert: 0x00010000)	Interner Meldungsähler bis
MSG_FILTER_PROCESSMSG	(Wert: 0x00020000)	Prozessmeldungen
MSG_FILTER_SYMSG	(Wert: 0x00040000)	Systemmeldungen
MSG_FILTER_BEDMSG	(Wert: 0x00080000)	Bedienmeldungen
MSG_FILTER_QUICKSELECT	(Wert: 0x00100000)	Schnellselektion (aktueller Tag, ...)
MSG_FILTER_TEXT_EQUAL	(Wert: 0x00200000)	Meldetexte genau

2.10 Funktionen für Meldungen (RT Professional)

MSG_FILTER_PRIORITY_FROM	(Wert: 0x00400000)	Priorität von
MSG_FILTER_PRIORITY_TO	(Wert: 0x00800000)	Priorität bis
MSG_FILTER_VISIBLEONLY	(Wert: 0x01000000)	Eingeblendete Meldungen anzeigen
MSG_FILTER_HIDDENONLY	(Wert: 0x02000000)	Ausgeblendete Meldungen anzeigen
MSG_FILTER_NODYNAMIC	(Wert: 0x04000000)	Keine Selektion auf dynamische Texte
MSG_FILTER_DATE	Wert: (MSG_FILTER_DATE_FROM MSG_FILTER_DATE_TO)	
MSG_FILTER_TIME	Wert: (MSG_FILTER_TIME_FROM MSG_FILTER_TIME_TO)	
MSG_FILTER_NR	Wert: (MSG_FILTER_NR_FROM MSG_FILTER_NR_TO)	

Schnellselektion

Der Endzeitpunkt der Schnellselektion bezieht sich auf die aktuelle Systemzeit des lokalen Rechners. Der Startzeitpunkt wird n * (Monate, Tage, Stunden zurückgerechnet)

MSG_FILTER_QUICK_MONTH	(Wert: 0x00000001)	Letzte n Monate
MSG_FILTER_QUICK_DAYS	(Wert: 0x00000002)	Letzte n Tage
MSG_FILTER_QUICK_HOUR	(Wert: 0x00000003)	Letzte n Stunden

Archivtypen

MSG_ARPRO_LONGARCH	(Wert: 0x00000001)	Folgearchiv
MSG_ARPRO_SHORTARCH	(Wert: 0x00000002)	Umlaufarchiv
MSG_ARPRO_ARCHPROT	(Wert: 0x00000003)	Archiv-Protokoll
MSG_ARPRO_MFPROT	(Wert: 0x00000004)	Meldefolge-Protokoll

Flags für Archive

MSG_ARPRO_PARAM_HSPONLY	(Wert: 0x00000001)	Kurzzeitarchiv im Hauptspeicher
MSG_ARPRO_PARAM_CIRCLE	(Wert: 0x00000002)	Umlaufarchiv
MSG_ARPRO_PARAM_ENDLES	(Wert: 0x00000004)	Endlosarchiv
MSG_ARPRO_PARAM_RELOAD	(Wert: 0x00000008)	Rückladen nach Netzausfall
MSG_ARPRO_PARAM_LINEPRINT	(Wert: 0x00000010)	Zeilenweise Drucken (nur Meldefolgeprotokoll)

ID für Sperren von Meldungen

MSG_LOCK_UNKNOWN	(Wert: 0)	Sperre ist unbekannt
MSG_LOCK_CLASS	(Wert: 1)	Sperre von Meldeklasse
MSG_LOCK_GROUP1	(Wert: 2)	Sperre von Meldegruppen Level 1
MSG_LOCK_GROUP2	(Wert: 3)	Sperre von Meldegruppen Level 2

ID für Protokolle

MSG_PROT_MSG	(Wert: 1)	Meldefolgeprotokoll
MSG_PROT_ARCHIV_CIRCLE	(Wert: 2)	Archivprotokoll von Umlaufarchiv
MSG_PROT_ARCHIV_LONG	(Wert: 3)	Archivprotokoll von Folgearchiv

ID für Auslagerung von Meldearchiven

MSG_DB_FMT_CSV	(Wert: 1)	CSV/Dateiformat
MSG_DB_FMT_DBASE3	(Wert: 2)	DBASE3-Dateiformat
MSG_DB_FMT_WATCOM	(Wert: 3)	Watcom-Format
MSG_DB_FLAGS_OVERWRITE	(Wert: 1)	Überschreiben vorhandener Datensätze
MSG_DB_FLAGS_ADD	(Wert: 2)	Nur hinzufügen
MSG_DB_FLAGS_DELETE	(Wert: 4)	Quelldatensätze löschen
MSG_DB_FLAGS_NOEXPORT	(Wert: 8)	Datensätze nicht exportieren --> nur löschen (nicht mehr verfügbar)

Flags für die generische Projektierung

Die Flags können logisch verknüpft werden.

MSG_GENERIC_APPEND	(Wert: 0x00000001)	Neu anlegen
MSG_GENERIC_OVERWRITE	(Wert: 0x00000002)	Überschreiben
MSG_GENERIC_DELETE	(Wert: 0x00000004)	Löschen

Modus für Variablen

MSG_CS_VAR_NOTSET	(Wert: 0x0000)	keine Variable gesetzt
MSG_CS_VARNAME_SET	(Wert: 0x0001)	Variablenname ist gesetzt
MSG_CS_VARID_SET	(Wert: 0x0002)	Variablen-ID ist gesetzt
MSG_CS_VARALL_SET	(Wert: 0x0003)	Variablen-ID und Name gesetzt

Enumeration der Archive

MSG_ARCHIV_ENUM_DESC	(Wert: 0x00000001)	Sortierung absteigend (Default-Einstellung: aufsteigend)
----------------------	--------------------	--

2.10.2 Strukturen (RT Professional)

2.10.2.1 MSG_RTCREATE_STRUCT_PLUS (RT Professional)

Deklaration

```
typedef struct {
    DWORD dwMsgState;
    DWORD dwMsgNrLow;
    DWORD dwMsgNrHigh;
    DWORD dwFlags;
    SYSTEMTIME stMsgTime;
    double dPValue[MSG_MAX_PVALUE];
    WORD wPValueUsed;
    MSG_TEXTVAL256_STRUCT_PLUS mtTextValueW[MSG_MAX_PVALUE];
    WORD wTextValueUsed;
}
MSG_RTCREATE_STRUCT_PLUS;
```

Members

dwMsgState

Meldungsstatus entsprechend der Konstanten in der Datei "m_global.h":

MSG_STATE_COME	Meldung gekommen
MSG_STATE_GO	Meldung gegangen
MSG_STATE_QUIT	Meldung quittiert
MSG_STATE_LOCK	Meldung gesperrt
MSG_STATE_UNLOCK	Meldung Freigabe
MSG_STATE_HIDE	Meldung ausgeblendet
MSG_STATE_UNHIDE	Meldung eingeblendet
MSG_STATE_HIDE_MANUAL	Meldung manuell ausgeblendet
MSG_STATE_UNHIDE_MANUAL	Meldung manuell eingeblendet

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

stMsgTime

Datum und Uhrzeit für Kommen oder Gehen einer Meldung. Beim Quittieren wird die Systemzeit der PLC oder der OS benutzt.

Wird eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt, so ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

dPValue

Numerische Prozesswerte.

wPValueUsed

Benutzte Prozesswerte in bitcodierter Darstellung.

Jedes Bit darf nur in einem der beiden Member wPValueUsed oder wTextValueUsed gesetzt werden, da technisch ein Begleitwert nur Zahl oder Text sein kann. Bei fehlerhafter Doppelversorgung hat dPValue Vorrang.

mtTextValue

Struktur vom Typ MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046) mit den Textbegleitwerten.

wTextValueUsed

Benutzte Text-Prozesswerte in bitkodierter Darstellung.

Jedes Bit darf nur in einem der beiden Member wPValueUsed oder wTextValueUsed gesetzt werden, da technisch ein Begleitwert nur Zahl oder Text sein kann. Bei fehlerhafter Doppelversorgung hat dPValue Vorrang.

Bemerkung

Bevor Sie die Struktur ausfüllen, sollten Sie sie mit "0" initialisieren.

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

AUTOHOTSPOT	Meldung erzeugen
-------------	------------------

Siehe auch

MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046)

Übersicht über die Strukturen (Seite 1007)

2.10.2.2 MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (RT Professional)

Deklaration

```
typedef struct {
    DWORD dwMsgState;
    DWORD dwMsgNrLow;
    DWORD dwMsgNrHigh;
    SYSTEMTIME stMsgTime;
    DWORD dwTimeDiff;
    DWORD dwCounter;
    DWORD dwFlags;
    WORD wPValueUsed;
    WORD wTextValueUsed;
    double dPValue[MSG_MAX_PVALUE];
    MSG_TEXTVAL256_STRUCT_PLUS mtTextValue[MSG_MAX_PVALUE];
    WCHAR szInstance[MSG_MAX_INSTANCE+1];
    WCHAR szComment[MSG_MAX_TB_CONTENT+1];
    WCHAR szUser[MSG_MAX_USERNAME+1];
    WCHAR szComputerName[MAX_COMPUTERNAME_LENGTH+1];
    WCHAR szApplicationName[MSG_MAX_APPLNAME+1];
    WCHAR szServerPrefix[_MAX_PATH+1];
}
MSG_RTDATA_INSTANCECOMMENT_STRUCTPlus;
```

Members

dwMsgState

Meldungsstatus entsprechend der Konstanten in der Datei "m_global.h":

Die aufgeführten Stati werden nur intern vom System benutzt und können vom API nur mit MSRTCreateMsgInstanceWithCommentPlus gesetzt werden.

MSG_STATE_COME	Meldung gekommen
MSG_STATE_GO	Meldung gegangen
MSG_STATE_QUIT	Meldung quittiert
MSG_STATE_LOCK	Meldung gesperrt
MSG_STATE_UNLOCK	Meldung Freigabe
MSG_STATE_HIDE	Meldung ausgeblendet
MSG_STATE_UNHIDE	Meldung eingeblendet
MSG_STATE_HIDE_MANUAL	Meldung manuell ausgeblendet
MSG_STATE_UNHIDE_MANUAL	Meldung manuell eingeblendet

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

stMsgTime

Datum und Uhrzeit für Kommen oder Gehen einer Meldung.

dwTimeDiff

Zeitdauer kommend bzw. gehend, Telegrammzeit in Sekunden.

dwCounter

Interner Meldungszähler.

dwFlags

Flags entsprechend der Konstanten in der Datei "m_global.h":

Flags für Meldungen in der Meldeanzeige		
MSG_RT_FLAG_COME	0x00000001	Meldung gekommen
MSG_RT_FLAG_GO	0x00000002	Meldung gegangen
MSG_RT_FLAG_QUIT_COME	0x00000004	Quittierung gekommen
MSG_RT_FLAG_QUIT_GO	0x00000008	Quittierung gegangen
MSG_RT_FLAG_NOQUIT	0x00000010	Keine Quittierpflicht
MSG_RT_FLAG_QUIT_NOGO	0x00000020	TM II Quittierung
MSG_RT_FLAG_FLASH	0x00000040	Blinkend
MSG_RT_FLAG_FIRSTFLASH	0x00000080	Nur erste Meldung darf blinken
MSG_RT_FLAG_DELETE	0x00010000	Meldung ist löschtbar
MSG_RT_FLAG_SUMTIME	0x00020000	Sommerzeit-Flag
MSG_RT_FLAG_SQUIT	0x00040000	Meldung hat Sammelquittierung
MSG_RT_FLAG_QUITNR	0x00080000	Meldungsnummer - Quit
MSG_RT_FLAG_MSGLIST	0x00100000	Meldung von Meldeliste
MSG_RT_FLAG_COMMENT	0x00200000	Meldung hat Kommentar
MSG_RT_FLAG_ARCHIV	0x00400000	Meldung wird archiviert
MSG_RT_FLAG_PROTOCOL	0x00800000	Meldung wird protokolliert
MSG_RT_FLAG_INSTANCE	0x01000000	Flag für Instanzmeldung
MSG_RT_FLAG_TIMEINVALID	0x02000000	Flag für ungültigen Datum/Uhrzeitstempel
MSG_RT_FLAG_LOCKGROUP	0x04000000	Flag für Gruppensperre
MSG_RT_FLAG_INSTANCE10	0x08000000	Bit für neue Instanzmeldung mit Prozesswert 10
MSG_RT_FLAG_TEXTREF1	0x10000000	Bit für Textreferenz 1
MSG_RT_FLAG_TEXTREF2	0x20000000	Bit für Textreferenz 2
MSG_RT_FLAG_QUITCOUNTER	0x40000000	Bit für Quittierung über Counter
MSG_RT_FLAG_QUITGROUP	0x80000000	Bit für Quittierung über Counter

Bei der Funktion MSRTCreateMsgInstanceWithCommentPlus können keine Flags an den Alarmserver weitergegeben werden. Flags werden ignoriert.

wPValueUsed

Benutzte Prozesswerte in bitcodierter Darstellung.

Jedes Bit darf nur in einem der beiden Member wPValueUsed oder wTextValueUsed gesetzt werden, da technisch ein Begleitwert nur Zahl oder Text sein kann. Bei fehlerhafter Doppelversorgung hat dPValue Vorrang.

wTextValueUsed

Benutzte Textwerte, bitcodiert.

Jedes Bit darf nur in einem der beiden Member wPValueUsed oder wTextValueUsed gesetzt werden, da technisch ein Begleitwert nur Zahl oder Text sein kann.

Bei fehlerhafter Doppelversorgung hat dPValue Vorrang.

dPValue

Prozesswerte.

mtTextValue

Struktur vom Typ MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046) mit den Textbegleitwerten.

szInstance

Instanzname. Wird szInstance nicht versorgt (Leerstring), so wird eine normale Meldung angelegt

szComment

Kommentarname.

szUser

Username.

szComputerName

Computername.

szApplicationName

Applikationsname.

szServerPrefix

Serverpräfix.

Bemerkung

Bevor Sie die Struktur ausfüllen, sollten Sie sie mit "0" initialisieren.

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)	Meldung erzeugen
MSRTGetLastMsgWithCommentPlus (Seite 1075)	Letzte archivierte Meldung abfragen
MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Callback-Funktion

Siehe auch

MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)
 MSRTGetLastMsgWithCommentPlus (Seite 1075)
 MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)
 MSRTGetSelectedMsgPlus (Seite 1084)
 MSRTCheckWinFilterPlus (Seite 1095)
 Übersicht über die Strukturen (Seite 1007)

2.10.2.3 MSG_FILTER_STRUCT_PLUS (RT Professional)

Deklaration

```

typedef struct {
    WCHAR          szFilterName[MSG_MAX_TEXTLEN+1];
    DWORD          dwFilter;
    SYSTEMTIME     st[2];
    DWORD          dwMsgNrLow[2];
    DWORD          dwMsgNrHigh[2];
    VARIANT        vMsgClasses;
    DWORD          dwMsgState;
    WORD           wAGNr[2];
    WORD           wAGSubNr[2];
    DWORD          dwArchivMode;
    WCHAR          szTB[MSG_MAX_TB][MSG_MAX_TB_CONTENT+1];
    DWORD          dwTB;
    double         dPValue[MSG_MAX_PVALUE][2];
    DWORD          dwPValue[2];
    ULONGLONG     uMsgCounter[2];
    DWORD          dwQuickSelect;
    DWORD          dwPriority[2];
    WCHAR          szInstance[MSG_MAX_INSTANCE+1];
}
MSG_FILTER_STRUCT_PLUS;
  
```

Members

szFilterName

Filtername

dwFilter

Die Filterbedingungen werden über die folgenden Konstanten aus der Datei "m_global.h" festgelegt:

MSG_FILTER_DATE_FROM	Datum von
MSG_FILTER_DATE_TO	Datum bis
MSG_FILTER_TIME_FROM	Uhrzeit von
MSG_FILTER_TIME_TO	Uhrzeit bis
MSG_FILTER_NR_FROM	Meldungsnummer von
MSG_FILTER_NR_TO	Meldungsnummer bis
MSG_FILTER_CLASS	Meldungsklassen
MSG_FILTER_STATE	Meldungsstatus
MSG_FILTER_AG_FROM	AG - Nummer von
MSG_FILTER_AG_TO	AG - Nummer bis
MSG_FILTER_AGSUB_FROM	AG - Subnummer von
MSG_FILTER_AGSUB_TO	AG - Subnummer bis
MSG_FILTER_TEXT	Meldungstexte
MSG_FILTER_PVALUE	Prozesswerte
MSG_FILTER_COUNTER_FROM	Interner Meldungsanzähler von
MSG_FILTER_COUNTER_TO	Interner Meldungsanzähler bis
MSG_FILTER_PROCESSMSG	Prozessmeldungen (reserviert)
MSG_FILTER_SYSMMSG	Systemmeldungen (reserviert)
MSG_FILTER_BEDMSG	Bedienmeldungen (reserviert)
MSG_FILTER_VISIBLEONLY	Sichtbare Meldungen anzeigen (wird nicht in die Filterbedingung mit einbezogen)
MSG_FILTER_HIDDENONLY	Versteckte Meldungen anzeigen (wird nicht in die Filterbedingung mit einbezogen)
MSG_FILTER_NODYNAMIC	Keine Selektion auf dynamische Texte (wird nicht in die Filterbedingung mit einbezogen)
MSG_FILTER_DATE	Datum von bis
MSG_FILTER_TIME	Uhrzeit von bis
MSG_FILTER_NR	Meldungsnummer von bis

Die einzelnen Filterbedingungen können verodert werden.

st

st[0] Datum und Uhrzeit des Startzeitpunktes, st[1] Datum und Uhrzeit des Endzeitpunktes

Sie müssen st belegen, wenn Sie die Filterkriterien MSG_FILTER_DATE, MSG_FILTER_DATE_FROM, MSG_FILTER_DATE_TO, MSG_FILTER_TIME, MSG_FILTER_TIME_FROM, oder MSG_FILTER_TIME_To verwenden.

Wird eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt, so ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgNrHigh Nummer der letzten Meldung.

Sie müssen dwMsgNrLow und dwMsgNrHigh belegen, wenn Sie die Filterkriterien MSG_FILTER_NR, MSG_FILTER_NR_FROM oder MSG_FILTER_NR_TO verwenden.

vMsgClasses

Meldungsklassen bitcodiert

0x0001 = Klasse 1, 0x0002 = Klasse 2, 0x0004 = Klasse 3, ... 0x0040=Klasse 7

0x0204 = Klassen 3 und 10

Sie müssen dwMsgClass und dwMsgType belegen, wenn Sie das Filterkriterium MSG_FILTER_CLASS verwenden.

dwMsgState

Meldungsstatus bitcodiert

Belegen Sie dieses Feld, wenn Sie das Filterkriterium MSG_FILTER_STATE verwenden.

Folgende Statuswerte werden von MSRTCheckWinFilter geprüft:

MSG_STATE_COME	Meldung gekommen
MSG_STATE_GO	Meldung gegangen
MSG_STATE_QUIT	Meldung quittiert
MSG_STATE_QUIT_SYSTEM	Meldung quittiert durch System
MSG_STATE_QUIT_EMERGENCY	Not-Quittierung

Andere Statuswerte, z.B. MSG_STATE_HIDE, MSG_STATE_UNHIDE, müssen Sie selbst filtern, da diese nicht in der Filterbedingung berücksichtigt werden. Verwenden Sie zum Filtern dieser Statuswerte z.B. MSG_FILTER_HIDDENONLY.

wAGNr

wAGNr [0] Nummer der ersten PLC, wAGNr [1] Nummer der letzten PLC.

Sie müssen wAGNr belegen, wenn Sie die Filterkriterien MSG_FILTER_AG_FROM oder MSG_FILTER_AG_TO verwenden.

wAGSubNr

wAGSubNr [0] Nummer der ersten PLC, wAGSubNr [1] Nummer der letzten PLC.

Sie müssen wAGSubNr belegen, wenn Sie die Filterkriterien MSG_FILTER_AGSUB_FROM oder MSG_FILTER_AGSUB_TO verwenden.

dwArchivMode

Kennung für Archiv oder Protokoll. Der Parameter muss mit 0 belegt werden.

szTB

Texte der Textblöcke

Sie müssen szTB belegen, wenn Sie das Filterkriterium MSG_FILTER_TEXT verwenden.

dwTB

Aktive Textblöcke, bitcodiert

Sie müssen dwTB belegen, wenn Sie das Filterkriterium MSG_FILTER_TEXT benutzen

dPValue

ProzessWerte von - bis

Sie müssen dPValue belegen, wenn Sie das Filterkriterium MSG_FILTER_PVALUE verwenden.

dwPValue

Aktive Prozesswerte, bitcodiert

Sie müssen dwPValue belegen, wenn Sie das Filterkriterium MSG_FILTER_PVALUE verwenden.

uMsgCounter

Interner MeldungsZähler

Sie müssen dwMsgCounter belegen, wenn Sie ein Filterkriterium MSG_FILTER_COUNTER_... verwenden.

dwQuickSelect

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

dwPriority

Priorität der Meldung

szInstance

Instanzname. Wird szInstance nicht versorgt (Leerstring), so wird eine normale Meldung angelegt.

Bemerkung

MSG_FILTER_STRUCT_PLUS wird auch von der Struktur AUTOHOTSPOT verwendet.

Benötigte Dateien

CCMsRTCliPlus.h

API - Funktionen

MSRTCheckWinFilterPlus (Seite 1095)	Meldungsfilterbedingungen überprüfen
AUTOHOTSPOT	Meldungsfilter parametrieren
MSRTEnumProtDataPlus (Seite 1060)	Protokolldaten auflisten

MSRTGetFilterDataPlus (Seite 1094)	Meldungsfilterbedingungen abfragen
MSRTSetMsgFilterPlus (Seite 1097)	Meldungsfilter setzen
MSRTSetMsgWinFilterPlus (Seite 1098)	Meldungsfilter für ein Meldefenster setzen
MSRTGetLastMsgWithCommentPlus (Seite 1075)	Letzte archivierte Meldung abfragen
MSRTStartMsgServicePlus (Seite 1049)	dienst starten

Siehe auch

MSRTGetLastMsgWithCommentPlus (Seite 1075)

MSRTSetMsgWinFilterPlus (Seite 1098)

MSRTStartMsgServicePlus (Seite 1049)

MSRTSetMsgFilterPlus (Seite 1097)

MSRTCheckWinFilterPlus (Seite 1095)

MSRTGetFilterDataPlus (Seite 1094)

MSRTEnumProtDataPlus (Seite 1060)

Übersicht über die Strukturen (Seite 1007)

2.10.2.4 MSG_RTDATA_INSTANCE_STRUCT_PLUS (RT Professional)**Deklaration**

```
typedef struct {
    DWORD                dwMsgState;
    DWORD                dwMsgNrLow;
    DWORD                dwMsgNrHigh;
    SYSTEMTIME           stMsgTime;
    DWORD                dwTimeDiff;
    ULONGLONG           ullCounter;
    DWORD                dwFlags;
    WORD                wPValueUsed;
    WORD                wTextValueUsed;
    double               dpValue[MSG_MAX_PVALUE];
    MSG_TEXTVAL256_STRUCT_PLUS mtTextValue[MSG_MAX_PVALUE];
    TCHAR                szInstance[MSG_MAX_INSTANCE+1];
}
MSG_RTDATA_INSTANCE_STRUCT_PLUS;
```

Members**dwMsgState**

Meldungsstatus entsprechend der Konstanten in der Datei "m_global.h":

Die aufgeführten Stati werden nur intern vom System benutzt und können vom API nur mit `MSRTCreateMsgInstanceWithCommentPlus` gesetzt werden.

MSG_STATE_COME	Meldung gekommen
MSG_STATE_GO	Meldung gegangen
MSG_STATE_QUIT	Meldung quittiert
MSG_STATE_LOCK	Meldung gesperrt
MSG_STATE_UNLOCK	Meldung Freigabe
MSG_STATE_HIDE	Meldung ausgeblendet
MSG_STATE_UNHIDE	Meldung eingeblendet
MSG_STATE_HIDE_MANUAL	Meldung manuell ausgeblendet

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

stMsgTime

Datum und Uhrzeit für Kommen oder Gehen einer Meldung.

dwTimeDiff

Zeitdauer kommend bzw. gehend, Telegrammzeit in Sekunden.

ullCounter

Interner Meldungs-zähler.

dwFlags

Flags entsprechend der Konstanten in der Datei "m_global.h":

Flags für Meldungen in der Meldeanzeige		
MSG_RT_FLAG_COME	0x00000001	Meldung gekommen
MSG_RT_FLAG_GO	0x00000002	Meldung gegangen
MSG_RT_FLAG_QUIT_COME	0x00000004	Quittierung gekommen
MSG_RT_FLAG_QUIT_GO	0x00000008	Quittierung gegangen
MSG_RT_FLAG_NOQUIT	0x00000010	Keine Quittierpflicht
MSG_RT_FLAG_QUIT_NOGO	0x00000020	TM II Quittierung
MSG_RT_FLAG_FLASH	0x00000040	Blinkend
MSG_RT_FLAG_FIRSTFLASH	0x00000080	Nur erste Meldung darf blinken
MSG_RT_FLAG_DELETE	0x00010000	Meldung ist löschar
MSG_RT_FLAG_SUMTIME	0x00020000	Sommerzeit-Flag
MSG_RT_FLAG_SQUIT	0x00040000	Meldung hat Sammelquittierung
MSG_RT_FLAG_QUITNR	0x00080000	Meldungsnummer - Quit
MSG_RT_FLAG_MSGLIST	0x00100000	Meldung von Meldeliste
MSG_RT_FLAG_COMMENT	0x00200000	Meldung hat Kommentar
MSG_RT_FLAG_ARCHIV	0x00400000	Meldung wird archiviert

Flags für Meldungen in der Meldeanzeige		
MSG_RT_FLAG_PROTOCOL	0x00800000	Meldung wird protokolliert
MSG_RT_FLAG_INSTANCE	0x01000000	Flag für Instanzmeldung
MSG_RT_FLAG_TIMEINVALID	0x02000000	Flag für ungültigen Datum/Uhrzeitstempel
MSG_RT_FLAG_LOCKGROUP	0x04000000	Flag für Gruppensperre
MSG_RT_FLAG_INSTANCE10	0x08000000	Bit für neue Instanzmeldung mit Prozesswert 10
MSG_RT_FLAG_TEXTREF1	0x10000000	Bit für Textreferenz 1
MSG_RT_FLAG_TEXTREF2	0x20000000	Bit für Textreferenz 2
MSG_RT_FLAG_QUITCOUNTER	0x40000000	Bit für Quittierung über Counter
MSG_RT_FLAG_QUITGROUP	0x80000000	Bit für Quittierung über Counter

Flags für Meldungen im Archiv		
MSG_FLAG_SUMTIME	0x00000001	Sommerzeit aktiv
MSG_FLAG_COMMENT	0x00000002	Meldung hat Kommentar
MSG_FLAG_ARCHIV	0x00000004	Archivieren
MSG_FLAG_PROTOCOL	0x00000008	Protokollieren
MSG_FLAG_TEXTVALUES	0x00000010	Meldung hat Textbegleitwerte
MSG_FLAG_TIMEINVALID	0x00000020	Bit für ungültigen Datum/Uhrzeit-Stempel
MSG_FLAG_INSTANCE	0x00000040	Kennzeichnung von Instanzmeldungen
MSG_FLAG_INSTANCE10	0x08000000	Instanzmeldung mit Prozesswert10, verfügbar ab V5.0 SP2
MSG_FLAG_TEXTREF1	0x10000000	Textreferenz1
MSG_FLAG_TEXTREF2	0x20000000	Textreferenz2

Bei der Funktion `MSRTCreateMsgInstanceWithCommentPlus` können keine Flags an den Alarmserver weitergegeben werden. Flags werden ignoriert.

wPValueUsed

Benutzte Prozesswerte in bitcodierter Darstellung.

Jedes Bit darf nur in einem der beiden Member `wPValueUsed` oder `wTextValueUsed` gesetzt werden, da technisch ein Begleitwert nur Zahl oder Text sein kann. Bei fehlerhafter Doppelversorgung hat `dPValue` Vorrang.

wTextValueUsed

Benutzte Textwerte, bitcodiert.

Jedes Bit darf nur in einem der beiden Member `wPValueUsed` oder `wTextValueUsed` gesetzt werden, da technisch ein Begleitwert nur Zahl oder Text sein kann.

Bei fehlerhafter Doppelversorgung hat `dPValue` Vorrang.

dPValue

Prozesswerte.

mtTextValue

Struktur vom Typ MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046) mit den Textbegleitwerten.

szInstance

Instanzname. Wird szInstance nicht versorgt (Leerstring), so wird eine normale Meldung angelegt

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTCreateMsgInstancePlus (Seite 1066)	Meldung erzeugen
MSG_SERVICE_NOTIFY_PROC	Sende- und Empfangsservice (Callback)

Siehe auch

MSRTCreateMsgInstancePlus (Seite 1066)

MSG_SERVICE_NOTIFY_PROCPPlus (Seite 1055)

MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046)

Übersicht über die Strukturen (Seite 1007)

2.10.2.5 MSG_CLASS_STRUCT_PLUS (RT Professional)

Deklaration

```
typedef struct {
    WCHAR          szName[MSG_MAX_TEXTLEN+1];
    DWORD          dwName;
    DWORD          dwClassID;
    COLORREF       crStateCome[2];
    COLORREF       crStateGo[2];
    COLORREF       crStateQuit[2];
    WORD           wQuitType;
    WORD           wHornQuit;
    DWORD          dwHornVar;
    WCHAR          szState[MSG_MAX_STATE][MSG_MAX_TEXTLEN+1];
    DWORD          dwState[MSG_MAX_STATE];
    WCHAR          szStateQuit[MSG_MAX_TEXTLEN+1];
    DWORD          dwStateQuit;
    DWORD          dwClassTyp;
    DWORD          dwCreatorID;
}
MSG_CLASS_STRUCT_PLUS;
```

Members

szName

Der anzugebende Klassenname ist abhängig von der eingestellten Datensprache !

dwName

Wenn der Klassenname in den Projekttexten enthalten ist kann hier die ID angegeben werden.

dwClassID

ID der Meldeklasse (1..18(MSG_MAX_CLASS))

crStateCome

Farbe für den Zustand "Gekommen".

crStateGo

Farbe für den Zustand "Gegangen"

crStateQuit

Farbe für den Zustand "Quittiert".

wQuitType

Der Zustandsautomat wird über die folgenden Konstanten aus der Datei "m_global.h" festgelegt:

MSG_QUIT_QUIT_COME	Quittierung gekommen
MSG_QUIT_QUIT_GO	Quittierung gegangen
MSG_QUIT_NOQUIT	Keine Quittierpflicht
MSG_QUIT_QUIT_NOGO	Quittierung (ohne gehend Meldung)
MSG_QUIT_FLASH	Blinkend
MSG_QUIT_FIRSTFLASH	Nur erste Meldung darf blinken
MSG_QUIT_NOLIST	Meldungen nicht in Meldeliste aufnehmen

Die Konstanten des Zustandsautomaten können Sie, sofern sie sich nicht gegenseitig widersprechen, mit "oder" bitweise verknüpfen.

wHornQuit

Quittierung Zentraler Melder

dwHornVar

Variable Zentraler Melder

szState

Text für die Zustände "Gekommen", "Gegangen", "Gekommen und Gegangen".

dwState

TextID

szStateQuit

Text für den Zustand "Quittiert".

dwStateQuit

TextID

dwClassTyp

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

dwCreatorID

CreatorID

Bemerkung

Im Unterschied zur Struktur MSG_CLASS_STRUCT_EX verfügt MSG_CLASS_STRUCTPLUS über keine Erzeugerkennung.

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTGetClassInfoPlus (Seite 1073)	Meldeklassen-Information abfragen
-----------------------------------	-----------------------------------

Siehe auch

MSRTGetClassInfoPlus (Seite 1073)

Übersicht über die Strukturen (Seite 1007)

2.10.2.6 MSG_CSDATA_STRUCT_PLUS (RT Professional)

Deklaration

```
typedef struct {
    DWORD    dwMsgNrLow;
    DWORD    dwMsgNrHigh;
    DWORD    dwStatus;
    WORD     wClass;
    WORD     wTyp;
    DWORD    dwTextID[MSG_MAX_TB];
    DWORD    dwQuitVar;
    WORD     wQuitBit;
    DWORD    dwStateVar;
    WORD     wStateBit;
    DWORD    dwMsgVar;
    WORD     wMsgBit;
    WORD     wAGNr;
    WORD     wAGSubNr;
    DWORD    dwGroupID;
    DWORD    dwPriority;
    DWORD    dwHidingMask;
}
MSG_CSDATA_STRUCT_PLUS;
```

Members

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

dwStatus

Die Parameter der Einzelmeldung werden über die folgenden Konstanten aus der Datei "m_global.h" festgelegt:

MSG_PARAMS_SQUIT	Einzelquittierpflichtig
MSG_PARAMS_HORN	Steuert Hupe an
MSG_PARAMS_ARCHIV	Wird archiviert
MSG_PARAMS_PROTOCOL	Wird protokolliert
MSG_PARAMS_VARFLANK	Wird bei negativer Flanke erzeugt
MSG_PARAMS_INFOTEXT	Hat Infotext
MSG_PARAMS_LOOPINALARM	Hat Loop In Alarm
MSG_PARAMS_NORMDLL	Hat Normierungs - DLL

Sie können diese Konstanten, sofern sie sich nicht gegenseitig widersprechen, verodern.

wClass

Meldungsklasse

wTyp

Meldungsart

dwTextID

Referenz auf die Textblöcke

dwQuitVar

Quittiervariable. Verwenden Sie die dwID aus DM_VARKEY (Seite 241).

wQuitBit

Bit der Quittiervariable

dwStateVar

Meldestatus als Variable. Verwenden Sie die dwID aus DM_VARKEY (Seite 241)

wStateBit

Über wStateBit wird festgelegt welche Bits von dwStateVar den Status der Meldung charakterisieren.

Ist wStateBit = 1 und dwStateVar vom Datentyp "8 Bit unsigned", so legt Bit 1 von dwStateVar den Gekommen/Gegangen-Status fest, Bit 5 charakterisiert die Quittierpflicht.

Ist wStateBit = 2 und dwStateVar eine Wort-Variable, so legt Bit 2 von dwStateVar den Gekommen/Gegangen-Status fest, Bit 10 charakterisiert die Quittierpflicht.

dwMsgVar

Meldevariable. Verwenden Sie die dwID aus DM_VARKEY (Seite 241).

wMsgBit

Bit der Meldevariable

wAGNr

PLC-Nummer

wAGSubNr

PLC-Subnummer

dwGroupID

ID der Meldungsgruppe (Intern)

dwPriority

Priorität der Meldung

dwHidingMask

Anzeigeunterdrückungsmaske

Bemerkung

Über diese Struktur kann Ihre Applikation unter Verwendung der API-Funktionen die Projektierungsdaten einer Einzelmeldung ermitteln.

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTGetMsgCSDDataPlus (Seite 1078)	Meldung-Projektierungsdaten abfragen
------------------------------------	--------------------------------------

Siehe auch

MSRTGetMsgCSDDataPlus (Seite 1078)

DM_VARKEY (Seite 241)

Übersicht über die Strukturen (Seite 1007)

2.10.2.7 MSG_RTGROUPENUM_STRUCT_PLUS (RT Professional)**Deklaration**

```
typedef struct {
    BOOL      fIDUsed;
    DWORD     dwID;
    TCHAR     szName[MSG_MAX_TEXTLEN+1];
    DWORD     dwMsgCount;
    DWORD     dwMsg[MSG_MAX_GROUPITEMS];
}
MSG_RTGROUPENUM_STRUCT_PLUS;
```

Members**fIDUsed**

Legt fest, ob zur Spezifikation der Meldungsgruppe die ID der Gruppe (fIDUsed = TRUE) oder der Name (fIDUsed = FALSE) benutzt wird.

dwID

ID der Gruppe

szName

Name der Gruppe

dwMsgCount

Anzahl der Einzelmeldungen in dwMsg

dwMsg

Gesperrte Einzelmeldungen

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTEnumGroupMsgPlus (Seite 1089)	Meldungsgruppe Einzelmeldungen auflisten
MSG_SERVICE_NOTIFY_PROCPPlus (Seite 1055)	Callback-Funktion

Siehe auch

MSG_SERVICE_NOTIFY_PROCPPlus (Seite 1055)

MSRTEnumGroupMsgPlus (Seite 1089)

Übersicht über die Strukturen (Seite 1007)

2.10.2.8 MSG_RTGROUPSET_STRUCT_PLUS (RT Professional)

Deklaration

```
typedef struct {  
    BOOL        fIDUsed;  
    DWORD       dwID;  
    WCHAR       szName[MSG_MAX_TEXTLEN+1];  
    SYSTEMTIME  stTime;  
    DWORD       dwData;  
}  
MSG_RTGROUPSET_STRUCT_PLUS;
```

Members

fIDUsed

Legt fest, ob zur Spezifikation der Meldegruppe die ID der Gruppe (fIDUsed = TRUE) oder der Name (fIDUsed = FALSE) benutzt wird.

dwID

ID der Gruppe (bei Meldeklassen, Meldearten)

szName

Name der Gruppe

stTime

Zeitpunkt des Auftretens

dwData

Wenn die Struktur im Rahmen von MSRTLockGroupPlus verwendet wird, bedeutet dwData = 1 Sperren und dwData = 0 Freigeben

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTLockGroupPlus (Seite 1091)	Meldegruppe sperren
MSRTEnumGroupMsgPlus (Seite 1089)	Einzelmeldungen einer Meldegruppe auflisten
MSRTQuitGroupPlus (Seite 1092)	Meldegruppe quittieren

Siehe auch

MSRTEnumGroupMsgPlus (Seite 1089)

MSRTLockGroupPlus (Seite 1091)

MSRTQuitGroupPlus (Seite 1092)

Übersicht über die Strukturen (Seite 1007)

2.10.2.9 MSG_COMMENT_INSTANCE_STRUCT_PLUS (RT Professional)**Deklaration**

```
typedef struct {
    SYSTEMTIME    stTime;
    DWORD         dwMsgNrLow;
    DWORD         dwMsgNrHigh;
    WCHAR         szText[MSG_MAX_TB_CONTENT+1];
    WCHAR         szUser[MSG_MAX_USERNAME+1];
    WCHAR         szComputerName[MAX_COMPUTERNAME_LENGTH+1];
    WCHAR         szApplicationName[MSG_MAX_APPLNAME+1];
    WCHAR         szInstance[MSG_MAX_INSTANCE+1];
}
MSG_COMMENT_INSTANCE_STRUCT_PLUS;
```

Members

stTime

Datum / Uhrzeit

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

szText

Kommentar

szUser

Benutzername

szComputerName

Rechnername

szApplicationName

Applikationsname

szInstance

Instanzname

Bemerkung

Wird eine aktuelle Zeit für die Übergabe eines SYSTEMTIME -Parameters benötigt, so ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTGetCommentInstancePlus (Seite 1102)	Kommentartext abfragen
MSRTSetCommentInstancePlus (Seite 1103)	Kommentartext festlegen

Siehe auch

MSRTSetCommentInstancePlus (Seite 1103)

MSRTGetCommentInstancePlus (Seite 1102)

Übersicht über die Strukturen (Seite 1007)

2.10.2.10 MSG_INFOTEXT_STRUCT_PLUS (RT Professional)**Deklaration**

```
typedef struct {
    DWORD    dwMsgNrLow;
    DWORD    dwMsgNrHigh;
    WCHAR    szText[MSG_MAX_TB_CONTENT+1];
}
MSG_INFOTEXT_STRUCT_PLUS;
```

Members**dwMsgNrLow**

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

szText

Infotext

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTGetInfotextPlus (Seite 1105)	Infotext abfragen
MSRTSetInfotextPlus (Seite 1106)	Infotext festlegen

Siehe auch

MSRTGetInfotextPlus (Seite 1105)

MSRTSetInfotextPlus (Seite 1106)

Übersicht über die Strukturen (Seite 1007)

2.10.2.11 MSG_BACKUP_STRUCT_PLUS (RT Professional)

Deklaration

```
typedef struct {  
    SYSTEMTIME    stFrom;  
    SYSTEMTIME    stTo;  
    WCHAR         szArchivFile[MAX_PATH+1];  
    WCHAR         szArchivTextFile[MAX_PATH+1];  
    WCHAR         szCommentFile[MAX_PATH+1];  
    WCHAR         szInstanceFile[MAX_PATH+1];  
    WCHAR         szComment[MAX_PATH+1];  
    DWORD         dwFlags;  
    DWORD         dwFormat;  
}  
MSG_BACKUP_STRUCTPlus;
```

Members

stFrom

Datum von

stTo

Datum bis

szArchivFile

Dateiname und Pfad der Auslagerungsdatei für Archiv

szArchivTextFile

Dateiname und Pfad der Auslagerungsdatei für Archiv-Texte

szCommentFile

Dateiname und Pfad der Auslagerungsdatei für Kommentare

szInstanceFile

Dateiname und Pfad der Auslagerungsdatei für Instanzen

szComment

Frei definierbarer Kommentar

dwFlags

MSG_DB_FLAGS_OVERWRITE	0x0001	Vorhandene Datensätze überschreiben
MSG_DB_FLAGS_ADD	0x0002	Nur hinzufügen
MSG_DB_FLAGS_DELETE	0x0004	Quelldatensätze löschen

dwFormat

Für das Format der Exportdatei können Sie die folgende Konstanten aus der Datei "m_global.h" verwenden:

MSG_DB_FMT_CSV	CSV Dateiformat
MSG_DB_FMT_DBASE3	DBASE3 - Dateiformat
MSG_DB_FMT_WATCOM	Sybase - Format

Bemerkung

Wird eine aktuelle Zeit für die Übergabe eines SYSTEMTIME-Parameters benötigt, so ist die Funktion GetLocalTime zu verwenden und nicht GetSystemTime. Es besteht in der Regel ein beträchtlicher Zeitunterschied zwischen diesen beiden Funktionen.

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTEnumBackupListPlus (Seite 1108)	Einträge einer Auslagerungsdatei auflisten
MSRTGetBackupSize	Auslagerungsdatei Größe ermitteln
MSRTRestore	Meldungen importieren

Siehe auch

MSRTEnumBackupListPlus (Seite 1108)

MSRTExportPlus (Seite 1109)

Übersicht über die Strukturen (Seite 1007)

2.10.2.12 MSG_RTLOCK_STRUCT_PLUS (RT Professional)**Deklaration**

```
typedef struct {
    WCHAR    szName[MSG_MAX_TEXTLEN+1];
    WCHAR    szParent[MSG_MAX_TEXTLEN+1];
    DWORD    dwLockID;
    BOOL     fLock;
    DWORD    dwMsgNum;
    DWORD    dwMsgLock[MSG_MAX_LOCKITEMS];
}
MSG_RTLOCK_STRUCT_PLUS;
```

Members

- szName**
Name der Sperre
- szParent**
Vatergruppe
- dwLockID**
ID der Sperre
- fLock**
Sperrern bzw. freigeben
- dwMsgNum**
Anzahl gesperrter Meldungen
- dwMsgLock**
Gesperrte Einzelmeldungen

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Callback-Funktion
--	-------------------

Siehe auch

- MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)
- Übersicht über die Strukturen (Seite 1007)

2.10.2.13 MSG_TEXTVAL256_STRUCT_PLUS (RT Professional)

Deklaration

```
typedef struct {  
    WCHAR szText[MSG_MAX_TEXTVAL256+1];  
}  
MSG_TEXTVAL256_STRUCT_PLUS;
```

Bemerkung

Die Struktur wird innerhalb der folgenden Strukturen verwendet:

Members

szText

Text

Benötigte Dateien

CCMSRTCLIPlus.h

API - Funktionen

MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)	Meldung anlegen
MSRTGetLastMsgWithCommentPlus (Seite 1075)	Letzte anstehende Meldung abfragen
MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Sende- und Empfangsservice (Callback)

Siehe auch

MSG_RTDATA_INSTANCE_STRUCT_PLUS (Seite 1031)

MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024)

MSG_RTCREATE_STRUCT_PLUS (Seite 1022)

Übersicht über die Strukturen (Seite 1007)

2.10.3 Allgemeine Funktionen (RT Professional)**2.10.3.1 MSRTEnumArchivDataPlus (RT Professional)****Beschreibung**

Enumeriert alle Meldungen des angegebenen Archivs. Ihre Callbackfunktion MSG_SERVICE_NOTIFY_PROCPlus wird mit dem Benachrichtigungstyp MSG_NOTIFY_ARCHIVENUM aufgerufen. lpyData verweist auf Daten der Struktur MSG_RTARCHIV_STRUCTPlus.

Enumerationsfunktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe über Semaphoren synchronisieren müssen.

Deklaration

```
BOOL MSRTEnumArchivDataPlus (  
    DWORD          dwServiceID,  
    BOOL           fArchiv,  
    DWORD          dwMaxRecords  
    DWORD          dwParams  
    LPDWORD        lpdwTAID  
    LPCMN_ERROR    lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Send- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

fArchiv

Kennzeichnet den Archivtyp.

TRUE	Folgearchiv
------	-------------

FALSE	Umlaufarchiv
-------	--------------

dwMaxRecords

Anzahl der Datensätze die maximal aufgelistet werden sollen. Bei Überschreiten von dwMaxRecords wird die Enumeration abgebrochen. Ist dwMaxRecords = 0xFFFFFFFF, so werden alle Meldungen aufgelistet.

dwParams

Parameter für die Sortierreihenfolge:

MSG_ARCHIV_ENUM_DESC	Sortierung absteigend
----------------------	-----------------------

lpdwTAID

Zeiger auf TransactionID bei asynchroner Bearbeitung.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Meldungen aufgelistet.

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
-------------------	--------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Callback-Funktion
MSRTStartMsgServicePlus (Seite 1049)	Dienst starten

Siehe auch

MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)

MSRTStartMsgServicePlus (Seite 1049)

Übersicht über die Funktionen (Seite 1006)

2.10.3.2 MSRTStartMsgServicePlus (RT Professional)**Beschreibung**

Startet einen Dienst zum Senden und Empfangen von Nachrichten an das Meldesystem.

Deklaration

```

BOOL MSRTStartMsgService (
    LPDWORD                lpdwServiceID,
    LPWSTR                lpszServer,
    DWORD                 dwFlags,
    DWORD                 dwFilterCount
    LPMSG_FILTER_STRUCT_PLUS* lpMsgFilter,
    DWORD                 dwNotifyMask,
    MSG_SERVICE_NOTIFY_PROC_PLUS lpfnNotifyProcPlus
    LPVOID                lpvUserNotify,
    MSG_TAID_COMPLETION_PROC_PLUS lpfnTAIDProcPlus
    LPVOID                lpvUserTAID
    LPCMN_ERRORW          lpError );

```

Parameter

dwServiceID

Enthält, nach erfolgreichem Aufruf der Funktion, die Identifikationsnummer des Send- und Empfangsdienstes, die von vielen Funktionen benötigt wird.

lpszServer

Server-Präfix

dwFlags

Der Parameter ist für spätere Erweiterungen reserviert und muss mit 0 vorbesetzt werden.

dwFilterCount

Zeiger auf die Filterbedingungen(NULL = Alle Meldungen)

lpMsgFilter

Zeiger auf die Daten des Meldefilters in der Struktur MSG_FILTER_STRUCT_PLUS (Seite 1027).

dwNotifyMask

Kennzeichnet die Art der Nachricht, gemäß der Konstantendefinition in der Datei "msrtapi.h":

MSG_NOTIFY_MASK_MSGLIST	Nachrichten über Meldung
MSG_NOTIFY_MASK_LOCK	Nachrichten über Sperren
MSG_NOTIFY_MASK_ARCHIV	Nachrichten über archivierte Meldungen
MSG_NOTIFY_MASK_ALL	alle Arten von Nachrichten

lpfnNotifyProcPlus

Zeiger auf Ihre Callback-Funktion vom Typ MSG_SERVICE_NOTIFY_PROCPPlus. zur Übergabe der Nachrichten durch den Service.

Mit `IpfnNotifyProcPlus = NULL` werden vom Service keine Nachrichten zurückgeliefert,

Hinweis

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

IpvUserNotify

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpfnTAIDProcPlus

Rückruffunktion für asynchrone Bearbeitung Rückmeldung.

IpvUserTAID

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

IpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Service gestartet.

FALSE

Fehler

Bemerkung

Ihre Applikation kann maximal 16 Dienste installieren. Insgesamt können bis zu 128 Dienste installiert werden.

Fehlermeldungen

<code>MSG_ERR_RT_NOTCONNECTED</code>	Verbindung zum Meldesystem nicht aufgebaut
<code>MSG_ERR_RT_NOINIT</code>	Meldesystem nicht initialisiert
<code>MSG_ERR_RT_SERVICEMAX</code>	Maximale Anzahl Services erreicht
<code>MSG_ERR_NOSERVER</code>	Kein Standardserver projektiert und kein lokaler-Server verfügbar

Benötigte Dateien

- CCMSRTCLIPlus.h
- CCMSRTCLIPlus.lib
- CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStopMsgServicePlus (Seite 1053)	Dienst stoppen
MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Callback-Funktion

Siehe auch

- MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)
- MSRTCreateMsgInstancePlus (Seite 1066)
- MSRTExportPlus (Seite 1109)
- MSRTEnumBackupListPlus (Seite 1108)
- MSRTGetInfotextPlus (Seite 1105)
- MSRTSetInfotextPlus (Seite 1106)
- MSRTGetClassInfoPlus (Seite 1073)
- MSRTSetMsgFilterPlus (Seite 1097)
- MSG_FILTER_STRUCT_PLUS (Seite 1027)
- MSRTGetMsgCSDDataPlus (Seite 1078)
- MSRTCheckWinFilterPlus (Seite 1095)
- MSRTGetMsgPriorityPlus (Seite 1080)
- MSRTGetMsgQuitPlus (Seite 1081)
- MSRTGetMsgTextPlus (Seite 1083)
- MSRTGetSelectedMsgPlus (Seite 1084)
- MSRTResetMsgPlus (Seite 1086)
- MSRTLoopInAlarmPlus (Seite 1088)
- MSRTGetFilterDataPlus (Seite 1094)
- MSRTEnumArchivDataPlus (Seite 1047)
- MSRTStopMsgServicePlus (Seite 1053)
- MSRTSetCommentInstancePlus (Seite 1103)
- MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)
- MSRTQuitHornPlus (Seite 1054)
- MSRTGetCommentInstancePlus (Seite 1102)

MSRTEnumGroupMsgPlus (Seite 1089)

MSRTLckGroupPlus (Seite 1091)

MSRTGetMsgActualPlus (Seite 1077)

Übersicht über die Funktionen (Seite 1006)

2.10.3.3 MSRTStopMsgServicePlus (RT Professional)

Beschreibung

Stoppt den Dienst zum Senden und Empfangen von Nachrichten.

Deklaration

```
BOOL MSRTStopMsgServicePlus (  
    DWORD          dwServiceID,  
    LPCMN_ERRORW  lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

LPCMN_ERRORW

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Dienst beendet.

FALSE

Fehler

Bemerkung

Hinweis

Der Aufruf darf nicht im Destruktor einer Applikation (EXE, DLL, OCX, ...) benutzt werden, weil das aufgrund Microsoft spezifischer Mechanismen gegebenenfalls zum Hängenbleiben des Aufrufs und damit des Programms führen kann.

Fehlermeldungen

MSG_ERR_RT_NOTCONNECTED	Verbindung zum Meldesystem nicht aufgebaut
MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert

Benötigte Dateien

- CCMSRTCLIPlus.h
- CCMSRTCLIPlus.lib
- CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

- MSRTEnumBackupListPlus (Seite 1108)
- MSRTStartMsgServicePlus (Seite 1049)
- Übersicht über die Funktionen (Seite 1006)

2.10.3.4 MSRTQuitHornPlus (RT Professional)

Beschreibung

Quittiert klassenspezifisch den zentralen Melder.

Deklaration

```
BOOL MSRTQuitHornPlus(  
    DWORD          dwServiceID,  
    LPDWORD        lpdwTAID,  
    LPCMN_ERRORW   lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpdwTAID

Zeiger auf die TransactionID für die asynchrone Bearbeitung.

LPCMN_ERRORW

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Melder quittiert.

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
-------------------	--------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

Übersicht über die Funktionen (Seite 1006)

2.10.3.5 MSG_SERVICE_NOTIFY_PROCPlus (RT Professional)**Beschreibung**

Ihre Callback Funktion für Enumerationsfunktionen. Die Enumerationsfunktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe über Semaphoren synchronisieren müssen.

Deklaration

```

BOOL ( * MSG_SERVICE_NOTIFY_PROC) (
    DWORD      dwNotify,
    LPBYTE     lpbyData,
    DWORD      dwItems,
    LPVOID     lpvUserNotify);
    
```

Parameter

dwNotify

Der Benachrichtigungstyp wird durch die Konstanten in der Datei "MSRTAPI.h" festgelegt:

Allgemeine Benachrichtigungen		lpbyData zeigt auf Struktur
MSG_NOTIFY_CONNECT	Verbindung aufgebaut	-
MSG_NOTIFY_DISCONNECT	Verbindung abgebaut	-
MSG_NOTIFY_RESTART	Meldesystem Runtime neue Daten	-
MSG_NOTIFY_SHUTDOWN	Meldesystem wird beendet	-
MSG_NOTIFY_MSGENUM	Enumeration der Meldeliste	MSG_RTDATA_INSTANCE_STRUCTPlus
MSG_NOTIFY_ARCHIVENUM	Enumeration Archiv Meldung(en)	MSG_RTDATA_STRUCTPlus (in Blöcken)
MSG_NOTIFY_ARCHIVENUMINST	Archiv Meldung(en) Enumeration mit Instanzen	MSG_RTDATA_INSTANCE_STRUCTPlus (in Blöcken)
MSG_NOTIFY_LOCKENUM	Enumeration der Sperren	MSG_LOCK_STRUCTPlus (in Blöcken)
MSG_NOTIFY_WINENUM	Meldefenster - Enumeration	MSG_WIN_STRUCTPlus (in Blöcken)
MSG_NOTIFY_BACKUPENUM	Enumeration der Auslagerungsliste	MSG_BACKUP_STRUCTPlus (in Blöcken)
MSG_NOTIFY_ARCDAYENUM	Enumeration der Tagesliste	MSG_ARCHIVDAY_STRUCTPlus
MSG_NOTIFY_ENUMMAX	Enumeration wurde bei maximaler Anzahl abgebrochen	-
MSG_NOTIFY_LANGUAGESET	Sprachumschaltung erfolgt	-

Einzelmeldungen		lpbyData zeigt auf Struktur
MSG_NOTIFY_MSGADD	Neue Meldung hinzugefügt	MSG_RTDATA_INSTANCE_STRUCTPlus (einzeln)
MSG_NOTIFY_MSGDELETE	Meldung gelöscht	MSG_RTDATA_INSTANCE_STRUCTPlus (einzeln)
MSG_NOTIFY_MSGMODIFY	Meldung geändert	MSG_RTDATA_INSTANCE_STRUCTPlus (einzeln)
MSG_NOTIFY_MSGCOUNT	Aktuelle Anzahl Meldungen	-

Meldungsgruppen		lpbyData zeigt auf Struktur
MSG_NOTIFY_GROUPENUM	Enumeration von Einzelmeldungen in einer Meldungsgruppe	MSG_RTGROUPENUM_STRUCTPlus (in Blöcken)

Sperrungen von Meldungen		lpbyData zeigt auf Struktur
MSG_NOTIFY_LOCKMODIFY	Sperre gesetzt, geändert, gelöscht	MSG_RTGROUPENUM_STRUCTPlus

Archivmeldungen		lpbyData zeigt auf Struktur
MSG_NOTIFY_ARCHIVADD	Archiv Meldung(en) hinzugefügt	MSG_RTDATA_INSTANCE_STRUCTPlus (einzeln)
MSG_NOTIFY_ARCHIVDELETE	Archiv - Meldung(en) gelöscht	MSG_RTDATA_INSTANCE_STRUCTPlus (einzeln)

lpbyData

Je nach Benachrichtigungstyp enthält lpbyData Zeiger auf Daten.

Allgemeine Benachrichtigungen	keine Daten
Einzelmeldungen	Runtime-Daten, abgelegt in Strukturen vom Typ <ul style="list-style-type: none"> • MSG_RTDATA_INSTANCE_STRUCT_PLUS (Seite 1031)
Meldungsgruppen	Daten der Einzelmeldungen in Strukturen vom Typ <ul style="list-style-type: none"> • MSG_RTGROUPENUM_STRUCT_PLUS (Seite 1039)
Sperrungen	Sperrdaten in Strukturen vom Typ <ul style="list-style-type: none"> • MSG_RTLOCK_STRUCT_PLUS (Seite 1045)
Archivmeldungen	Archivdaten in Strukturen vom Typ <ul style="list-style-type: none"> • MSG_RTDATA_INSTANCE_STRUCT_PLUS (Seite 1031)

dwItems

Enthält die Anzahl der in lpbyData zurückgegebenen Objekte. Ist dwItems = 0, dann ist auch lpbyData = NULL. Der Notify wird beendet und die Semaphore wird freigegeben.

lpvUserNotify

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird in der Callback-Funktion wieder zu Verfügung gestellt.

Rückgabewert

TRUE

Benachrichtigung wurde erkannt und verarbeitet.

FALSE

Fehler

Bemerkung

Hinweis

Es sollten hier möglichst nur Daten umkopiert werden. Folgende Arten von Funktionsaufrufen innerhalb der Callback können zu Deadlocks oder Stackoverflow führen:

- Funktionen in denen auf eine Message-Loop zugegriffen wird, z.B.: GetMessage
- API-Funktionen aus derselben DLL
- Enumerationen, die weitere Enumerationen aufrufen

Meldet ein Programm eine Notify-Routine an, so muss es regelmäßig seine Message-Queue leeren. Nicht abgeholte Messages können WinCC-Notifizierungen und damit das gesamte WinCC blockieren.

In seltenen Fällen kann es sogar vorkommen, dass die Notify bereits geliefert wird, bevor der Funktionsaufruf zurückgekehrt ist.

Wird die Callback Funktion als Projektfunktion in Script ausprogrammiert, so hat der asynchrone Aufruf außerdem zur Folge, dass der Funktion nicht der volle Script-Kontext zur Verfügung steht, insbesondere gehen die GetTag...- und SetTag-Aufrufe nicht. Bearbeitungen müssen gegebenenfalls in anderen Aktionen ausgeführt werden, die vom Callback nur getriggert werden.

Benötigte Dateien

CCMSRTCLIPlus.h

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSG_RTDATA_INSTANCE_STRUCT_PLUS (Seite 1031)

MSG_RTGROUPENUM_STRUCT_PLUS (Seite 1039)

MSG_RTLOCK_STRUCT_PLUS (Seite 1045)

MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024)

MSRTStartMsgServicePlus (Seite 1049)

MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046)

MSRTEnumGroupMsgPlus (Seite 1089)

Übersicht über die Funktionen (Seite 1006)

2.10.3.6 MSRTWebClientPlus (RT Professional)

Beschreibung

Initialisiert die DLL für einen WebClient

Deklaration

```
VOID WINAPI MSRTWebClientPlus(  
    LPCWSTR      pszProjectPath,  
    DWORD        dwDataLocale);
```

Parameter

pszProjectPath
Pfad zum WebProjekt

dwDataLocale

Rückgabewert

2.10.4 Funktionen zum Bearbeiten von Protokollen (RT Professional)

2.10.4.1 MSRTActivateMProtPlus (RT Professional)

Beschreibung

Protokollservice zur Ausgabe eines Meldefolgeprotokolls anhalten oder fortsetzen.

Deklaration

```
BOOL MSRTActivateMProtPlus(  
    BOOL          fActive,  
    LPCMN_ERROR   lpError );
```

Parameter

fActive

Legt fest, wie der Protokollservice behandelt werden soll:

TRUE	Protokollservice fortsetzen
FALSE	Protokollservice anhalten

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Funktion erfolgreich durchgeführt.

FALSE

Fehler

Fehlermeldungen

MSG_ERR_PROT_ACTIVE	Protokollservice bereits aktiv
---------------------	--------------------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

2.10.4.2 MSRTEnumProtDataPlus (RT Professional)

Beschreibung

Enumeration der Protokolldaten. Die einzelnen Protokolldaten sind durch Kommas getrennt. Die Funktion wird nicht mehr unterstützt.

Enumerationsfunktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe über Semaphoren synchronisieren müssen.

Deklaration

```

BOOL MSRTEnumProtDataPlus (
    DWORD          dwProtID,
    MSG_PROT_NOTIFY_PROC  lpfnEnum,
    LPMSG_FILTER_STRUCT lpMsgFilter,
    LPVOID         lpvUser,
    LPCMN_ERROR    lpError );

```

Parameter

dwProtService

Service-ID für den Protokolldatenaustausch.

lpfnEnum

Ihre Callback-Funktion, die die Protokolldaten entgegennimmt.

lpMsgFilter

Zeiger auf die Daten des Meldefilters in der Struktur MSG_FILTER_STRUCT_PLUS (Seite 1027).

lpvUser

Zeiger auf applikationsspezifische Daten. Dieser Zeiger wird von der Funktion nicht ausgewertet, sondern in der Callback-Funktion wieder zu Verfügung gestellt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Protokolldaten aufgelistet.

FALSE

Fehler

Bemerkung

WARNUNG

Mit dieser Funktion kann Ihre Applikation Meldungen quittieren, ohne dass sichergestellt wird, dass der Anlagenbediener diese zur Kenntnis genommen hat. Aus diesem Sachverhalt kann Gefahr für Leib und Leben oder Gefahr für Sachwerte entstehen!

Fehlermeldungen

MSG_ERR_PROT_ID	Ungültige Protokollkennung
-----------------	----------------------------

Benötigte Dateien

CCMSRTCLIPlus.h
CCMSRTCLIPlus.lib
CCMSRTCLIPlus.dll

Verwandte Funktionen

MSG_PROT_NOTIFY_PROC	Callback-Funktion
----------------------	-------------------

Siehe auch

MSG_FILTER_STRUCT_PLUS (Seite 1027)

2.10.4.3 MSRTPrintMProtPlus (RT Professional)

Beschreibung

Druckt die aktuell anstehenden Meldungen des Meldefolgeprotokolls, auch wenn die Seite noch nicht voll ist.

Deklaration

```
BOOL MSRTPrintMProtPlus(  
    DWORD*      pdwLines,  
    LPCMN_ERRORW lpError );
```

Parameter

pdwLines

Zeiger auf die Zeilenzahl. Hier wird die Anzahl gedruckter Zeilen zurückgegeben. Daher sollte der Parameter vor dem Aufruf mit NULL vorbelegt werden.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Aktuell anstehenden Meldungen gedruckt.

FALSE

Fehler

Fehlermeldungen

MSG_ERR_NOPROT	Meldefolgeprotokoll nicht aktiv
----------------	---------------------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

2.10.5 Funktionen zum Bearbeiten von Meldungen (RT Professional)**2.10.5.1 MSRTEnumArchivInstancePlus (RT Professional)****Beschreibung**

Enumeriert alle Meldungen des angegebenen Archivs mit Instanznamen. Es wird die Struktur MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS zurückgeliefert.

Deklaration

```

BOOL WINAPI MSRTEnumArchivInstancePlus(
    DWORD           dwServiceID,
    BOOL            fArchiv,
    DWORD           dwMaxRecords,
    DWORD           dwParams,
    LPDWORD         lpdwTAID,
    LPCMN_ERRORW   lpError );

```

Parameter**dwServiceID**

ID des installierten Service

fArchiv

Umlauf- oder Folgearchiv

dwMaxRecords

Maximale Anzahl Records für Enumeration\n. Bei Überschreiten der maximalen Anzahl wird die Enumeration abgebrochen.

dwParams

Zusatzparameter, MSG_ENUM_ARCHIV_DESC = Sortierung absteigend

lpdwTAID

Zeiger auf die TransactionID für asynchrone Bearbeitung Rückmeldung

lpError

Zeiger auf die erweiterte Fehlerstruktur

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

2.10.5.2 MSRTCreateMsgInstanceWithCommentPlus (RT Professional)

Beschreibung

Erzeugt eine Meldung mit der angegebenen Meldungsnummer mit Instanznamen und Kommentar. Die Textbegleitwerte dürfen im Unterschied zu den meisten anderen Create-Aufrufen bis zu 256 Zeichen enthalten (255 + EndOfString). Die Meldung wird in die aktuelle Meldungsliste mit den angegebenen Daten eingetragen.

Deklaration

```
BOOL WINAPI MSRTCreateMsgInstanceWithCommentPlus (
    DWORD dwServiceID,
    LPMSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS lpMsgCreatePlus,
    LPDWORD lpdwTAID,
    LPCMN_ERRORW lpError );
```

Parameter**dwServiceID**

ID des Service - Dienstes

lpMsgCreatePlus

Zeiger auf Createstruct zum Erzeugen der Meldung

Verwenden Sie für unterschiedliche Meldungen immer unterschiedliche Zeitstempel.

lpdwTAID

Zeiger auf die TransactionID asynchroner Bearbeitung Rückmeldung

lpError

Zeiger der die erweiterte Fehlermeldung aufnimmt

Rückgabewert**TRUE**

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Die Create-Funktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe ev. über Semaphoren synchronisieren müssen.

Benutzt man den Aufruf zum Quittieren einer Meldung, so muss als Zeit die millisekundengenaue Gekommen-Zeit angegeben werden.

 WARNUNG
--

Mit dieser Funktion kann Ihre Applikation Meldungen quittieren, ohne dass sichergestellt wird, dass der Anlagenbediener diese zur Kenntnis genommen hat. Aus diesem Sachverhalt kann Gefahr für Leib und Leben oder Gefahr für Sachwerte entstehen!

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib
CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTCreateMsgInstancePlus (Seite 1066)	Instanz-Meldung erzeugen
MSRTStartMsgServicePlus (Seite 1049)	Dienst starten

Siehe auch

MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024)
MSRTStartMsgServicePlus (Seite 1049)
MSRTCreateMsgInstancePlus (Seite 1066)
MSRTSetCommentInstancePlus (Seite 1103)
MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046)

2.10.5.3 MSRTCreateMsgInstancePlus (RT Professional)

Beschreibung

Erzeugt eine Meldung mit der angegebenen Meldungsnummer. Die Meldung wird in die aktuelle Meldungsliste mit den angegebenen Daten eingetragen.

Deklaration

```
BOOL WINAPI MSRTCreateMsgInstancePlus (
    DWORD                dwServiceID,
    LPMSG_RTDATA_INSTANCE_STRUCT_PLUS lpMsgCreatePlus,
    LPDWORD              lpdwTAID,
    LPCMN_ERRORW         lpError );
```

Parameter

dwServiceID

ID des installierten Service

lpMsgCreatePlus

Zeiger auf Createstruct zum Erzeugen der Meldung

Verwenden Sie für unterschiedliche Meldungen immer unterschiedliche Zeitstempel.

lpdwTAID

Zeiger auf die TransactionID für asynchrone Bearbeitung Rückmeldung

IpError

Zeiger auf die erweiterte Fehlerstruktur

Rückgabewert**TRUE**

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Die Create-Funktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe ev. über Semaphoren synchronisieren müssen.

Benutzt man den Aufruf zum Quittieren einer Meldung, so muss als Zeit die millisekundengenaue Gekommen-Zeit angegeben werden.

 WARNUNG
--

Mit dieser Funktion kann Ihre Applikation Meldungen quittieren, ohne dass sichergestellt wird, dass der Anlagenbediener diese zur Kenntnis genommen hat. Aus diesem Sachverhalt kann Gefahr für Leib und Leben oder Gefahr für Sachwerte entstehen!

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)	Instanz-Meldung mit Kommentar erzeugen
MSRTStartMsgServicePlus (Seite 1049)	Dienst starten

Siehe auch

- MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)
- MSRTStartMsgServicePlus (Seite 1049)
- MSG_RTDATA_INSTANCE_STRUCT_PLUS (Seite 1031)
- MSRTSetCommentInstancePlus (Seite 1103)

2.10.5.4 MSRTDialogMsgLockPlus (RT Professional)

Beschreibung

Dialog zum Sperren bzw. Freigeben von Meldungen.



Folgende Möglichkeiten gibt es :

- Sperren von einzelnen Meldungen
- Sperren von Meldeklassen oder Meldegruppen
- Sperren von Meldungen über die Meldenummer.

Hinweis

Gesperrte Meldung:

Gesperrte Meldungen werden nicht in die Meldeliste eingetragen, nicht archiviert und nicht protokolliert.

Gesperrte Meldungen schicken bei Auftreten ihr Quittierungssignal zurück!

Gesperrte Meldungen sind nach einem Neustart der WinCC Runtime nicht mehr gesperrt. Weiterhin gesperrt bleiben nur Meldungen, die über Datenbausteine direkt in der AS gesperrt werden (Sperrung über Quelle).

Gesperrte Meldeklassen / Meldegruppen:

Die Sperrung von Meldeklassen und Meldegruppen bleibt auch nach einem Neustart der WinCC Runtime erhalten.

Deklaration

```

BOOL MSRTDialogMsgLockPlus(
    HWND          hwndParent,
    LPCMN_ERRORW  lpError );

```

Parameter

hwndParent

Vater-Fenster, das den Dialog besitzt.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Dialog erfolgreich geschlossen.

FALSE

Fehler oder der Dialog wurde abgebrochen.

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
-------------------	--------------------

Benötigte Dateien

CMSRTCLIPlus.h

CMSRTCLIPlus.lib

CMSRTCLIPlus.dll

2.10.5.5 MSRTCreateMsgPlus (RT Professional)

Beschreibung

Erzeugt eine Meldung mit der angegebenen Meldungsnummer. Die Meldung wird in die aktuelle Meldungsliste mit den angegebenen Daten eingetragen.

Deklaration

```
BOOL WINAPI MSRTCreateMsgPlus(  
    DWORD dwServiceID,  
    LMSG_RTCREATE_STRUCT_PLUS lpMsgCreatePlus,  
    LPDWORD lpdwTAID,  
    LPCMN_ERRORW lpError );
```

Parameter

dwServiceID

ID des installierten Service

lpMsgCreatePlus

Zeiger auf Createstruct zum Erzeugen der Meldung

Verwenden Sie für unterschiedliche Meldungen immer unterschiedliche Zeitstempel.

lpdwTAID

Zeiger auf TransactionID für asynchrone Bearbeitung

lpError

Zeiger auf die erweiterte Fehlerstruktur

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Die Create-Funktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe ev. über Semaphoren synchronisieren müssen.

WARNUNG

Mit dieser Funktion kann Ihre Applikation Meldungen quittieren, ohne dass sichergestellt wird, dass der Anlagenbediener diese zur Kenntnis genommen hat. Aus diesem Sachverhalt kann Gefahr für Leib und Leben oder Gefahr für Sachwerte entstehen!

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h
 CCMSRTCLIPlus.lib
 CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)	Instanz-Meldung erzeugen
MSRTCreateMsgInstancePlus (Seite 1066)	Instanz-Meldung mit Kommentar erzeugen
MSRTStartMsgServicePlus (Seite 1049)	Dienst starten

Siehe auch

MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)
 MSRTCreateMsgInstancePlus (Seite 1066)
 MSRTStartMsgServicePlus (Seite 1049)

2.10.5.6 MSRTEnumLockedMsgPlus (RT Professional)

Beschreibung

Enumeriert alle gesperrten Meldungen.

Enumerationsfunktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe über Semaphoren synchronisieren müssen.

Deklaration

```

BOOL WINAPI MSRTEnumLockedMsgPlus(
    DWORD          dwServiceID,
    LPDWORD        lpdwTAID,
    LPCMN_ERRORW   lpError);

```

Parameter

dwServiceID

ID des installierten Service

lpdwTAID

Zeiger auf die TransactionID für die asynchrone Bearbeitung

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR.

Rückgabewert

TRUE

Meldungen enumeriert.

FALSE

Fehler.

2.10.5.7 MSRTEnumMsgRTDataPlus (RT Professional)

Beschreibung

Enumeriert alle Meldungen, die in der aktuellen Meldeliste vorhanden sind. Ihre Callbackfunktion MSG_SERVICE_NOTIFY_PROCPPlus wird mit dem Benachrichtigungstyp MSG_NOTIFY_MSGENUM, Enumeration der Meldungen, aufgerufen. lpbyData verweist auf Daten der Struktur MSG_RTDATA_STRUCTPlus.

Enumerationsfunktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe über Semaphoren synchronisieren müssen.

Deklaration

```
BOOL MSRTEnumMsgRTData (  
    DWORD          dwServiceID,  
    LPDWORD        pdwTAID  
    LPCMN_ERRORW  lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Send- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpdwTAID

Zeiger auf die TransactionID für die asynchrone Bearbeitung

LPCMN_ERRORW

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Meldungen enumeriert.

FALSE

Fehler.

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

2.10.5.8 MSRTGetClassInfoPlus (RT Professional)**Beschreibung**

Informationen zu einer Meldeklasse holen.

Deklaration

```
BOOL WINAPI MSRTGetClassInfoPlus(  
    DWORD                dwServiceID,  
    LPCWSTR              lpszServer,  
    DWORD                dwMsgNrLow,  
    DWORD                dwMsgNrHigh,  
    LPMSG_CLASS_STRUCT_PLUS lpClassPlus,  
    LPCMN_ERRORW         lpError );
```

Parameter

dwServiceID

ID des Service - Dienstes, dessen Server verschalten werden soll

lpszServer

Server auf den verschalten werden soll

dwMsgNrLow

Meldungsnummer unteres DWORD

dwMsgNrHigh

Meldungsnummer oberes DWORD WinCC

lpClassPlus

Zeiger der die Informationen aufnimmt

lpError

Zeiger der die erweiterte Fehlermeldung aufnimmt

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Die Funktion MSRTGetClassInfoPlus kann auch ohne MSRTStartMsgServicePlus aufgerufen werden. Diese Aufrufe sind jedoch wesentlich langsamer. Sie verbessern die Performance, wenn Sie vorher ein MSRTStartMsgServicePlus durchführen und über alle Aufrufe beibehalten. In diesem Fall ist nur der erste Aufruf von MSRTGetClassInfoPlus verlangsamt.

Fehlermeldungen

MSG_ERR_RT_NOINIT	Alarm Logging RT nicht initialisiert
MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_MSG_NOEXIST	Meldung existiert nicht

Benötigte Dateien

CCMSRTCLIPlus.h
 CCMSRTCLIPlus.lib
 CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSG_CLASS_STRUCT_PLUS (Seite 1034)
 MSRTStartMsgServicePlus (Seite 1049)

2.10.5.9 MSRTGetLastMsgWithCommentPlus (RT Professional)

Beschreibung

Holt anhand von Filterbedingung und Instanzname die letzte Meldung aus einem Archiv.

Deklaration

```

BOOL WINAPI MSRTGetLastMsgWithCommentPlus (
    LPCWSTR                lpszServer,
    LPCWSTR                lpszInstance,
    LPMSG_FILTER_STRUCT_PLUS* lppFilter,
    DWORD                 dwFilterCount,
    LPMSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS lpMsg,
    LPCMN_ERRORW          lpError );

```

Parameter

lpszServer

Serverprefix ohne ':\'. Bei Angabe von Leerstring oder NULL wird der eingestellte Default-Server verwendet.

lpzInstance

Instanzname. Ein Leerstring oder auch NULL wird zur Filterbedingung mit hinzugefügt.

lppFilter

Zeiger auf den Filterpointer ARRAY zur Auswahl der Art der Meldung.

dwFilterCount

Anzahl der gefilterten Meldungen.

lpMsg

Gibt die Meldungsdaten zurück.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR.

Rückgabewert

TRUE

Meldung abgefragt.

FALSE

Fehler.

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_RT_NOTCONNECTED	Verbindung nicht aufgebaut

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Siehe auch

MSG_FILTER_STRUCT_PLUS (Seite 1027)

MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024)

MSG_TEXTVAL256_STRUCT_PLUS (Seite 1046)

2.10.5.10 MSRTGetMsgActualPlus (RT Professional)

Beschreibung

Die Funktion ermittelt die Anzahl der in der Meldeliste aktuell anstehenden Meldungen.

Deklaration

```
BOOL MSRTGetMsgActualPlus (  
    LPDWORD          lpdwCount,  
    LPCMN_ERRORW    lpError );
```

Parameter

lpdwCount

Zeiger auf einen Puffer in dem die Anzahl der aktuell anstehenden Meldungen hinterlegt wird.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Anzahl ermittelt.

FALSE

Fehler

Bemerkung

Die Funktion MSRTGetMsgActualPlus kann auch ohne MSRTStartMsgServicePlus aufgerufen werden. Diese Aufrufe sind jedoch wesentlich langsamer. Sie verbessern die Performance, wenn Sie vorher ein MSRTStartMsgServicePlus durchführen und über alle Aufrufe beibehalten. In diesem Fall ist nur der erste Aufruf von MSRTGetMsgActualPlus verlangsamt.

Fehlermeldungen

MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert
-------------------	---------------------------------

Benötigte Dateien

CMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

2.10.5.11 MSRTGetMsgCSDDataPlus (RT Professional)

Beschreibung

Projektierungsdaten einer Meldung abfragen.

Deklaration

```
BOOL MSRTGetMsgCSDDataPlus (
    DWORD           dwServiceNr,
    LPCWSTR         lpszServer,
    DWORD           dwMsgNrLow,
    DWORD           dwMsgNrHigh,
    LPMMSG_CSDATA_STRUCT_PLUS lpmCSDDataPlus,
    LPCMN_ERRORW    lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgService zurückgeliefert wird.

lpszServer

Name des Servers.

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

IpmCSDDataPlus

Zeiger auf die Projektierungsdaten (Meldeklasse, Meldeart, Textblockindizes,...) dieser Meldung in der Struktur MSG_CSDATA_STRUCT_PLUS (Seite 1037).

IpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Projektierungsdaten abgefragt.

FALSE

Fehler

Bemerkung

Die Funktion MSRTGetMsgCSDDataPlus kann auch ohne MSRTStartMsgServicePlus aufgerufen werden. Diese Aufrufe sind jedoch wesentlich langsamer. Sie verbessern die Performance, wenn Sie vorher ein MSRTStartMsgServicePlus durchführen und über alle Aufrufe beibehalten. In diesem Fall ist nur der erste Aufruf von MSRTGetMsgCSDDataPlus verlangsamt.

Fehlermeldungen

MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert
MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_MSG_NOEXIST	Meldung existiert nicht

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSG_CSDATA_STRUCT_PLUS (Seite 1037)

MSRTStartMsgServicePlus (Seite 1049)

2.10.5.12 MSRTGetMsgPriorityPlus (RT Professional)

Beschreibung

Priorität der angegebenen Meldung abfragen.

Deklaration

```
BOOL MSRTGetMsgPriorityPlus (  
    DWORD                dwServiceID,  
    LPCWSTR              lpszServer,  
    DWORD                dwMsgNrLow,  
    DWORD                dwMsgNrHigh,  
    long*                plPriority,  
    LPCMN_ERRORW        lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgService zurückgeliefert wird.

lpszServer

Name des Servers.

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgHigh Nummer der letzten Meldung.

plPriority

Zeiger auf eine Variable vom Datentyp long, in der die Priorität abgelegt wird.

Dieser Parameter darf nicht NULL sein!

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Priorität abgefragt.

FALSE

Fehler

Bemerkung

Die Funktion MSRTGetMsgPriorityPlus kann auch ohne MSRTStartMsgServicePlus aufgerufen werden. Diese Aufrufe sind jedoch wesentlich langsamer. Sie verbessern die Performance, wenn Sie vorher ein MSRTStartMsgServicePlus durchführen und über alle Aufrufe beibehalten. In diesem Fall ist nur der erste Aufruf von MSRTGetMsgPriorityPlus verlangsamt.

Fehlermeldungen

MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert
MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_MSG_NOEXIST	Meldung existiert nicht

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

2.10.5.13 MSRTGetMsgQuitPlus (RT Professional)**Beschreibung**

Anzahl quittierpflichtiger Meldungen abfragen.

Deklaration

```

BOOL MSRTGetMsgQuitPlus (
    LPDWORD      lpdwCount,
    LPCMN_ERRORW lpError );

```

Parameter

lpdwCount

Zeiger für die Anzahl quittierpflichtiger Meldungen

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Anzahl ermittelt.

FALSE

Fehler

Bemerkung

Die Funktion MSRTGetMsgPriorityPlus kann auch ohne MSRTStartMsgServicePlus aufgerufen werden. Diese Aufrufe sind jedoch wesentlich langsamer. Sie verbessern die Performance, wenn Sie vorher ein MSRTStartMsgServicePlus durchführen und über alle Aufrufe beibehalten. In diesem Fall ist nur der erste Aufruf von MSRTGetMsgPriorityPlus verlangsamt.

Fehlermeldungen

MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert
-------------------	---------------------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

2.10.5.14 MSRTGetMsgTextPlus (RT Professional)**Beschreibung**

Text zu einem Textblock holen

Deklaration

```

BOOL WINAPI MSRTGetMsgTextPlus (
    WORD                wTextBlock,
    DWORD               dwTextNr,
    LPWSTR              lpszMsgText,
    DWORD*              lpdwCount,
    LPCMN_ERRORW        lpError );

```

Parameter**wTextBlock**

Textblocknummer

dwTextNr

Textblock-ID

lpszMsgText

Zeiger auf den Textblockstring

lpdwCount

Zeiger auf die Länge des Textblockstrings. Die Länge wird als Anzahl der Zeichen angegeben.

lpdwCount=0	Textgröße wird abgefragt.
lpdwCount>0 UND Puffergröße von lpszMsgText > lpdwCount	Text wird begrenzt und aktuelle Anzahl der Zeichen zurückgegeben.
lpdwCount>0 UND Puffergröße von lpszMsgText > lpdwCount UND Puffer initialisiert	Text wird begrenzt und zurückgegeben.

lpError

Zeiger der die erweiterte Fehlermeldung aufnimmt

Rückgabewert**TRUE**

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Die Funktion `MSRTGetMsgPriorityPlus` kann auch ohne `MSRTStartMsgServicePlus` aufgerufen werden. Diese Aufrufe sind jedoch wesentlich langsamer. Sie verbessern die Performance, wenn Sie vorher ein `MSRTStartMsgServicePlus` durchführen und über alle Aufrufe beibehalten. In diesem Fall ist nur der erste Aufruf von `MSRTGetMsgPriorityPlus` verlangsamt.

Fehlermeldungen

MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert
-------------------	---------------------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

2.10.5.15 MSRTGetSelectedMsgPlus (RT Professional)

Beschreibung

Die Funktion ermittelt zu einer in einem Meldefenster oder einer Meldeanzeige selektierten Meldung die Meldungsnummer. Die Spezifikation der Anzeige erfolgt über `IpszTemplate`.

In Einzel- und Mehrplatzprojekten funktionieren Abfragen auf das Template ohne Einschränkungen. Abfragen in der Anzeige funktionieren nicht für Engineering Daten und nur auf in der Anzeige vorhandenen Prozesswerte.

Deklaration

```
BOOL MSRTGetSelectedMsgPlus (
    LPCWSTR                                lpszTemplate,
    LPMSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS lpMsgRTPlus,
    LPCMN_ERROR                             lpError );
```

Parameter

lpszTemplate

Zeiger auf den Namen der Anzeige in dem eine Meldung selektiert ist.

lpMsgRTPlus

Zeiger auf eine Struktur vom Typ MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024) , in der die Meldungsnummer zurückgeliefert wird.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Daten abgefragt.

FALSE

Fehler

Bemerkung

Die Funktion MSRTGetMsgPriorityPlus kann auch ohne MSRTStartMsgServicePlus aufgerufen werden. Diese Aufrufe sind jedoch wesentlich langsamer. Sie verbessern die Performance, wenn Sie vorher ein MSRTStartMsgServicePlus durchführen und über alle Aufrufe beibehalten. In diesem Fall ist nur der erste Aufruf von MSRTGetMsgPriorityPlus verlangsamt.

Hinweis

Je nachdem, welche Sicht in der Anzeige eingestellt ist, werden die Flags in dwFlags der Struktur MSG_RTDATA_INSTANCECOMMENT_STRUCTPLUS anders zugeordnet:

- für die Online-Sicht: MSG_RT_FLAG_XXX-Flags,
 - für die Archiv-Sicht: MSG_FLAG_-Flags, und
 - für die Hitliste: keine Zuordnung möglich, Flags werden ignoriert
-

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
-------------------	--------------------

Benötigte Dateien

CCMSRTCLIPlus.h
CCMSRTCLIPlus.lib
CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Beispiele

Get selected Msg "MS02.cpp"

Siehe auch

MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024)
MSRTStartMsgServicePlus (Seite 1049)

2.10.5.16 MSRTResetMsgPlus (RT Professional)

Beschreibung

Die angegebene Meldung wird quittiert.

Deklaration

```
BOOL MSRTResetMsgPlus (  
    DWORD          dwServiceID,  
    DWORD          dwMsgNrLow,  
    DWORD          dwMsgNrHigh,  
    LPDWORD       lpdwTAID  
    LPCMN_ERRORW  lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Send- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgService zurückgeliefert wird.

dwMsgNrLow

dwMsgNrLow Nummer der ersten Meldung

dwMsgNrHigh

dwMsgNrHigh Nummer der letzten Meldung.

Sie müssen dwMsgNrLow und dwMsgNrHigh belegen, wenn Sie die Filterkriterien MSG_FILTER_NR, MSG_FILTER_NR_FROM oder MSG_FILTER_NR_TO verwenden.

lpdwTAID

Zeiger auf die TransactionID für asynchrone Bearbeitung

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Meldung quittiert.

FALSE

Fehler

Bemerkung

 WARNUNG
Mit dieser Funktion kann Ihre Applikation Meldungen quittieren, ohne dass sichergestellt wird, dass der Anlagenbediener diese zur Kenntnis genommen hat. Aus diesem Sachverhalt kann Gefahr für Leib und Leben oder Gefahr für Sachwerte entstehen!

Fehlermeldungen

MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert
MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_MSG_NOEXIST	Meldung existiert nicht
MSG_ERR_MSG_NOTFOUND	Meldung nicht gefunden
MSG_ERR_MSG_NOQUIT	Meldung hat keine Quittierpflicht
MSG_ERR_MSG_ALREADYQUIT	Meldung schon quittiert

Benötigte Dateien

CMSRTCLIPlus.h

CMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

2.10.5.17 MSRTLoopInAlarmPlus (RT Professional)

Beschreibung

API - Funktion für Loop-In-Alarm. Massgebend ist die Meldungsnummer.

Deklaration

```
BOOL MSRTLoopInAlarmPlus (  
    DWORD          dwServiceID,  
    LPCWSTR        lpszServer,  
    DWORD          dwMsgNrLow,  
    DWORD          dwMsgNrHigh,  
    LPCMN_ERRORW   lpError );
```

Parameter

dwServiceID

ID des Service - Dienstes, dessen Server verschalten werden soll

lpszServer

Server auf den verschalten werden soll

dwMsgNrLow

Meldungsnummer unteres DWORD

dwMsgNrHigh

Meldungsnummer oberes DWORD WinCC

lpError

Zeiger der die erweiterte Fehlermeldung aufnimmt

Rückgabewert**TRUE**

Funktion erfolgreich

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_MSG_NOEXIST	Meldung existiert nicht

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

2.10.6 Funktionen zum Bearbeiten von Meldegruppen (RT Professional)**2.10.6.1 MSRTEnumGroupMsgPlus (RT Professional)****Beschreibung**

Alle Einzelmeldungen der Meldungsgruppe enumerieren. Maximal werden MSG_MAX_GROUPITEMS Einzelmeldungen enumeriert. Ihre Callbackfunktion MSG_SERVICE_NOTIFY_PROCPlus wird mit dem Benachrichtigungstyp MSG_NOTIFY_GROUPENUM aufgerufen. lpybyData verweist auf Daten der Struktur MSG_RTGROUPENUM_STRUCTPlus.

Die Enumerationsfunktionen des Meldesystems arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe über Semaphoren synchronisieren müssen.

Deklaration

```
BOOL WINAPI MSRTEnumGroupMsgPlus (
    DWORD                dwServiceID,
    LPMSG_RTGROUPSET_STRUCT_PLUS  lpmGroupPlus,
    LPDWORD              lpdwTAID,
    LPCMN_ERRORW        lpError );
```

Parameter

dwServiceID

ID des Service - Dienstes

lpmGroupPlus

Zeiger auf Meldungsgruppen-Info

lpdwTAID

Zeiger auf die TransactionID für asynchrone Bearbeitung Rückmeldung

lpError

Zeiger der die erweiterte Fehlermeldung aufnimmt

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Callback-Funktion
MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
MSG_RTGROUPENUM_STRUCT_PLUS (Seite 1039)	Einzelmeldungen einer Meldungsgruppe

Siehe auch

MSG_RTGROUPSET_STRUCT_PLUS (Seite 1040)
 MSRTStartMsgServicePlus (Seite 1049)
 MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)
 MSG_RTGROUPENUM_STRUCT_PLUS (Seite 1039)

2.10.6.2 MSRTLockGroupPlus (RT Professional)**Beschreibung**

Sperren einer projektierten Meldungsgruppe. Alle Einzelmeldungen der Meldungsgruppe sowie die Einzelmeldungen der unterlagerten Gruppen werden ebenfalls mitgesperrt.

Deklaration

```

BOOL MSRTLockGroupPlus (
    DWORD                dwServiceID,
    LPMSG_RTGROUPSET_STRUCTPlus  lpmGroup,
    LPDWORD              pdwTAID
    LPCMN_ERRORW        lpError );

```

Parameter**dwServiceID**

Identifikationsnummer des Send- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpmGroup

Zeiger auf die Struktur MSG_RTGROUPSET_STRUCT_PLUS (Seite 1040) mit den Informationen der Meldungsgruppen.

Sperren: dwData != 0; Freigeben: dwData = 0

pdwTAID**lpError**

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Meldungen der Meldegruppe gesperrt.

FALSE

Fehler

Bemerkung

Es werden nur projektierte Meldungsgruppen gesperrt oder freigegeben. In Runtime definierte Gruppen werden nicht erfasst und die Funktion hat keine Wirkung.

Fehlermeldungen

MSG_ERR_RT_NOINIT	Meldesystem nicht initialisiert
MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSG_RTGROUPSET_STRUCT_PLUS (Seite 1040)

MSRTStartMsgServicePlus (Seite 1049)

2.10.6.3 MSRTQuitGroupPlus (RT Professional)

Beschreibung

Quittieren einer Meldegruppe. Alle Einzelmeldungen der Meldegruppe sowie die Meldungen von unterlagerten Gruppen werden quittiert.

Deklaration

```
BOOL MSRTQuitGroupPlus (
    DWORD                dwServiceID,
    LPMSG_RTGROUPSET_STRUCTPlus  lpmGroupPlus,
    LPDWORD              pdwTAID
    LPCMN_ERRORW        lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Sendee- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpmGroup

Zeiger auf eine Struktur vom Typ MSG_RTGROUPSET_STRUCT_PLUS (Seite 1040) mit den Informationen der Meldegruppen.

pdwTAID

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Meldungen der Meldegruppe quittiert

FALSE

Fehler

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Siehe auch

MSG_RTGROUPSET_STRUCT_PLUS (Seite 1040)

2.10.7 Funktionen zum Bearbeiten von Meldefiltern (RT Professional)

2.10.7.1 MSRTGetFilterDataPlus (RT Professional)

Beschreibung

Liest die Selektionskriterien der Meldeanzeige aus und legt das Ergebnis in die Struktur MSG_FILTER_STRUCT_PLUS ab.

Deklaration

```
BOOL WINAPI MSRTGetFilterDataPlus(  
    DWORD          dwServiceID,  
    LPCWSTR        lpszName,  
    LPMMSG_FILTER_STRUCT_PLUS** lpppMsgFilter,  
    LPDWORD        lpdwFilterCount,  
    LPCMN_ERRORW  lpError );
```

Parameter

dwServiceID

ID des Service - Dienstes

lpszName

Name der Meldeanzeige. Den Namen legen Sie in den Eigenschaften der Meldeanzeige fest.

lpMsgFilterPlus

Zeiger auf Filterpointer array auf Filterstrukturen

lpdwFilterCount

lpError

Zeiger, der die erweiterte Fehlermeldung aufnimmt

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Nur projektierte Meldefilter werden ermittelt. Meldefilter die mit MSRTStartMsgServicePlus oder mit MSRTSetMsgFilterPlus gesetzt wurden, werden nicht erfasst.

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig
MSG_ERR_API_NODATA	Keine Daten

Benötigte Dateien

CCMSRTCLIPlus.h
 CCMSRTCLIPlus.lib
 CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTSetMsgFilterPlus (Seite 1097)	Meldefilter setzen
MSRTStartMsgServicePlus (Seite 1049)	Dienst starten

Siehe auch

MSG_FILTER_STRUCT_PLUS (Seite 1027)
 MSRTStartMsgServicePlus (Seite 1049)
 MSRTSetMsgFilterPlus (Seite 1097)

2.10.7.2 MSRTCheckWinFilterPlus (RT Professional)**Beschreibung**

Überprüft, ob die angegebene Meldung den Filterbedingung entspricht.

Deklaration

```

BOOL MSRTCheckWinFilterPlus (
    DWORD                dwServiceID,
    LPCWSTR              lpszServer,
    LPMSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS lpMsgRTPlus,
    LPMSG_FILTER_STRUCT_PLUS* lppMsgFilter,
    DWORD                dwFilterCount,
    LPWORD               lpwReturn,
    LPCMN_ERRORW        lpError );
  
```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpszServer

Zeiger auf den Servernamen. Der Name wird ohne Servertrennzeichen :: angegeben. Bei Angabe von Leerstring oder NULL wird der eingestellte Default-Server verwendet

lpMsgRTPlus

Zeiger auf eine Struktur vom Typ MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024) mit den Runtime-Daten einer Meldung.

lpMsgFilter

Zeiger auf eine Struktur vom Typ MSG_FILTER_STRUCT_PLUS (Seite 1027) mit den Daten des Meldefilters.

lpwReturn

Zeiger auf das Funktionsergebnis.

TRUE	Meldung entspricht den Filterbedingungen.
FALSE	Meldung entspricht nicht den Filterbedingungen.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Überprüfung wurde durchgeführt.

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
-------------------	--------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSG_FILTER_STRUCT_PLUS (Seite 1027)

MSRTStartMsgServicePlus (Seite 1049)

MSG_RTDATA_INSTANCECOMMENT_STRUCT_PLUS (Seite 1024)

2.10.7.3 MSRTSetMsgFilterPlus (RT Professional)

Beschreibung

Setzt den Filter für den angegebenen Service neu.

Deklaration

```

BOOL WINAPI MSRTSetMsgFilterPlus(
    DWORD                dwServiceID,
    LPMMSG_FILTER_STRUCT_PLUS* lppMsgFilterPlus,
    DWORD                dwFilterCount,
    LPCMN_ERRORW         lpError );

```

Parameter

dwServiceID

Identifikationsnummer des Send- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lppMsgFilterPlus

Zeiger auf Pointer ARRAY für Filterbedingungen

dwFilterCount

Anzahl Filter im ARRAY

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERRORW. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
MSRTGetFilterDataPlus (Seite 1094)	Selektionskriterien der Meldenanzeige auslesen

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

MSG_FILTER_STRUCT_PLUS (Seite 1027)

MSRTGetFilterDataPlus (Seite 1094)

2.10.7.4 MSRTSetMsgWinFilterPlus (RT Professional)

Beschreibung

Legt neue Filterbedingungen für die Meldeanzeige fest. Alle derzeit aktuellen Meldeanzeigen mit diesem Namen werden aktualisiert.

Deklaration

```

BOOL MSRTSetMsgWinFilterPlus (
    LPMSG_FILTER_STRUCT_PLUS    lppMsgFilterPlus,
    DWORD                       dwFilterCount,
    LPCMN_ERRORW                lpError );

```

Parameter

lppMsgFilterPlus

Zeiger auf eine Struktur vom Typ MSG_FILTER_STRUCT_PLUS (Seite 1027) mit den Daten des Meldefilters. Der Namen der Filterstruktur ist gleich dem Fensternamen.

dwFilterCount

Anzahl der Filter

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR . Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Filter wurde gesetzt

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Siehe auch

MSG_FILTER_STRUCT_PLUS (Seite 1027)

2.10.8 Funktionen zum Bearbeiten von Meldeanzeige (RT Professional)

2.10.8.1 MSRTMsgWinCommandPlus (RT Professional)

Beschreibung

Führt in einer Meldeanzeige eine Bedienfunktion aus. Wenn Sie diese Funktion nutzen, verwenden Sie innerhalb eines Bildes eine Meldefenster Vorlage nur einmal. Die Kommandos werden auch an Controls mit dem Caption-Namen gleich dem Templatennamen weitergereicht. (Durch die Umsetzung von Bildern mit Meldetemplates in Bilder mit Controls entstehen Controls mit dem Caption Namen gleich dem Template-Namen).

Deklaration

```

BOOL MSRTMsgWinCommand(
    LPTSTR      lpszTemplate,
    DWORD      dwCommandID,
    LPCMN_ERROR lpError );
    
```

Parameter

lpszTemplate

Zeiger auf den Fensternamen

dwCommandID

Die Festlegung der ID der Bedienfunktion erfolgt über die Konstanten in der Datei "m_global.h":

MSG_TB_MSGWIN	Prozessmeldefenster abrufen
MSG_TB_ARC_S	Umlaufarchiv abrufen
MSG_TB_ARC_L	Folgearchiv abrufen
MSG_TB_QH	Hupenquittierung auslösen
MSG_TB_QM	Einzelquittierung auslösen
MSG_TB_QS	Sammelquittierung auslösen
MSG_TB_SCROLL	Scrollfunktion ein-, ausschalten
MSG_TB_SELECT	Selektion - Dialog aufblenden
MSG_TB_LOCK	Sperrdialog aufblenden
MSG_TB_PROTOCOL	Protokolldialog aufblenden
MSG_TB_RESET	Notquittierung auslösen
MSG_TB_MSGFIRST	An den Anfang der Liste positionieren (Autoscroll darf nicht aktiviert sein)
MSG_TB_MSGLAST	Ans Ende der Liste positionieren (Autoscroll darf nicht aktiviert sein)
MSG_TB_MSGNEXT	Zur nächste Meldung positionieren (Autoscroll darf nicht aktiviert sein)

MSG_TB_MSGPREV	Zur vorherigen Meldung positionieren (Autoscroll darf nicht aktiviert sein)
MSG_TB_INFO	Infotextdialog zur aktuell selektierten Meldung aufblenden
MSG_TB_COMMENT	Kommentar eingeben zur aktuell selektierten Meldung
MSG_TB_LOOPINALARM	Loop In Alarm zur aktuell selektierten Meldung auslösen
MSG_TB_PRINT	Druck starten

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Bedienfunktion ausgeführt

FALSE

Fehler

Bemerkung

Alle geöffneten Meldeanzeigen mit dem angegebenen Namen führen die Bedienfunktion aus.

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
-------------------	--------------------

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

2.10.9 Funktionen zum Bearbeiten von Kommentaren (RT Professional)

2.10.9.1 MSRTGetCommentInstancePlus (RT Professional)

Beschreibung

Ermittelt den Kommentartext zu einer archivierten Meldung.

Deklaration

```
BOOL MSRTGetCommentInstancePlus (
    DWORD dwServiceID,
    LPMSG_COMMENT_INSTANCE_STRUCT_PLUS lpmCommentPlus,
    LPCMN_ERRORW lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpmCommentPlus

Zeiger auf eine Struktur vom Typ MSG_COMMENT_INSTANCE_STRUCT_PLUS (Seite 1041) mit den Daten eines Kommentars. Folgende Werte in der Struktur sind notwendig: Meldenummer, Datum und Uhrzeit und Instanzname.

Hinweis

Datum und Uhrzeit müssen auf die Millisekunde genau dem Zeitstempel der Meldung entsprechen, zu welcher der Kommentartext ermittelt werden soll.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR W. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Komentartext ermittelt.

Ist kein Kommentar vorhanden, so kommt der Aufruf mit TRUE zurück, aber das Element szText ist leer.

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

MSG_COMMENT_INSTANCE_STRUCT_PLUS (Seite 1041)

2.10.9.2 MSRTSetCommentInstancePlus (RT Professional)**Beschreibung**

Legt den Kommentartext zu einer archivierten Meldung fest.

Deklaration

```

Bool MSRTSetCommentInstancePlus (
    DWORD                dwServiceID,
    LPMSG_COMMENT_INSTANCE_STRUCT_PLUS  lpmCommentPlus,
    LPDWORD              pdwTAID,
    LPCMN_ERRORW        lpError );

```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpmCommentPlus

Zeiger auf eine Struktur vom Typ MSG_COMMENT_INSTANCE_STRUCT_PLUS (Seite 1041) mit den Daten eines Kommentars. Folgende Werte in der Struktur sind notwendig: Meldenummer, Instanzname, Datum, Uhrzeit, Text und Benutzername.

pdwTAID

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Erfolgreich an Server abgeschickt.

FALSE

Fehler

Bemerkung

Die Daten der Funktion MSRTSetCommentPlus werden asynchron geschrieben. Dadurch werden nicht alle Fehler an den Anwender zurückgegeben. Überprüfen Sie mit MSRTGetCommentInstancePlus die Schreibweise.

Wenn MSRTSetCommentInstancePlus nach MSRTCreateMsgPlus oder MSRTCreateMsgInstancePlus ausgeführt wird, kann es aufgrund der Asynchronität und je nach Anlagenauslastung zu Überholvorgängen kommen. Das heißt der Kommentar wird festgelegt, bevor die Meldung erzeugt wird.

Verwenden Sie statt dieser Aufrufkombination die Funktion MSRTCreateMsgInstanceWithCommentPlus.

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h
 CCMSRTCLIPlus.lib
 CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)	Meldung erzeugen
MSRTCreateMsgInstancePlus (Seite 1066)	Meldung erzeugen

Siehe auch

MSG_COMMENT_INSTANCE_STRUCT_PLUS (Seite 1041)
 MSRTStartMsgServicePlus (Seite 1049)
 MSRTCreateMsgInstanceWithCommentPlus (Seite 1064)
 MSRTCreateMsgInstancePlus (Seite 1066)

2.10.10 Funktionen zum Bearbeiten von Infotexten (RT Professional)**2.10.10.1 MSRTGetInfotextPlus (RT Professional)****Beschreibung**

Holt den Infotext zu einer Meldung. Folgende Werte müssen hierzu übergeben werden :
 Meldungsnummer

Deklaration

```

BOOL WINAPI MSRTGetInfotextPlus (
    DWORD                dwServiceID,
    LPMSG_INFOTEXT_STRUCT_PLUS  lpmInfotext,
    LPCMN_ERRORW        lpError );

```

Parameter

dwServiceID
 ID des Service - Dienstes

lpmInfotext

Zeiger der Infotextstruktur

lpError

Zeiger der die erweiterte Fehlermeldung aufnimmt

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSG_INFOTEXT_STRUCT_PLUS (Seite 1043)

MSRTStartMsgServicePlus (Seite 1049)

2.10.10.2 MSRTSetInfotextPlus (RT Professional)

Beschreibung

Setzt den Infotext zu einer Meldung.

Deklaration

```

BOOL WINAPI MSRTSetInfotextPlus (
    DWORD                dwServiceID,
    LPMSG_INFOTEXT_STRUCT_PLUS  lpmInfotextPlus,
    DWORD*               lpdwTAID,
    LPCMN_ERRORW         lpError );

```

Parameter

dwServiceID

ID des Service-Dienstes

lpmInfotextPlus

Zeiger der Infotextstruktur

lpdwTAID

Zeiger auf die TransactionID mit der Rückmeldung für die asynchrone Bearbeitung.

lpError

Zeiger der die erweiterte Fehlermeldung aufnimmt.

Rückgabewert

TRUE

Funktion erfolgreich

FALSE

Fehler

Bemerkung

Der neue Infotext ist ab dem übernächsten Minutenwechsel wirksam. Bis dahin wird der vorhergehende Infotext verwendet.

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSG_INFOTEXT_STRUCT_PLUS (Seite 1043)

MSRTStartMsgServicePlus (Seite 1049)

2.10.11 Archivierungsfunktionen (RT Professional)

2.10.11.1 MSRTEnumBackupListPlus (RT Professional)

Beschreibung

Listet die Einträge der Auslagerungsdatei für das Folgearchiv auf.

Deklaration

```
BOOL MSRTEnumBackupListPlus (  
    DWORD          dwServiceID,  
    LPCMN_ERROR    lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Send- und Empfangsdienstes, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Auslagerungsliste erstellt.

FALSE

Fehler

Bemerkung

Die Enum-Funktionen arbeiten asynchron. Dies hat zur Folge, dass Sie die Aufrufe ev. über Semaphoren synchronisieren müssen

Die Callback-Funktion MSG_SERVICE_NOTIFY_PROCPlus wird mit dem Parameter MSG_NOTIFY_BACKUPENUM aufgerufen. Der Parameter lpbyData zeigt auf die Daten der Struktur MSG_BACKUP_STRUCT_PLUS (Seite 1044).

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

CCMSRTCLIPlus.h

CCMSRTCLIPlus.lib

CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)	Callback-Funktion

Siehe auch

MSG_BACKUP_STRUCT_PLUS (Seite 1044)

MSRTStartMsgServicePlus (Seite 1049)

MSG_SERVICE_NOTIFY_PROCPlus (Seite 1055)

2.10.11.2 MSRTExportPlus (RT Professional)**Beschreibung**

Exportiert Meldungen aus dem Umlaufarchiv und den verbundenen Archiv-Backupdateien.

Deklaration

```
BOOL MSRTExportPlus (
    DWORD                dwServiceID,
    LMSG_BACKUP_STRUCT_PLUS lpMsgBackup,
    LPCMN_ERRORW        lpError );
```

Parameter

dwServiceID

Identifikationsnummer des Sende- und Empfangsdiensts, wie sie beim Aufruf von MSRTStartMsgServicePlus zurückgeliefert wird.

lpMsgBackup

Zeiger auf eine Struktur vom Typ MSG_BACKUP_STRUCT_PLUS (Seite 1044).

lpError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Meldungen exportiert.

FALSE

Fehler

Fehlermeldungen

MSG_ERR_API_PARAM	Parameter ungültig
MSG_ERR_API_SERVICE	dwServiceID ungültig

Benötigte Dateien

- CCMSRTCLIPlus.h
- CCMSRTCLIPlus.lib
- CCMSRTCLIPlus.dll

Verwandte Funktionen

MSRTStartMsgServicePlus (Seite 1049)	Dienst starten
--------------------------------------	----------------

Siehe auch

MSRTStartMsgServicePlus (Seite 1049)

MSG_BACKUP_STRUCT_PLUS (Seite 1044)

2.11 Funktionen zur Anzeige von PLC-Code (RT Professional)**2.11.1 Anzeige in STEP 7 (RT Professional)****2.11.1.1 Grundlagen (RT Professional)****Allgemeines**

Mit diesen Funktionen können Sie von einem Bild in WinCC Runtime direkt zu der Verwendungsstelle einer Prozessvariablen im Programmcode von STEP 7 wechseln. Sie haben damit die Möglichkeit zur schnellen und einfachen Diagnose von Störungen. Um die Einsprung-Funktionen zu beschleunigen besteht die Möglichkeit, das TIA Portal zu starten und ein Projekt zu öffnen. Dies erfolgt typischerweise beim Anlauf von WinCC Runtime.

Beim Einsprung wird zunächst geprüft, ob das TIA Portal bereits geöffnet ist. Ist dies nicht der Fall, wird das TIA Portal automatisch gestartet. Im TIA Portal wird der entsprechende Editor geöffnet und die Verwendungsstelle, die Zuweisung, der Call oder der Schritt gesucht.

Header-Dateien

Die Deklaration der hier erläuterten Funktionen und Strukturen erfolgt in den Header-Dateien:

kopapi.h	API Schnittstellendefinition Header-Datei
----------	---

Librarys

Die DLL-Anbindung der hier erläuterten Funktionen erfolgt in den Lib-Dateien kopapi.dll

kopapi.lib	Library
kopapi.dll	

Siehe auch

OpenTIAPortalProject (Seite 1112)

OpenTIAPortalIECPLByCall (Seite 1114)

OpenTIAPortalIECPLByAssignment (Seite 1116)

OpenTIAPortalS7GraphByBlock (Seite 1118)

Beispiel: Einbinden in eine WinCC Funktion (Seite 1121)

2.11.1.2 OpenTIAPortalProject (RT Professional)

Beschreibung

Mit dieser Funktion können, sie das TIA Portal starten und ein Projekt öffnen. Dadurch beschleunigen sich die einzelnen Einsprungfunktionen. Sie sollten jedoch Bedenken, dass ein im Hintergrund geöffnetes Programm Speicher und Rechenzeit benötigt.

Deklaration

```
BOOL OpenTIAPortalProject (  
    DWORD        dwFlags,  
    LPCTSTR      lpszTiaPortalProjectPath,  
    LPCTSTR      lpszErrorTag,  
    LPCMN_ERROR  lpdmError);
```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

- IECPLVIEWER_PIN_SUBSTRING_SEARCH=0x0001: Bei der Suche nach dem Pin-Namen wird nach einem Teilstring gesucht, d.h. der Pin-Name beginnt mit der in lpszPin übergebenen Zeichenkette. Ist dieses Bit nicht gesetzt, wird der vollständige Pin-Name mit lpszPin verglichen wird.
- KOPAPI_FLAG_TIAPORTAL_SUPPRESS_PROGRAM_STATUS=0x0004: Das TIA Portal geht nach dem Öffnen des Bausteins nicht in den Online Modus. Ist dieses Bit nicht gesetzt, wird nach dem Öffnen des Bausteins der Online-Modus gestartet.

- KOPAPI_FLAG_TIAPORTAL_CHECK_PROJECT_STATE 0x0010L: Wenn dieses Bit gesetzt ist, wird der Status des Projekts geprüft.. Das Projekt wird nicht geöffnet.
 - Das Projekt ist geöffnet.
Rückgabewert FALSE
error.dwError1 = KOPAPI_E_TIAPORTAL_PROJECT_NOT_OPEN
 - Das Projekt ist geöffnet und enthält keine Änderungen.
Rückgabewert TRUE
error.dwError1 = 0
 - Das Projekt ist geöffnet und enthält Änderungen.
Rückgabewert FALSE
error.dwError1 = KOPAPI_E_TIAPORTAL_PROJECT_MODIFIED
 - Das Projekt ist schreibgeschützt geöffnet.
Rückgabewert FALSE
error.dwError1 = KOPAPI_E_TIAPORTAL_PROJECT_READ_ONLY
- KOPAPI_FLAG_TIAPORTAL_DONT_USE_MODIFIED_PROJECT 0x0008L: Wenn dieses Bit gesetzt ist, wird der Aufruf abgebrochen, wenn das Projekt bereits geöffnet ist und Änderungen enthält.

IpszTiaPortalProjectPath

Name der Projektdatei einschließlich der Angabe des absoluten Pfades, z. B. "D:\TIAProjects\Project1\Project1.ap12"

Bitte beachten Sie, dass innerhalb eines C-Skriptes der Backslash über eine Escape-Sequenz geschrieben werden muss:

```
FunctionX(..., "D:\\TIAProjects\\Project1\\Project1.ap12", ...);
```

IpszErrorTag

Name einer internen WinCC Variablen vom Datentyp String. Beim Aufruf von asynchronen Funktionen, die nicht sofort ein Ergebnis liefern, werden in IpszErrorTag (Seite 1134) die Fehlerinformationen zurückgeliefert.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

kopapi.h

kopapi.lib

kopapi.dll

Siehe auch

Grundlagen (Seite 1111)

2.11.1.3 OpenTIAPortalIECPLByCall (RT Professional)

Beschreibung

Die Funktion wird für die Sprachen KOP und FUP verwendet und zeigt die Vorverschaltung eines Netzwerkeingangs eines Standardbausteins.

Deklaration

```
BOOL OpenTIAPortalIECPLByCall (  
    DWORD        dwFlags,  
    LPCTSTR      lpszTiaPortalProjectPath,  
    LPCTSTR      lpszCpuName,  
    LPCTSTR      lpszContainingBlock,  
    LPCTSTR      lpszCalledBlock,  
    LPCTSTR      lpszPin,  
    LPCTSTR      lpszErrorTag,  
    LPCMN_ERROR  lpdmError);
```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

- IECPLVIEWER_PIN_SUBSTRING_SEARCH=0x0001: Bei der Suche nach dem Pin-Namen wird nach einem Teilstring gesucht, d.h. der Pin-Name beginnt mit der in lpszPin übergebenen Zeichenkette. Ist dieses Bit nicht gesetzt, wird der vollständige Pin-Name mit lpszPin verglichen wird.
- KOPAPI_FLAG_TIAPORTAL_SUPPRESS_PROGRAM_STATUS=0x0004: Das TIA Portal geht nach dem Öffnen des Bausteins nicht in den Online Modus. Ist dieses Bit nicht gesetzt, wird nach dem Öffnen des Bausteins der Online-Modus gestartet
- KOPAPI_FLAG_TIAPORTAL_DONT_USE_MODIFIED_PROJECT 0x0008L: Wenn dieses Bit gesetzt ist, wird der Aufruf abgebrochen, wenn das Projekt bereits geöffnet ist und Änderungen enthält.

lpszTiaPortalProjectPath

Name der Projektdatei einschließlich der Angabe des absoluten Pfades, z. B. "D:\TIAProjects\Project1\Project1.ap12"

Bitte beachten Sie, dass innerhalb eines C-Skriptes der Backslash über eine Escape-Sequenz geschrieben werden muss:

```
FunctionX(..., "D:\\TIAProjects\\Project1\\Project1.ap12", ...);
```

IpszCpuName

Name der S7-CPU. Der Name ist identisch mit dem Stationsnamen der in der Projektnavigation im TIA Portal angezeigt wird.

IpszContainingBlock

Name des Bausteins der geöffnet und angezeigt werden soll oder Name der Instanz eines FBs.

Als Name können verwendet werden:

- Name eines Singleinstanz DBs. Es wird dann dessen FB angezeigt. Beispiel "Station1"
- Name einer Multiinstanz in einem Instanz-DB. Es wird dann deren FB angezeigt. Bei der Angabe von Multiinstanz-Namenspfaden handelt es sich um die Datenhierarchie, wie sie z.B. im DB-Editor angezeigt wird und nicht um die Call-Structure. Der erste Teil des Namens ("Line1") muss nicht in Anführungszeichen gesetzt werden, da aus dem Kontext erkenntlich ist, dass es sich um ein globales Symbol handelt. Anführungszeichen sind für die einzelnen Namenskomponenten notwendig, wenn Sonderzeichen wie Leerzeichen, Punkt, etc. darin vorkommen. Beispiel: "Line1.Cell1.Station1"
- Name eines FCs oder OBs

Die Verwendung des Names eines FBs ist nicht zulässig.

IpszCalledBlock

Name der lokalen oder globalen Instanz, die im zu IpszContainingBlock gehöriem Code-Baustein aufgerufen wird.

- Bei lokalen Instanzen muss hier das Hash-Zeichen # mit angegeben werden, z.B. "#feeder1".
- Bei globalen Instanz DBs muss der globale Name ohne das Hash-Zeichen # angegeben werden, z.B. "feeder3".

Die Verwendung des Namens eines FCs ist zulässig.

Wenn IpszCalledBlock innerhalb von IpszContainingBlock, bzw. dessen FB, mehrfach aufgerufen wird, dann wird immer zum ersten Aufruf von IpszCalledBlock gesprungen.

Mit IpszCalledBlock=NULL wird nur IpszContainingBlock geöffnet und im Status angezeigt, aber nicht nach einem bestimmten Bausteinaufruf oder einem bestimmten Netzwerk gesucht. In diesem Fall wird IpszPin ignoriert.

IpszPin

Name eines Input-Pins von IpszCalledBlock. Der Parameter dient dazu, dass im Editor innerhalb des Netzwerks der spezifizierte Pin sichtbar angezeigt wird.

Mit IpszPin=NULL wird nur IpszCalledBlock sichtbar angezeigt.

lpszErrorTag

Name einer internen WinCC Variablen vom Datentyp String. Beim Aufruf von asynchronen Funktionen, die nicht sofort ein Ergebnis liefern, werden in lpszErrorTag (Seite 1134) die Fehlerinformationen zurückgeliefert.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

kopapi.h

kopapi.lib

kopapi.dll

Siehe auch

Grundlagen (Seite 1111)

2.11.1.4 OpenTIAPortalIECPLByAssignment (RT Professional)

Beschreibung

Die Funktion wird für die Sprachen KOP und FUP verwendet und zeigt die Zuweisung auf einen Operanden und dessen Vorverschaltung.

Deklaration

```
BOOL OpenTIAPortalIECPLByAssignment (
    DWORD          dwFlags,
    LPCTSTR        lpszTiaPortalProjectPath,
    LPCTSTR        lpszCpuName,
    LPCTSTR        lpszContainingBlock,
    LPCTSTR        lpszOperand,
    LPCTSTR        lpszErrorTag,
    LPCMN_ERROR    lpdmError);
```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

- KOPAPI_FLAG_TIAPORTAL_SUPPRESS_PROGRAM_STATUS=0x0004: Das TIA Portal geht nach dem Öffnen des Bausteins nicht in den Online Modus. Ist dieses Bit nicht gesetzt, wird nach dem Öffnen des Bausteins der Online-Modus gestartet
- KOPAPI_FLAG_TIAPORTAL_DONT_USE_MODIFIED_PROJECT 0x0008L: Wenn dieses Bit gesetzt ist, wird der Aufruf abgebrochen, wenn das Projekt bereits geöffnet ist und Änderungen enthält.

IpszTiaPortalProjectPath

Name der Projektdatei einschließlich der Angabe des absoluten Pfades, z. B. "D:\TIAProjects\Project1\Project1.ap12"

Bitte beachten Sie, dass innerhalb eines C-Skriptes der Backslash über eine Escape-Sequenz geschrieben werden muss:

```
FunctionX(..., "D:\\TIAProjects\\Project1\\Project1.ap12", ...);
```

IpszCpuName

Name der S7-CPU. Der Name ist identisch mit dem Stationsnamen der in der Projektnavigation im TIA Portal angezeigt wird.

IpszContainingBlock

Name des Bausteins der geöffnet und angezeigt werden soll oder Name der Instanz eines FBs.

Als Name können verwendet werden:

- Name eines Singleinstanz DBs. Es wird dann dessen FB angezeigt. Beispiel "Station1"
- Name einer Multiinstanz in einem Instanz-DB. Es wird dann deren FB angezeigt. Bei der Angabe von Multiinstanz-Namenspfaden handelt es sich um die Datenhierarchie, wie sie z.B. im DB-Editor angezeigt wird und nicht um die Call-Structure. Der erste Teil des Namens ("Line1") muss nicht in Anführungszeichen gesetzt werden, da aus dem Kontext erkenntlich ist, dass es sich um ein globales Symbol handelt. Anführungszeichen sind für die einzelnen Namenskomponenten notwendig, wenn Sonderzeichen wie Leerzeichen, Punkt, etc. darin vorkommen. Beispiel: "Line1.Cell1.Station1"
- Name eines FCs oder OBs

Die Verwendung des Names eines FBs ist nicht zulässig.

IpszOperand

Name eines lokalen oder globalen Operanden auf den zugewiesen wird.

Name der lokalen oder globalen Instanz, die im zu IpszContainingBlock gehörigem Code-Baustein aufgerufen wird.

- Bei lokalen Operanden muss hier das Hash-Zeichen # mit angegeben werden.
- Bei globalen Operanden muss der globale Name ohne das Hash-Zeichen # angegeben werden.

Wenn IpszOperand innerhalb von IpszContainingBlock, bzw. dessen FB, mehrfach geschrieben wird, dann wird immer zum ersten Schreibzugriff von IpszOperand gesprungen.

IpszErrorTag

Name einer internen WinCC Variablen vom Datentyp String. Beim Aufruf von asynchronen Funktionen, die nicht sofort ein Ergebnis liefern, werden in IpszErrorTag (Seite 1134) die Fehlerinformationen zurückgeliefert.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

kopapi.h

kopapi.lib

kopapi.dll

Siehe auch

Grundlagen (Seite 1111)

2.11.1.5 OpenTIAPortalS7GraphByBlock (RT Professional)

Beschreibung

Die Funktion wird für die Sprachen S7 Graph verwendet und zeigt einen Schritt in einer Ablaufkette.

Deklaration

```

BOOL OpenTIAPortalS7GraphByBlock (
    DWORD          dwFlags,
    LPCTSTR        lpszTiaPortalProjectPath,
    LPCTSTR        lpszCpuName,
    LPCTSTR        lpszBlock,
    DWORD          dwStepNumber,
    LPCTSTR        lpszErrorTag,
    LPCMN_ERROR    lpdmError);

```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

- KOPAPI_FLAG_TIAPORTAL_SUPPRESS_PROGRAM_STATUS=0x0004: Das TIA Portal geht nach dem Öffnen des Bausteins nicht in den Online Modus. Ist dieses Bit nicht gesetzt, wird nach dem Öffnen des Bausteins der Online-Modus gestartet
- KOPAPI_FLAG_TIAPORTAL_DONT_USE_MODIFIED_PROJECT 0x00000008L: Wenn dieses Bit gesetzt ist, wird der Aufruf abgebrochen, wenn das Projekt bereits geöffnet ist und Änderungen enthält.

lpszTiaPortalProjectPath

Name der Projektdatei einschließlich der Angabe des absoluten Pfades, z. B. "D:\TIAProjects\Project1\Project1.ap12"

Bitte beachten Sie, dass innerhalb eines C-Skriptes der Backslash über eine Escape-Sequenz geschrieben werden muss:

```
FunctionX(..., "D:\\TIAProjects\\Project1\\Project1.ap12", ...);
```

lpszCpuName

Name der S7-CPU. Der Name ist identisch mit dem Stationsnamen der in der Projektnavigation im TIA Portal angezeigt wird.

lpszBlock

Instanzname des S7-Graph Bausteins der angezeigt werden soll.

dwStepNumber

Nummer des Schritts der angezeigt werden soll

Mit dwStepNumber=0 wird der aktive Schritt automatisch gesucht und der Modus "Track active step" aktiviert.

lpszErrorTag

Name einer internen WinCC Variablen vom Datentyp String. Beim Aufruf von asynchronen Funktionen, die nicht sofort ein Ergebnis liefern, werden in lpszErrorTag (Seite 1134) die Fehlerinformationen zurückgeliefert.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

kopapi.h

kopapi.lib

kopapi.dll

Siehe auch

Grundlagen (Seite 1111)

2.11.1.6 Beispiel: Einbinden in eine WinCC Funktion (RT Professional)

Beschreibung

Das Beispiel zeigt die Einbettung der Funktionsaufrufe in eine benutzerdefinierte Funktion. Der Aufruf sollte wegen der besseren Wartbarkeit in eine Global Script Funktion ausgelagert werden, in welcher dann auch der Pfad zum TIA Portal und der Name der PLC Station bestimmt werden.

```
void OnClick(char* screenName, char* objectName, char* propertyName)
{
    // Funktionen bekannt machen'
    #pragma code("KOPAPI.dll")
    #include "KOPAPI.h"
    #pragma code()

    BOOL useTiaPortal = TRUE; // use the TIA Portal or the viewer control?
    char* pTiaPortalProject = "c:\\Projects\\myproject\\myproject.ap12";
    char* pStationName = "OW063IZ1M01_PLC2_STD"; // TODO: read from internal
tag
    char* pContainingBlock = "SG01_FG02_ST001+FX001_IDB"; // TODO: get from
current selection
    char* pOperand = "#HOME_SEQ1.HOME_MISC[8]"; // TODO: get from current
selection
    char* pErrorTag = "IECPLViewerErrorTag"; // TODO: define an own tag
    CMN_ERROR error;
    BOOL result;

    if(useTiaPortal)
    {
        result = OpenTIAPortalIECPLByAssignment(0, pTiaPortalProject,
pStationName, pContainingBlock, pOperand, pErrorTag, &error);
    }
    else
    {
        char *pServerPrefix = NULL, *pTagPrefix = NULL, *pWindowPrefix =
NULL;

        // determine ServerPrefix of the current environment
        GetServerTagPrefix(&pServerPrefix, &pTagPrefix, &pWindowPrefix);
        // make the screen which contains the viewer control visible
        SetVisible("SYSTEM#Basic_Screen", "Screen_window_IECPLViewer",
TRUE);
        result = OpenViewerIECPLByAssignment(0, pServerPrefix,
"SYSTEM#IECPLViewer", "IECPLViewerObject", pStationName, pContainingBlock,
pOperand, pErrorTag, &error);
    }
}
```

Siehe auch

Grundlagen (Seite 1111)

2.11.2 Anzeige in der PLC-Code-Anzeige (RT Professional)

2.11.2.1 Grundlagen (RT Professional)

Allgemeines

Mit diesen Funktionen können Sie den aktuellen Programmstatus von PLC-Programmen in einer PLC-Code-Anzeige darstellen.

Das Bild, welches die PLC-Code-Anzeige enthält, muss z. B. über `OpenPicture()` oder `SetPictureName()` geöffnet worden sein.

Header-Dateien

Die Deklaration der hier erläuterten Funktionen und Strukturen erfolgt in den Header-Dateien:

kopapi.h	API Schnittstellendefinition Header-Datei
----------	---

Librarys

Die DLL-Anbindung der hier erläuterten Funktionen erfolgt in den Lib-Dateien:pdecsccli.dll

kopapi.lib	Library
kopapi.dll	

Hinweis zur Angabe von Multiinstanznamen

Bei der Angabe von Multiinstanz-Namenspfaden handelt es sich um die Datenhierarchie, wie sie z.B. im DB-Editor angezeigt wird und nicht um die Call-Structure. Der erste Teil des Namens ("Line1") muss nicht in Anführungszeichen gesetzt werden, da aus dem Kontext erkenntlich ist, dass es sich um ein globales Symbol handelt. Anführungszeichen sind für die einzelnen Namenskomponenten notwendig, wenn Sonderzeichen wie Leerzeichen, Punkt, etc. darin vorkommen. Beispiel: "Line1.Cell1.Station1". Namen, die keine Sonderzeichen enthalten, dürfen nicht in Anführungszeichen stehen.

Siehe auch

[OpenViewerIECPLByCall](#) (Seite 1125)

[OpenViewerIECPLByAssignment](#) (Seite 1132)

[OpenViewerS7GraphByBlock](#) (Seite 1123)

2.11.2.2 OpenViewerS7GraphByBlock (RT Professional)

Beschreibung

Die Funktion zeigt einen in der PLC Sprache S7 Graph aufgerufenen Schritt aus einer Ablaufkette in der PLC-Code-Anzeige an.

Deklaration

```
BOOL OpenViewerS7GraphByBlock (  
    DWORD          dwFlags,  
    LPCTSTR        lpszServerPrefix,  
    LPCTSTR        lpszPictureName,  
    LPCTSTR        lpszObjectName,  
    LPCTSTR        lpszCpuName,  
    LPCTSTR        lpszBlock,  
    DWORD          dwStepNumber,  
    LPCMN_ERROR    lpdmError);
```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

lpszServerPrefix

Der Parameter ist für spätere Erweiterungen reserviert.

lpszPictureName

Name des Bildes mit der PLC-Code-Anzeige.

lpszObjectName

Name der PLC-Code-Anzeige.

lpszCpuName

Name der S7-CPU. Der Name ist identisch mit dem Stationsnamen, der in der Projektnavigation im TIA Portal angezeigt wird.

lpszBlock

Instanzname des S7-Graph Bausteins, der angezeigt werden soll. Wenn im Namen Sonderzeichen wie Leerzeichen, Punkt, etc. vorkommen, müssen Anführungszeichen verwendet werden.

dwStepNumber

Nummer des Schritts, der angezeigt werden soll.

Mit `dwStepNumber=0` wird der aktive Schritt automatisch gesucht und der Modus "Track active step" aktiviert.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur `CMN_ERROR`. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

`kopapi.h`

`kopapi.lib`

`kopapi.dll`

Beispiel

Das Beispiel zeigt die Einbettung des Funktionsaufrufs in einer benutzerdefinierten C-Funktion. Der Aufruf sollte wegen der besseren Wartbarkeit in eine Global Script Funktion ausgelagert werden, in welcher dann auch der Name der PLC-Station bestimmt wird.

```
BOOL OpenCodeViewerByS7Graph(char* screenName, char* objectName, char*
cpuName, char* instanceDBName)
{
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()
BOOL result;
CMN_ERROR error;
char* serverPrefix = "";
DWORD dwFlags = 0;
DWORD stepNumber = 0;
result = OpenViewerS7GraphByBlock(dwFlags, serverPrefix, screenName,
objectName,
cpuName, instanceDBName, stepNumber, &error);
if(!result)
{
// there are only few reasons why the call to OpenViewerIECPLByAssignment
will // fail, in most cases the viewer control could not be found
// most of the errors have to be handled in the OnError event of the viewer
printf("OpenViewerS7GraphByBlock failed: err1=%ld, err2=%ld, err3=%ld,
err4=%ld, err5=%ld, text=\"%s\"\r\n", result, error.dwError1,
error.dwError2, error.dwError3, error.dwError4, error.dwError5,
error.szErrorText);
}
return result;
}
```

Siehe auch

Fehlerbehandlung (Seite 1134)

Grundlagen (Seite 1122)

2.11.2.3 OpenViewerIECPLByCall (RT Professional)

Beschreibung

Die Funktion zeigt in den PLC-Sprachen KOP und FUP die Vorverschaltung eines Netzwerkeingangs eines Standardbausteins in der PLC-Code-Anzeige an.

Hinweis

Wenn der Eingang mit einer Konstanten (TRUE, FALSE) belegt ist, wird dieser Wert in der PLC-Code-Anzeige nicht angezeigt werden. Um den Wert in der PLC-Code-Anzeige richtig anzuzeigen, sollten Sie den Eingang mit einer Variablen belegen, die den entsprechenden Wert besitzt.

Deklaration

```
BOOL OpenViewerIECPLByCall (  
    DWORD        dwFlags,  
    LPCTSTR      lpszServerPrefix,  
    LPCTSTR      lpszPictureName,  
    LPCTSTR      lpszObjectName,  
    LPCTSTR      lpszCpuName,  
    LPCTSTR      lpszContainingBlock,  
    LPCTSTR      lpszCalledBlock,  
    LPCTSTR      lpszPin,  
    LPCMN_ERROR  lpdmError);
```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

- IECPLVIEWER_PIN_SUBSTRING_SEARCH=0x0001: Bei der Suche nach dem Pin-Namen wird nach einem Teilstring gesucht, d. h. der Pin-Name beginnt mit der in lpszPin übergebenen Zeichenkette. Ist dieses Bit nicht gesetzt, wird der vollständige Pin-Name mit lpszPin verglichen.

lpszServerPrefix

Der Parameter ist für spätere Erweiterungen reserviert.

lpszPictureName

Name des Bildes mit der PLC-Code-Anzeige.

lpszObjectName

Name der PLC-Code-Anzeige.

lpszCpuName

Name der S7-CPU. Der Name ist identisch mit dem Stationsnamen, der in der Projektnavigation im TIA Portal angezeigt wird.

IpszContainingBlock

Name des Bausteins der geöffnet und angezeigt werden soll oder Name der Instanz eines FBs.

Als Name können verwendet werden:

- Name eines Singleinstanz DBs. Es wird dann dessen FB angezeigt. Beispiel "Station1"
- Name einer Multiinstanz in einem Instanz-DB. Es wird dann deren FB angezeigt. Bei der Angabe von Multiinstanz-Namenspfaden handelt es sich um die Datenhierarchie, wie sie z. B. im DB-Editor angezeigt wird und nicht um die Call-Structure. Der erste Teil des Namens ("Line1") muss nicht in Anführungszeichen gesetzt werden, da aus dem Kontext erkenntlich ist, dass es sich um ein globales Symbol handelt. Anführungszeichen sind für die einzelnen Namenskomponenten notwendig, wenn Sonderzeichen wie Leerzeichen, Punkt, etc. darin vorkommen. Beispiel: "Line1.Cell1.Station1"
- Name eines FCs oder OBs

Die Verwendung des Names eines FBs ist nicht zulässig.

IpszCalledBlock

Name der lokalen oder globalen Instanz, die im zu IpszContainingBlock gehörigem Code-Baustein aufgerufen wird.

- Bei lokalen Instanzen muss hier das Hash-Zeichen # mit angegeben werden, z.B. "#feeder1".
- Bei globalen Instanz DBs muss der globale Name ohne das Hash-Zeichen # angegeben werden, z.B. "feeder3".

Die Verwendung des Namens eines FCs ist zulässig.

Wenn IpszCalledBlock innerhalb von IpszContainingBlock, bzw. dessen FB, mehrfach aufgerufen wird, dann wird immer zum ersten Aufruf von IpszCalledBlock gesprungen.

IpszPin

Name eines Input-Pins von IpszCalledBlock. Der Parameter dient dazu, dass in der PLC-Code-Anzeige das Netzwerk angezeigt wird, das mit dem Input-Pin verschaltet ist.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert**TRUE**

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

kopapi.h
kopapi.lib
kopapi.dll

Beispiel

Das Beispiel zeigt die Einbettung des Funktionsaufrufs in einer benutzerdefinierten C-Funktion. Der Aufruf sollte wegen der besseren Wartbarkeit in eine Global Script Funktion ausgelagert werden, in welcher dann auch der Name der PLC-Station bestimmt wird.

```
BOOL OpenCodeViewerByCall(BOOL matchSubstringPin, char* screenName, char*
objectName, char* cpuName, char* containingBlock, char* calledBlock, char*
pinName)
{
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()
BOOL result;
CMN_ERROR error;
char* serverPrefix = "";
DWORD dwFlags = 0;
if(matchSubstringPin)
    dwFlags |= KOPAPI_FLAG_TIAPORTAL_PIN_SUBSTRING_SEARCH;
result = OpenViewerIECPLByCall(dwFlags, serverPrefix, screenName,
objectName,
    cpuName, containingBlock, calledBlock, pinName, &error);
if(!result)
{
    // there are only few reasons why the call to OpenViewerIECPLByCall will
fail, in // most cases the viewer control could not be found
    // most of the errors have to be handled in the OnError event of the viewer
    printf("OpenViewerIECPLByCall failed: err1=%ld, err2=%ld, err3=%ld, err4=
%ld, err5=%ld, text=\"%s\"\\r\\n", result, error.dwError1,
error.dwError2, error.dwError3, error.dwError4, error.dwError5,
error.szErrorText);
}
return result;
}
```

Siehe auch

Fehlerbehandlung (Seite 1134)

Grundlagen (Seite 1122)

2.11.2.4 OpenViewerIECPLByFCCall (RT Professional)

Beschreibung

Die Funktion zeigt für die PLC-Sprachen KOP und FUP die Vorverschaltung eines Netzwerkeingangs eines Standardbausteins unter Berücksichtigung der UDT-Instanz in der PLC-Code-Anzeige an.

Hinweis

Wenn der Eingang mit einer Konstanten (TRUE, FALSE) belegt ist, wird dieser Wert in der PLC-Code-Anzeige nicht angezeigt werden. Um den Wert in der PLC-Code-Anzeige richtig anzuzeigen, sollten Sie den Eingang mit einer Variablen belegen, die den entsprechenden Wert besitzt.

Deklaration

```
BOOL OpenViewerIECPLByFCCall (  
    DWORD          dwFlags,  
    LPCTSTR        lpszServerPrefix,  
    LPCTSTR        lpszPictureName,  
    LPCTSTR        lpszObjectName,  
    LPCTSTR        lpszCpuName,  
    LPCTSTR        lpszContainingBlock,  
    LPCTSTR        lpszCalledBlock,  
    LPCTSTR        lpszPin,  
    LPCTSTR        lpszUdtInstance  
    LPCMN_ERROR    lpdmError);
```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

- IECPLVIEWER_PIN_SUBSTRING_SEARCH=0x0001: Bei der Suche nach dem Pin-Namen wird nach einem Teilstring gesucht, d. h. der Pin-Name beginnt mit der in lpszPin übergebenen Zeichenkette. Ist dieses Bit nicht gesetzt, wird der vollständige Pin-Name mit lpszPin verglichen.

lpszServerPrefix

Der Parameter ist für spätere Erweiterungen reserviert und sollte ein Leerstring ("") sein.

lpszPictureName

Name des Bildes mit der PLC-Code-Anzeige.

lpszObjectName

Name der PLC-Code-Anzeige.

lpszCpuName

Name der S7-CPU. Der Name ist identisch mit dem Stationsnamen, der in der Projektnavigation im TIA Portal angezeigt wird.

lpszContainingBlock

Name des Bausteins, der geöffnet und angezeigt werden soll, oder Name der Instanz eines FBs.

Als Name können verwendet werden:

- Name eines Singleinstanz DBs. Es wird dann dessen FB angezeigt. Beispiel "Station1"
- Name einer Multiinstanz in einem Instanz-DB. Es wird dann deren FB angezeigt. Bei der Angabe von Multiinstanz-Namenspfaden handelt es sich um die Datenhierarchie, wie sie z. B. im DB-Editor angezeigt wird und nicht um die Call-Structure. Der erste Teil des Namens ("Line1") muss nicht in Anführungszeichen gesetzt werden, da aus dem Kontext erkenntlich ist, dass es sich um ein globales Symbol handelt. Anführungszeichen sind für die einzelnen Namenskomponenten notwendig, wenn Sonderzeichen wie Leerzeichen, Punkt, etc. darin vorkommen. Beispiel: "Line1.Cell1.Station1"
- Name eines FCs oder OBs

Die Verwendung des Names eines FBs ist nicht zulässig.

lpszCalledBlock

Name der lokalen oder globalen Instanz, die im zu lpszContainingBlock gehörigem Code-Baustein aufgerufen wird.

- Bei lokalen Instanzen muss hier das Hash-Zeichen # mit angegeben werden, z.B. "#feeder1".
- Bei globalen Instanz DBs muss der globale Name ohne das Hash-Zeichen # angegeben werden, z.B. "feeder3".

Die Verwendung des Namens eines FCs ist zulässig.

Wenn lpszCalledBlock innerhalb von lpszContainingBlock, bzw. dessen FB, mehrfach aufgerufen wird, dann wird immer zum ersten Aufruf von lpszCalledBlock gesprungen.

lpszPin

Name eines Input-Pins von lpszCalledBlock. Der Parameter dient dazu, dass in der PLC-Code-Anzeige das Netzwerk angezeigt wird, das mit dem Input-Pin verschaltet ist.

lpszUdtInstance

Mit dem Parameter kann die Anzeige von mehrfach aufgerufenen FBs oder FCs eingegrenzt werden. Die Eingrenzung erfolgt anhand der an einem beliebigen Input-Pin oder InOut-Pin verschalteten UDT-Instanz.

lpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

kopapi.h

kopapi.lib

kopapi.dll

Beispiel

Das Beispiel zeigt die Einbettung des Funktionsaufrufs in einer benutzerdefinierten C-Funktion. Der Aufruf sollte wegen der besseren Wartbarkeit in eine Global Script Funktion ausgelagert werden, in welcher dann auch der Name der PLC-Station bestimmt wird.

```

BOOL OpenCodeViewerByFCCall(BOOL matchSubstringPin, char* screenName,
char* objectName, char* cpuName, char* containingBlock, char* calledBlock,
char* pinName, char* udtInstance)
{
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()
BOOL result;
CMN_ERROR error;
char* serverPrefix = "";
DWORD dwFlags = 0;
if(matchSubstringPin)
    dwFlags |= KOPAPI_FLAG_TIAPORTAL_PIN_SUBSTRING_SEARCH;
result = OpenViewerIECPLByFCCall(dwFlags, serverPrefix, screenName,
objectName, cpuName, containingBlock, calledBlock, pinName, udtInstance,
&error);
if(!result)
{
    // there are only few reasons why the call to OpenViewerIECPLByFCCall will
    fail,
    // in most cases the viewer control could not be found
    // most of the errors have to be handled in the OnError event of the viewer
    control
    printf("OpenViewerIECPLByFCCall failed: err1=%ld, err2=%ld, "" err3=%ld,
err4=%ld, err5=%ld, text=\"%s\"\r\n", result, error.dwError1,
error.dwError2, error.dwError3, error.dwError4, error.dwError5,
error.szErrorText);
}
return result;
}

```

2.11.2.5 OpenViewerIECPLByAssignment (RT Professional)

Beschreibung

Die Funktion zeigt für die PLC-Sprachen KOP und FUP die Zuweisung auf einen Operanden und dessen Vorverschaltung in der PLC-Code-Anzeige an.

Deklaration

```
BOOL OpenViewerIECPLByAssignment (  
    DWORD          dwFlags,  
    LPCTSTR        lpszServerPrefix,  
    LPCTSTR        lpszPictureName,  
    LPCTSTR        lpszObjectName,  
    LPCTSTR        lpszCpuName,  
    LPCTSTR        lpszContainingBlock,  
    LPCTSTR        lpszOperand,  
    LPCMN_ERROR    lpdmError);
```

Parameter

dwFlags

Bitfeld, in dem die einzelnen Werte bitweise mit ODER verknüpft werden. Standardmäßig sollte dwFlags 0 sein.

lpszServerPrefix

Der Parameter ist für spätere Erweiterungen reserviert.

lpszPictureName

Name des Bildes mit der PLC-Code-Anzeige.

lpszObjectName

Name der PLC-Code-Anzeige.

lpszCpuName

Name der S7-CPU. Der Name ist identisch mit dem Stationsnamen, der in der Projektnavigation im TIA Portal angezeigt wird.

lpszContainingBlock

Name des Bausteins, der geöffnet und angezeigt werden soll oder Name der Instanz eines FBs.

Als Name können verwendet werden:

- Name eines Singleinstanz DBs. Es wird dann dessen FB angezeigt. Beispiel "Station1"
- Name einer Multiinstanz in einem Instanz-DB. Es wird dann deren FB angezeigt. Bei der Angabe von Multiinstanz-Namenspfaden handelt es sich um die Datenhierarchie, wie sie z.B. im DB-Editor angezeigt wird und nicht um die Call-Structure. Der erste Teil des Namens ("Line1") muss nicht in Anführungszeichen gesetzt werden, da aus dem Kontext erkenntlich ist, dass es sich um ein globales Symbol handelt. Anführungszeichen sind für die einzelnen Namenskomponenten notwendig, wenn Sonderzeichen wie Leerzeichen, Punkt, etc. darin vorkommen. Beispiel: "Line1.Cell1.Station1"
- Name eines FCs oder OBs

Die Verwendung des Namens eines FBs ist nicht zulässig.

IpszOperand

Name eines lokalen oder globalen Operanden auf den zugewiesen wird.

Name der lokalen oder globalen Instanz, die im zu IpszContainingBlock gehörigem Code-Baustein aufgerufen wird.

- Bei lokalen Operanden muss hier das Hash-Zeichen # mit angegeben werden.
- Bei globalen Operanden muss der globale Name ohne das Hash-Zeichen # angegeben werden.

Wenn IpszOperand innerhalb von IpszContainingBlock, bzw. dessen FB, mehrfach geschrieben wird, dann wird immer zum ersten Schreibzugriff von IpszOperand gesprungen.

IpdmError

Zeiger auf die Daten der erweiterten Fehlermeldung in der Struktur CMN_ERROR. Im Fehlerfall schreibt das System Fehlerinformationen in diese Struktur.

Rückgabewert

TRUE

Die Funktion wurde fehlerfrei durchlaufen.

FALSE

Es ist ein Fehler aufgetreten.

Benötigte Dateien

kopapi.h

kopapi.lib

kopapi.dll

Beispiel

Das Beispiel zeigt die Einbettung des Funktionsaufrufs in einer benutzerdefinierten C-Funktion. Der Aufruf sollte wegen der besseren Wartbarkeit in eine Global Script Funktion ausgelagert werden, in welcher dann auch der Name der PLC Station bestimmt wird.

```

BOOL OpenCodeViewerByAssignment(char* screenName, char* objectName, char*
cpuName, char* containingBlock, char* operand)
{
#pragma code("kopapi.dll")
#include "kopapi.h"
#pragma code()
BOOL result;
CMN_ERROR error;
char* serverPrefix = "";
DWORD dwFlags = 0;
result = OpenViewerIECPLByAssignment(dwFlags, serverPrefix, screenName,
objectName, cpuName, containingBlock, operand, &error);
if(!result)
{
// there are only few reasons why the call to OpenViewerIECPLByAssignment
will
// fail, in most cases the viewer control could not be found
// most of the errors have to be handled in the OnError event of the viewer
printf("OpenViewerIECPLByAssignment failed: err1=%ld, err2=%ld, err3=%ld,
err4=%ld, err5=%ld, text=\"%s\"\\r\\n", result, error.dwError1,
error.dwError2, error.dwError3, error.dwError4, error.dwError5,
error.szErrorText);
}
return result;
}

```

Siehe auch

Fehlerbehandlung (Seite 1134)

Grundlagen (Seite 1122)

2.11.3 Fehlerbehandlung (RT Professional)

Allgemeines

Die Abarbeitung der Funktionen zum Netzwerkeinsprung ist aufgeteilt in einen synchronen Anteil und einen asynchronen Anteil. Der synchrone Anteil überprüft die Parameter und übergibt sie an den asynchronen Anteil.

Der Rückgabewert der Funktionen gibt an, ob im synchronen Anteil der Funktion ein Fehler aufgetreten ist. Im Fehlerfall ist der Rückgabewert FALSE. Derartige Fehler treten auf, wenn Parameter, z. B. der Parameter `IpszErrorTag` nicht oder falsch versorgt werden.

Fehler, die im asynchronen Anteil der Funktionen auftreten werden über den Parameter `IpszErrorTag` zurückgemeldet. Derartige Fehler können z. B. sein: Projekt nicht gefunden,

Baustein nicht vorhanden, ... IpszErrorTag enthält den Name einer Variablen vom Datentyp String. Um auf den Rückgabewert zu reagieren müssen Sie eine Funktion zu projektieren, die auf Wertänderung der Fehlervariablen mit dem Zyklus "Auf Änderung" getriggert wird. Die Fehlervariable sollte auch dazu verwendet werden, weitere Funktionsaufrufe oder die Bedienung von z. B. einer Schaltfläche zu verhindern, solange die Fehlervariable den Wert "RUNNING" besitzt.

IpszErrorTag

Name einer Variablen vom Datentyp String. Wenn die Variable nicht benötigt wird, kann NULL als Parameter übergeben werden. Bei Bedarf kann der Variablenname einen ServerPrefix enthalten.

Der Wert der Fehlervariablen wird wie folgt geändert:

- Wenn der asynchrone Teil gestartet wird, wird der Wert der Fehlervariablen auf "RUNNING" gesetzt.
- Wenn der asynchrone Teil ohne Fehler beendet wurde, wird der Wert der Fehlervariablen auf "OK" gesetzt.
- Wenn der asynchrone Teil mit einem Fehler beendet wurde, befindet sich in der Fehlervariablen ein mehrzeiliger String, dessen Zeilen durch Linefeed-Zeichen ('\n') getrennt sind.

Im Fehlerfall besitzt die Fehlervariable folgenden Aufbau:

- Zeile 1: "ERROR"
- Zeile 2 - Zeile 6: Dezimalzahlen, Datentyp: 32 Bit vorzeichenlos; DWORD
- Zeile 7: Fehlertext

Fehlertext

Wenn der asynchrone Teil mit einem Fehler beendet wurde, befindet sich in der Fehlervariablen ein mehrzeiliger String, dessen Zeilen durch Linefeed-Zeichen ('\n') getrennt sind. Die siebte Zeile enthält einen der folgenden Fehlertexte:

IDS_E_IS_TIA_PROJECT	KOPAPI: In einem TIA Portal-Projekt kann diese Funktion nicht verwendet werden!
IDS_E_NO_TIA_PROJECT	KOPAPI: Der Funktionsaufruf ist gesperrt, da kein TIA Portal-Projekt geöffnet ist!
IDS_E_TIAPORTAL_UNKNOWN_FLAGS	dwFlags enthält einen undefinierten Wert.
IDS_E_TIAPORTAL_ERRORTAG_NOT_EXIST	Die Fehlervariable [%s] existiert nicht.
IDS_E_TIAPORTAL_PREVIOUS_CALL_IS_RUNNING	Der vorherige Aufruf läuft noch (RUNNING), bitte diesen zuerst beenden.
IDS_E_TIAPORTAL_CANNOT_WRITE_ERRORTAG	Die Fehlervariable [%s] kann nicht geschrieben werden.
IDS_E_TIAPORTAL_COMACCESS_REGIS-TERPS_FAILED	Keine Anbindung an TIA Portal möglich.

IDS_E_TIAPORTAL_CANNOT_START_PORTAL	TIA Portal kann nicht gestartet werden.
IDS_E_TIAPORTAL_CANNOT_SEARCH_START- ED_PORTAL	Es kann kein gestartetes Portal gesucht werden.
IDS_E_TIAPORTAL_NO_PORTAL_STARTED	Es konnte kein Portal gestartet werden.
IDS_E_TIAPORTAL_EXCEPTION_SYNC_PART	Exception im synchronen Bearbeitungsteil.
IDS_E_TIAPORTAL_EXCEPTION_ASYNC_PART	Exception im asynchronen Bearbeitungsteil.
IDS_E_TIAPORTAL_NOT_INSTALLED	Kein TIA Portal installiert.
IDS_E_TIAPORTAL_PROJECT_CANNOT_OPEN	Projekt [%s] kann nicht geöffnet werden. Projekt [%s] kann nicht geöffnet werden.
IDS_E_TIAPORTAL_ALREADY_OPEN- ED_WITH_OTHER_PROJECT	TIA-Portal ist mit anderem Projekt geöffnet.
IDS_E_TIAPORTAL_CREATECOMMAND	CreateCommand Fehler.
IDS_E_TIAPORTAL_COMMAND_ADDARGUMENT	Command AddArgument Fehler.
IDS_E_TIAPORTAL_EXECUTECOMMAND	Schwerer Fehler in ExecuteCommand.
IDS_E_TIAPORTAL_COMMAND_ERROR	TIA Projekt Befehls-Fehler: [%s,%ld].
IDS_E_TIAPORTAL_COMMAND_UNKNOWN	TIA Projekt unbekannter Befehl: [%s,%ld].

2.12 Fehlerbehandlung (RT Professional)

2.12.1 CMN_ERROR (RT Professional)

Beschreibung

Die erweiterte Fehlerstruktur CMN_Error enthält den Fehlercode und einen Fehlertext für den aufgetretenen Fehler. Jede Applikation kann die Fehlerstruktur zur Auswertung bzw. zur Ausgabe von Fehlermeldungen nutzen.

Deklaration

```

Typedef struct {
    DWORD    dwError1;
    DWORD    dwError2;
    DWORD    dwError3;
    DWORD    dwError4;
    DWORD    dwError5;
    Char     szErrorText[512];
}
CMN_ERROR;

```

Members

dwError1 .. dwError5

Die API-Beschreibungen enthalten jeweils die Information, welche Werte die Einträge im Fehlerfall enthalten. Soweit nichts Anderes angegeben ist, stehen die Fehlercodes in dwError1.

szErrorText

Puffer für textuelle Beschreibung der Fehlerursache. Der Inhalt wird aus den Ressourcen ermittelt und ist in der Regel sprachabhängig.

Bemerkung

Einige Module verwenden je nach Bedarf folgende Belegungen der Fehlercodes

dwError1	API Fehlercode
dwError2	unterlagerter Fehlercode (z.B. CCStorageError.h)
dwError3	Zeilennummer des Quellcodes
dwError4	Array Index 1; Reihe; Record / Status /
dwError5	Array Index 2; Spalte; Record Element / Zähler /

Siehe auch

Fehlermeldungen (Seite 206)

2.12.2 CCStorageError.h (RT Professional)

Unterlagerte Fehlermeldungen CCStorageError.h

Die folgenden Fehlermeldungen können von den API-Funktionen in der Fehler-Struktur CMN_ERROR zusätzlich zur eigentlichen Fehlermeldung in dwError1 und in dwError2 zurückgegeben werden, sofern im jeweiligen Modul verfügbar.

Fehlermeldung	Wert	
CCF_STORAGE_PM_E_NO_PROJECT_OPEN	0x80046101L	Cannot open WinCC project
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_1	0x80046102L	Cannot open/create WinCC project - Project Manager can't start basis application SDIAGRT
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_2	0x80046103L	Cannot open/create WinCC project - Project Manager can't start basis application PASSDBRT
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_3	0x80046104L	Cannot open/create WinCC project - Project Manager can't start basis application SCRIPT
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_31	0x80046105L	Cannot open/create WinCC project - Project Manager can't start basis application XREF

Fehlermeldung	Wert	
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_4	0x80046106L	Cannot open/create WinCC project - Project Manager can't initialize ASO
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_5	0x80046107L	Cannot open/create WinCC project - Project Manager can open only one WinCC Project at the same time
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_6	0x80046108L	Cannot open/create WinCC project - Project Manager can't configure startline for the data source
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_7	0x80046109L	Cannot open/create WinCC project - Project Manager can't configure data source
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_8	0x8004610AL	Cannot open WinCC project - Project Manager for configured WinCC Server is not already running - Start Project Manager for configured server
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_9	0x8004610BL	Cannot open/create WinCC project - Project Manager can't create Distributed Configuration File - Project.dcf
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_10	0x8004610CL	Cannot open WinCC project - Project path not valid
CCF_STORAGE_PM_E_NO_PROJECT_CLOSE_1	0x8004610DL	Cannot close WinCC project - Some users in the network are already connected to this WinCC Project
CCF_STORAGE_PM_E_NO_PROJECT_CLOSE_2	0x8004610EL	Cannot close WinCC project - Project Manager don't know something about this WinCC Project - please check your Project ID
CCF_STORAGE_PM_E_NO_PROJECT_CLOSE_3	0x8004610FL	Cannot close WinCC project - An active WinCC project is already running on this computer
CCF_STORAGE_PM_E_NO_PROJECT_CLOSE_4	0x80046110L	Cannot close WinCC project - Project Manager don't support actual close project mode
CCF_STORAGE_PM_E_NO_PROJECT_CLOSE_5	0x80046111L	Cannot close WinCC project - Some users already connected to this WinCC Project - TimeOut!
CCF_STORAGE_PM_E_NO_PROJECT_LANGUAGE_1	0x80046112L	Project don't support desired language!
CCF_STORAGE_PM_E_NO_ACTIVATE	0x80046113L	No active WinCC project is running on this computer
CCF_STORAGE_PM_E_NO_ACTIVATE_1	0x80046114L	Cannot activated WinCC project - WinCC Server properties are changed - close and reopen the WinCC Project please!

Fehlermeldung	Wert	
CCF_STORAGE_PM_E_NO_PROJECT_CREATE	0x80046115L	Cannot create WinCC project
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_11	0x80046116L	Cannot open WinCC project - Close WinCC project is already in process.
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_12	0x80046117L	Cannot open WinCC project - WinCC project path exceed the maximum value of 255 characters.
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_13	0x80046118L	Cannot open WinCC project - WinCC project name exceed the maximum value of 64 characters.
CCF_STORAGE_PM_E_NO_ACTIVATE_2	0x80046119L	Cannot deactivate WinCC project - Time-out interval elapsed - > SCRIPT.EXE
CCF_STORAGE_PM_E_NO_ACTIVATE_3	0x8004611AL	Cannot deactivate WinCC project - Time-out interval elapsed - > PASSDBRT.EXE
CCF_STORAGE_PM_E_NO_ACTIVATE_4	0x8004611BL	Cannot deactivate WinCC project - Time-out interval elapsed - > SDIAGRT.EXE
CCF_STORAGE_PM_E_NO_PROJECT_OPEN_14	0x8004611CL	Cannot open WinCC project - No project path can be insert into simatic.cfg file
CCF_STORAGE_PM_E_NO_ACTIVATE_5	0x8004611DL	Cannot activate WinCC project - No WinCC project !!
CCF_STORAGE_PM_E_NO_ACTIVATE_6	0x8004611EL	Cannot activate WinCC project - WinCC client can't get the server's activation state over an broadcast
CCF_STORAGE_PM_E_NO_ACTIVATE_7	0x8004611FL	Activate for WinCC project in process - WinCC client's activation process not already finished
CCF_STORAGE_PM_E_CREATE_SHARE_FAILED	0x80046120L	Cannot create the WinCC-Project-Share - WinCC-Project cannot be opened.
CCF_STORAGE_PM_E_OPEN_FILE_FAILED	0x80046121L	Cannot open the WinCC-Project-file - WinCC-Project cannot be opened.
CCF_STORAGE_PM_E_CLIENT_CANNOT_ACTIVATE	0x80046122L	This client cannot activate this project,because it is not in the client list.
CCF_STORAGE_PM_E_SYBASE_MIGRATION	0x80046123L	This WinCC version doesn't support old Sybase projects. Use migration.
CCF_STORAGE_PM_E_NO_PROJECT_OPENED	0x80046124L	There is no project opened at the moment.
CCF_STORAGE_PM_E_COMPUTER_NOT_IN_MACHINE_LIST	0x80046125L	This client is not in the machine list of the specified server.

2.12 Fehlerbehandlung (RT Professional)

Fehlermeldung	Wert	
CCF_SQL_SERVER_NOT_INSTALLED_ERROR	0x80046126L	The SQL server may not be correct installed. Please check it.
CCF_SQL_SERVER_DATABASE_ACCESS_ERROR	0x80046127L	The SQL server cannot access database. Please check access rights.
CCF_PM_E_PROJECT_IS_LOCKED	0x80046128L	The Project is locked.
CCF_SQL_SERVER_DATABASE_NTFS_ERROR	0x80046129L	The SQL-SERVER 2005 does not support compressed NTFS volumes. Please uncompress at least the project directory and its contents.
CCF_PM_OPEN_ERROR_RDP	0xC004612AL	This project is generated by RDP and is not allowed to be opened by WinCC
CCF_PM_OPEN_CODEPAGE_MISMATCH	0x4004612BL	The current codepage does not match the codepage at creation time
CCF_PM_MIGRATION_NEEDED	0x8004612CL	
CCF_STORAGE_DBASO_E_NO_LICENSE	0x80046201L	No License for the Functionality
CCF_STORAGE_DBASO_E_LIMIT	0x80046202L	The limit is reached
CCF_STORAGE_DBASO_E_DATA_VERSION	0x80046203L	ASO can not work with this data version
CCF_STORAGE_DBASO_E_NO_CONNECTION	0x80046204L	no connection to database
CCF_STORAGE_DBASO_E_NO_TABLE	0x80046205L	Table doesn't exist or could not be opened. Please check the user rights.
CCF_STORAGE_DBASO_E_NO_RECORD	0x80046206L	no record
CCF_STORAGE_DBASO_E_NO_RESOURCE	0x80046207L	no resource
CCF_STORAGE_DBASO_E_NOT_CREATED	0x80046208L	record not created
CCF_STORAGE_DBASO_E_NOT_DELETED	0x80046209L	record not deleted
CCF_STORAGE_DBASO_E_NOT_MODIFIED	0x8004620AL	record not modified
CCF_STORAGE_DBASO_E_RECORD_EXIST	0x8004620BL	record already exists.
CCF_STORAGE_DBASO_E_RECORD_IN_USE	0x8004620CL	record in use
CCF_STORAGE_DBASO_E_RECORD_FIELDSINVALID	0x8004620DL	Invalid Count or Type of Fields
CCF_STORAGE_DBASO_E_RECORD_PROTECTED	0x8004620EL	Record can not be modified or removed.
CCF_STORAGE_DBASO_E_NO_CONTEXT	0x8004620FL	No or invalid Project Context ASO cannot work correct

Fehlermeldung	Wert	
CCF_STORAGE_DBASO_E_ENUM_NO-ENTRIES	0x80046210L	no entries found for enumeration
CCF_STORAGE_DBASO_E_NOT_FOUND	0x80046211L	specified object can not be found
CCF_STORAGE_DBASO_E_ID_CHANGES_NOT_POSSIBLE	0x80046212L	It is not allowed to change IDs.
CCF_STORAGE_DBASO_E_PARAMETER_TOO_LONG	0x80046213L	The parameter is too long.
CCF_STORAGE_DBASO_E_DATA_TOO_OLD	0x80046214L	Can't read data. Data from older WinCC version.
CCF_STORAGE_DBASO_E_DATA_NEWER_THAN_SOFTWARE	0x80046215L	Data newer than software. Please grade up.
CCF_STORAGE_DBASO_E_ENUM-SINK_RETURNED_WITH_ERROR	0x80046216L	Enum sink returned with error.
CCF_STORAGE_DBASO_E_DATA_EXIST_IN_REFERENCED_TABLE	0x80046217L	Primary key is referenced in another table.
CCF_STORAGE_DBASO_E_DATA_OUT_OF_RANGE	0x80046218L	Value out of Range.
CCF_STORAGE_DM_ASO_BULK_FAILED	0x80046219L	Bulk operation failed.
CCF_STORAGE_DBASO_E_RT_LIMIT_WARNING	0x8004621AL	The runtime limit is reached
CCF_STORAGE_DBASO_E_BAD_LOCAL_ID	0x8004621BL	
CCF_STORAGE_FILEASO_E_NO_PICTURE	0x80046301L	no picture available
CCF_STORAGE_GDO_BADTYPE_IN_VARIANT	0x80046405L	wrong type in vt
CCF_STORAGE_GDO_DATAINCONSISTENT	0x80046406L	data inconsistent some part of the data is not valid or out of range
CCF_STORAGE_GDO_UNITID_INVALID	0x80046407L	data inconsistent channel unit ID is invalid
CCF_STORAGE_GDO_USERTYPE_INVALID	0x80046408L	data inconsistent usertype is invalid
CCF_STORAGE_GDO_ADDRESSBUILD_FAILURE	0x80046409L	data inconsistent failure in building address for variable
CCF_STORAGE_GDO_MODIFY_NOCHANGE	0x8004640AL	data inconsistent no relevant modifications are made
CCF_STORAGE_GDO_CHANNELWRAPPER_LOAD	0x8004640BL	CCMfcChannelWrapper.DLL can not be found
CCF_STORAGE_GDO_CHANNEL_LOAD	0x8004640CL	Channel Load failure Channel can not be found
CCF_STORAGE_GDO_CHANNEL_FUNC	0x8004640DL	Channel Entry Points not found
CCF_STORAGE_GDO_CHANNEL_UNITNR	0x8004640EL	Invalid Unit Number

Fehlermeldung	Wert	
CCF_STORAGE_GDO_CONNECTION	0x8004640FL	Invalid Connection or Connection not found
CCF_STORAGE_GDO_TAGGROUP	0x80046410L	Invalid TagGroup or TagGroup not found
CCF_STORAGE_GDO_NAMENOTFOUND	0x80046411L	Specified name not found
CCF_STORAGE_GDO_TAGOFUSERTYPEEXISTS	0x80046412L	The object UserType could not be removed
CCF_STORAGE_GDO_PROJECTISACTIVE	0x80046413L	The project is activated. No changes or removes allowed.
CCF_STORAGE_GDO_UNIT	0x80046414L	Invalid channel unit or channel unit not found
CCF_STORAGE_GDO_MINMAX_LIMIT	0x80046415L	Min Max Limit violation
CCF_STORAGE_GDO_PROJECT_IN_RUNTIME	0x80046416L	This Operation is not allowed while project is in runtime.
CCF_STORAGE_GDO_BAD_POINTER	0x80046600L	NULL pointer for get property not allowed.
CCF_STORAGE_GDO_UNEXPECTED	0x80046601L	Unexpected error.
CCF_STORAGE_GDO_NO_RECORD	0x80046602L	No record.
CCF_STORAGE_GDO_OPERATION_NOT_COMMITTED	0x80046603L	Current operation not committed.
CCF_STORAGE_GDO_NO_PROJECT_CONNECTOR	0x80046604L	No project connector.
CCF_STORAGE_GDO_PROJECT_NOT_OPENED	0x80046605L	Project no opened.
CCF_STORAGE_GDO_GET_ASO_FAILED	0x80046606L	Get Aso failed.
CCF_STORAGE_GDO_BAD_TYPEID	0x80046607L	Bad Type ID.
CCF_STORAGE_GDO_OTHER_TYPE_NOT_COMMITTED	0x80046608L	Other type not committed.
CCF_STORAGE_GDO_CACHE_READ	0x80046609L	Error in cache read.
CCF_STORAGE_GDO_CACHE_WRITE	0x8004660AL	Error in cache write.
CCF_STORAGE_GDO_DM_GDO_ERROR	0x8004660BL	Error in DM Gdo.
CCF_STORAGE_GDO_TEXT_GDO_ERROR	0x8004660CL	Error in Text Gdo.
CCF_STORAGE_GDO_ENUMERATOR_CREATION	0x8004660DL	Creation of an ASO enumerator failed.
CCF_STORAGE_GDO_ENUMDATA_FAILED	0x8004660EL	Call to ICCTxtAsoEnum::EnumData() failed.
CCF_STORAGE_GDO_ENUM_SINK_CREATION	0x8004660FL	Creation of an enumeration sink object failed.
CCF_STORAGE_GDO_WRONG_INTERFACE	0x80046610L	Pointer to wrong interface passed.
CCF_STORAGE_GDO_BAD_LOCALE_ID	0x80046611L	Locale ID not found in map.
CCF_STORAGE_GDO_CREATE_OBJECT	0x80046612L	CreateObject failed.
CCF_STORAGE_GDO_MODIFY_OBJECT	0x80046613L	ModifyObject failed.
CCF_STORAGE_GDO_READ_OBJECT	0x80046614L	ReadObject failed.

Fehlermeldung	Wert	
CCF_STORAGE_GDO_NOT_IMPLEMENTED	0x80046615L	Not implemented.
CCF_STORAGE_GDO_REMOVE_OBJECT	0x80046616L	RemoveObject failed.
CCF_STORAGE_GDO_REMOVE_ALL	0x80046617L	RemoveAll failed.
CCF_STORAGE_GDO_ENUM_OBJECTS	0x80046618L	EnumObjects failed.
CCF_STORAGE_GDO_GET_OBJECT_COUNT	0x80046619L	GetObjectCount failed.
CCF_STORAGE_GDO_GET_RW_INTERFACE_FROM_ASO	0x8004661AL	GetRWInterface from Aso failed.
CCF_STORAGE_GDO_ON_ENUM_DATA	0x8004661BL	OnEnumData of sink object failed.
CCF_STORAGE_GDO_ASO_CREATE	0x8004661CL	Create call to Aso failed.
CCF_STORAGE_GDO_CREATE_CACHED_OBJECT	0x8004661DL	CreateCachedObject failed.
CCF_STORAGE_GDO_CREATE_SINK_OBJECT_FOR_ASO	0x8004661EL	Create Sink Object for Aso failed.
CCF_STORAGE_GDO_CREATE_DM_GDO	0x8004661FL	Error in create of DM Gdo.
CCF_STORAGE_GDO_CONNECT_DM_GDO	0x80046620L	Error in connect to DM Gdo.
CCF_STORAGE_GDO_CREATE_TEXT_GDO	0x80046621L	Error in create Text Gdo.
CCF_STORAGE_GDO_CONNECT_TEXT_GDO	0x80046622L	Error in connect to Text Gdo.
CCF_STORAGE_GDO_INVALID_ENUM_SINK_POINTER	0x80046623L	Invalid enum sink pointer.
CCF_STORAGE_GDO_VARIANT_CHANGE_TYPE	0x80046624L	Error in Variant ChangeType.
CCF_STORAGE_GDO_GET_COLLECT_ID	0x80046625L	Error in GetCollectID.
CCF_STORAGE_GDO_COMMIT_TO_TYPE_HANDLER	0x80046626L	CommitToTypeHandler failed.
CCF_STORAGE_GDO_SET_TAG_ID	0x80046627L	Error in SetTagId.
CCF_STORAGE_GDO_SET_TEXT_ID	0x80046628L	Error in SetTextId.
CCF_STORAGE_GDO_COMMIT_CACHED_OBJECT	0x80046629L	Error in CommitCachedObject.
CCF_STORAGE_GDO_QUERY_INTERFACE_OF_SINK	0x8004662AL	QueryInterface of sink failed.
CCF_STORAGE_GDO_BAD_PROPERTY_PARAMETER	0x8004662BL	Bad Parameter for Property provided.
CCF_STORAGE_GET_TAGNAME_FROM_ID	0x8004662CL	Cannot get tag-name from tag-id.
CCF_STORAGE_DELETE_RECORD_OF_CREATORID	0x8004662DL	Cannot delete records for CreatorID not equal 0.
CCF_STORAGE_CACHE_OVERFLOW	0x8004662EL	Cache overflow.
CCF_STORAGE_SERVER_DOWN	0x8004662FL	Server down.

2.12 Fehlerbehandlung (RT Professional)

Fehlermeldung	Wert	
CCF_STORAGE_ASO_PTR_NOT_INITIALIZED	0x80046630L	Aso smart pointer not initialized.
CCF_STORAGE_CREATE_CONNECTION_CONTROLLER	0x80046631L	Cannot create connection controller.
CCF_STORAGE_CONNECTION_CONTROLLER_CONNECT_TO_PROJECT	0x80046632L	Cannot connection controller to project.
CCF_STORAGE_ASO_WRAPPER_SET_CONNECTION_CONTROLLER	0x80046633L	Cannot set connection controller for aso wrapper.
CCF_STORAGE_ASO_WRAPPER_SET_PROJECT_STORAGE	0x80046634L	Cannot set project storage for aso wrapper.
CCF_STORAGE_PROGID_FROM_CLSID	0x80046635L	Error in ProgIDFromCLSID.
CCF_STORAGE_ASO_WRAPPER_GET_ASO	0x80046636L	Error in Aso wrapper, GetAso call.
CCF_STORAGE_ASO_WRAPPER_SET_PROJECT_STORAGE_NULL	0x80046637L	Error in Aso wrapper, SetProjectStorage(NULL).
CCF_STORAGE_ASO_WRAPPER_GET_RWINTERFACE	0x80046638L	Error in Aso wrapper, GetRWInterface call.
CCF_STORAGE_GDO_GET_PROJECT	0x80046639L	Error in GdoCoreBase, GetProject call.
CCF_STORAGE_TLGGDO_VAR_NO_TRIGGER	0x8004663AL	Error in GdoCoreBase, GetProject call.
CCF_STORAGE_PACKAGES_PROJECT_ASO	0x80046501L	can't instantiate Project-ASO
CCF_STORAGE_PACKAGES_EXPORT_DATA	0x80046502L	Export Data failed
CCF_STORAGE_PACKAGES_LIST	0x80046503L	Enum Packages failed
CCF_STORAGE_CCPACKAGEMGR_EXPORT_IS_RUNNING	0x80046504L	Export Data is running
CCF_STORAGE_CCPACKAGEMGR_UPDATE_IS_RUNNING	0x80046505L	Update Data is running
CCF_STORAGE_PACKAGES_EXPORT_USER_CANCELED	0x80046506L	Export was canceled by user
CCF_STORAGE_PACKAGES_IMPORT_ONLY_ONE_CAS_CAN_BE_IMPORTED	0x80046507L	Only one central archive server can be imported.
CCF_STORAGE_PACKAGES_FORBID_STANDARD_SERVER	0x80046508L	Configuration of standard-server is not allowed for this server-type
CCF_STORAGE_PACKAGES_IMPORT_IMPORT_RED	0x80046509L	Do not import redundand server.
CCF_STORAGE_PACKAGES_IMPORT_CHANGE_RED	0x8004650AL	Do not update package with redundand server.
CCF_STORAGE_PACKAGES_IMPORT_IMPORT_PHYS	0x8004650BL	Do not import package with identical server.
CCF_STORAGE_PACKAGES_IMPORT_CHANGE_PHYS	0x8004650CL	Do not update package with identical server.
CCF_STORAGE_PACKAGES_IMPORT_CHANGE_OWN	0x8004650DL	Do not import or update package with own server.

Fehlermeldung	Wert	
CCF_STORAGE_PACKAGES_IMPORT_INCLUDE_OWN_RED	0x8004650EL	Do not import or update package with own redundand server.
PCC_E_CANNOT_CONNECT_TO_DATABASE	0x80047000L	Cannot connect to database
PCC_E_RECONNECT_TO_DATABASE_FAILED	0x80047001L	Reconnect to database failed
PCC_E_JOB_ID_IS_INVALID	0x80047002L	Job ID is invalid
PCC_E_INVALID_CONTEXT_INFO_PROVIDED	0x80047003L	Invalid context info provided
PCC_E_DATABASE_ERROR_SEE_EVENT_LOG	0x80047004L	Database error - see event log for further information
PCC_E_ERROR_SEE_EVENT_LOG	0x80047005L	Unspecified error - see event log for further information
PCC_E_INVALID_PARAMETERS	0x80047006L	Incorrect parameters provided
PCC_E_ROLLBACK_TRANSACTION_FAILED	0x80047007L	Cannot rollback transaction
PCC_E_COMMIT_TRANSACTION_FAILED	0x80047008L	Cannot commit transaction
PCC_E_BEGIN_TRANSACTION_FAILED	0x80047009L	Cannot start transaction
PCC_E_COPY_WORLDMAP_FAILED_WORLDMAP_LOCKED	0x8004700AL	Failed to copy worldmap - worldmap locked
PCC_E_MODIFY_WORLDMAP_FAILED_WORLDMAP_LOCKED	0x8004700BL	Failed to modify worldmap - worldmap locked
PCC_E_REMOVE_WORLDMAP_FAILED_WORLDMAP_LOCKED	0x8004700CL	Failed to remove worldmap - worldmap locked
PCC_E_MODIFY_VIEW_FAILED_VIEW_LOCKED	0x8004700DL	Failed to modify view - view locked
PCC_E_REMOVE_VIEW_FAILED_VIEW_LOCKED	0x8004700EL	Failed to remove view - view locked
PCC_E_COPY_WORLDMAP_FAILED_VIEWS_LOCKED	0x8004700FL	Copy worldmap failed - view(s) locked
PCC_E_MODIFY_WORLDMAP_FAILED_VIEWS_LOCKED	0x80047010L	Modify worldmap failed - view(s) locked
PCC_E_REMOVE_WORLDMAP_FAILED_VIEWS_LOCKED	0x80047011L	Remove worldmap failed - view(s) locked
PCC_E_COPY_WORLDMAP_FAILED_OBJECTS_LOCKED	0x80047012L	Copy worldmap failed - object(s) locked
PCC_E_MODIFY_WORLDMAP_FAILED_OBJECTS_LOCKED	0x80047013L	Modify worldmap failed - object(s) locked
PCC_E_REMOVE_WORLDMAP_FAILED_OBJECTS_LOCKED	0x80047014L	Remove worldmap failed - object(s) locked
PCC_E_MODIFY_OBJECTS_FAILED_OBJECTS_LOCKED	0x80047015L	Failed to modify object(s) - object(s) locked
PCC_E_REMOVE_OBJECTS_FAILED_OBJECTS_LOCKED	0x80047016L	Failed to remove object(s) - object(s) locked
PCC_E_COPY_WORLDMAP_FAILED_NO_ACTIVATED_WORLDMAP	0x80047017L	Failed to copy worldmap in job - no activated worldmap

2.12 Fehlerbehandlung (RT Professional)

Fehlermeldung	Wert	
PCC_E_REMOVE_WORLDMAP_FAILED_NO_ACTIVATED_WORLDMAP	0x80047018L	Failed to remove worldmap in job - no activated worldmap
PCC_E_REMOVE_OBJECTS_FAILED_NO_ACTIVATED_OBJECTS	0x80047019L	Failed to remove object(s) in job - no activated object(s)
PCC_E_MODIFY_OBJECTS_FAILED_NO_ACTIVATED_OBJECTS	0x8004701AL	Failed to modify object(s) in job - no activated object(s)
PCC_E_REMOVE_VIEW_FAILED_NO_ACTIVATED_VIEW	0x8004701BL	Failed to remove view in job - no activated view
PCC_E_MODIFY_VIEW_FAILED_NO_ACTIVATED_VIEW	0x8004701CL	Failed to modify view in job - no activated view
PCC_E_MODIFY_WORLDMAP_FAILED_NO_ACTIVATED_JOB	0x8004701DL	Failed to modify worldmap in job - no activated job
PCC_E_COPY_WORLDMAP_FAILED_NOT_FOUND	0x8004701EL	Copy worldmap failed - worldmap not found
PCC_E_WORLDMAP_NAME_IN_USE	0x8004701FL	Worldmap name already in use
PCC_E_VERSION_NUMBER_INVALID	0x80047020L	Version number invalid
PCC_E_USER_DOES_NOT_HAVE_WRITE_PERMISSION	0x80047021L	User does not have 'write' permission
PCC_E_CREATE_AREALOCK_FAILED_OBJECTS_LOCKED	0x80047022L	Failed to create area lock. There are locked objects
PCC_E_MODIFY_AREALOCK_FAILED_OBJECTS_LOCKED	0x80047023L	Failed to modify area lock. There are locked objects
PCC_E_JOB_NOT_FOUND	0x80047024L	The job was not found
PCC_E_JOB_MUST_EXIST_MORE_THAN_ONCE	0x80047025L	The job must exist more than once
PCC_E_CANNOT_MODIFY_ROOT_CTX	0x80047026L	The root context can not be modified
PCC_E_CANNOT_MODIFY_JOB	0x80047027L	The job can not be modified
PCC_E_CANNOT_MODIFY_JOB_RESERVED_BY_OTHER_USER	0x80047028L	The job can not be modified. It is reserved by another user
WINCC_DBPLUGIN_E_ROW_EXISTS	0x80047100L	Row already exists
WINCC_DBPLUGIN_E_CONSTRAINT_CONFLICT	0x80047101L	A constraint conflict occurred
WINCC_DBPLUGIN_E_INVALID_OBJECT_NAME	0x80047102L	Invalid object name
WINCC_DBPLUGIN_E_INVALID_COLUMN_NAME	0x80047103L	Invalid column name
WINCC_DBPLUGIN_E_DATA_CONVERSION	0x80047104L	Error converting data
WINCC_DBPLUGIN_E_DATA_CONVERSION_DATETIME	0x80047105L	String conversion to Datetime-format failed
WINCC_DBPLUGIN_E_STRING_TOO_LONG	0x80047106L	String will be truncated
WINCC_DBPLUGIN_E_ARITHMETIC_OVERFLOW	0x80047107L	Arithmetic overflow data type cannot store value
WINCC_DBPLUGIN_RECORD_LOCKED	0x80047108L	Record locked
WINCC_DBPLUGIN_TABLE_LOCKED	0x80047109L	Table locked

Fehlermeldung	Wert	
WINCC_DBPLUGIN_CANNOTOPENDB	0x8004710AL	Can not open database
PCC_E_CANNOT_DO_MAKE_UNIQUE_NAME	0x8004710BL	Object cannot get a unique name
PCC_E_NO_DB_FOUND_ON_RTS	0x8004710CL	Could not access database on RTS master or standby
PCC_E_OPERATION_NOT_ALLOWED_ON_RTS	0x8004710DL	This operation is not allowed on the RTS database. It must be performed in the engineering database on the PSOS
PCC_E_NO_DB_FOUND_ON_PSOS	0x8004710EL	Could not access database on PSOS

VB-Scripting (Panels, Comfort Panels, RT Advanced, RT Professional)

3

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

3.1.1 AcknowledgeAlarm (Panels, Comfort Panels, RT Advanced)

Beschreibung

Quittiert alle ausgewählten Meldungen.

Verwenden Sie diese Systemfunktion, wenn das Bediengerät keine ACK-Taste besitzt oder die integrierte Taste der Meldeanzeige nicht genutzt werden soll.

Diese Systemfunktion kann nur für Funktionstasten verwendet werden.

Verwendung in der Funktionsliste

QuittiereMeldung

Verwendung in benutzerdefinierten Funktionen

AcknowledgeAlarm

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.2 ActivatePreviousScreen (Panels, Comfort Panels, RT Advanced)

Beschreibung

Führt einen Bildwechsel zu dem Bild aus, das vor dem aktuellen Bild aktiviert war. Der Bildwechsel wird nicht ausgeführt, wenn zuvor kein Bild aktiviert war.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Die 10 zuletzt aufgerufenen Bilder werden gespeichert. Wenn Sie zu einem Bild wechseln, das nicht mehr gespeichert ist, wird eine Systemmeldung ausgegeben.

Hinweis

Wenn Sie die Systemfunktion verwenden wollen, muss das Bild zu dem sie wechseln in der Navigationsstruktur verwendet werden.

Verwendung in der Funktionsliste

AktiviereVorherigesBild

Verwendung in benutzerdefinierten Funktionen

ActivatePreviousScreen

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.3 ActivateScreen (Panels, Comfort Panels, RT Advanced)

Beschreibung

Führt einen Bildwechsel zu dem angegebenen Bild aus.

Um vom Grundbild in das Permanentfenster oder umgekehrt zu wechseln, verwenden Sie die Systemfunktion "AktiviereBildMitNummer".

Verwendung in der Funktionsliste

AktiviereBild (Bildname, Objektnummer)

Verwendung in benutzerdefinierten Funktionen

ActivateScreen (Screen_name, Object_number)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bildname

Name des Bildes, zu dem gewechselt wird.

Objektnummer

Das Bedienelement, das nach dem Bildwechsel im angegebenen Bild den Fokus erhält. Die Nummer des Bedienelements legen Sie während der Projektierung durch die Tab-Reihenfolge fest.

Wenn Sie "0" angeben:

- Ist der Fokus beim Aufruf der Systemfunktion im Permanentfenster, behält das Permanentfenster den Fokus.
- Ist der Fokus beim Aufruf der Systemfunktion im Grundbild, erhält das erste Bedienelement im angegebenen Bild den Fokus.

Hinweis

Wenn die Systemfunktion "AktiviereBild" an das Ereignis "Rand erreicht" projiziert wird, so ist für den Parameter "Objektnummer" nur der Wert 0 zulässig. Das aktive Objekt wird nicht anhand der Objektnummer festgelegt, sondern anhand der X-Position des aktiven Objekts vor dem Bildwechsel.

Beispiel

Der folgende Programmcode aktiviert mit der Funktion ActivateScreen beim Klicken auf eine Taste das Bild "Screen_2".

```
{  
  
// User defined code  
// i.e. when pressing a button  
ActivateScreen ("Screen_2", 0);  
...  
}
```

3.1.4 ActivateScreenByNumber (Panels, Comfort Panels, RT Advanced)

Beschreibung

Führt abhängig von einem Variablenwert einen Bildwechsel zu einem Bild aus.

Das Bild wird über seine Bildnummer identifiziert.

Verwendung in der Funktionsliste

AktiviereBildMitNummer (Bildnummer, Objektnummer)

Verwendung in benutzerdefinierten Funktionen

ActivateScreenByNumber (Screen_number, Object_number)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bildnummer

Variable, welche die Bildnummer enthält, zu dem gewechselt wird.

Wenn Sie vom Grundbild ins Permanentfenster oder umgekehrt wechseln wollen, geben Sie "0" oder "-1" an:

0 = Wechselt vom Grundbild ins Permanentfenster.

-1 = Wechselt vom Permanentfenster ins Grundbild

Objektnummer

Nummer des Bildobjekts, das nach dem Bildwechsel im angegebenen Bild den Fokus erhält. Die Nummer des Bedienelements legen Sie während der Projektierung durch die Tab-Reihenfolge fest.

Wenn Sie "0" angeben:

- Ist der Fokus beim Aufruf der Systemfunktion im Permanentfenster, behält das Permanentfenster den Fokus.
- Ist der Fokus beim Aufruf der Systemfunktion im Grundbild, erhält das erste Bedienelement im angegebenen Bild den Fokus.

3.1.5 ActivateSystemDiagnosticsView (Panels, Comfort Panels, RT Advanced)

Beschreibung

Aktiviert die System-Diagnoseanzeige. Die System-Diagnoseanzeige zeigt die Detailansicht des betroffenen Geräts.

Verwendung in der Funktionsliste

AktiviereSystemdiagnoseAnzeige(Bildname, Objektname)

Verwendung in benutzerdefinierten Funktionen

ActivateSystemDiagnosticsView (Screen_name, Object_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bildname

Name des Bildes, die die System-Diagnoseanzeige enthält.

Objektnamen

Objektnamen der System-Diagnoseanzeige.

3.1.6 ArchiveLogFile (Panels, Comfort Panels, RT Advanced)

Beschreibung

Diese Systemfunktion verschiebt oder kopiert ein Archiv zur Langzeitarchivierung an einen anderen Speicherort.

Verwenden Sie bei Audit Trails immer den Modus "Verschieben (hmiMove)", da sonst durch doppelte Datenhaltung gegen die FDA-Richtlinie verstoßen wird.

Führen Sie vor "ArchiviereProtokolldatei" immer die Systemfunktion "SchlieÙeAlleArchive" aus.

Führen Sie nach Abschluss dieser Systemfunktion die Systemfunktion "ÖffneAlleArchive" aus.

Im Modus " Archiv kopieren" werden die Archive erst wieder geöffnet, nachdem das Archiv erfolgreich kopiert wurde, oder wenn beim Kopiervorgang eine Zeitüberschreitung eintritt. Im Modus "Archiv verschieben" werden die zu verschiebenden Archive umbenannt und die neuen Archive werden sofort geöffnet. Für das Verschieben der umbenannten Archive wird ein Auftrag abgesetzt, der alle 300 Sekunden versucht diese zu verschieben, falls der Server nicht erreichbar ist. Der Auftrag bleibt auch nach einem Neustart von Runtime bestehen, bis er ausgeführt ist.

Verwendung in der Funktionsliste

ArchiviereProtokolldatei (Archivtyp, Archiv, Verzeichnisname, Modus)

Verwendung in benutzerdefinierten Funktionen

ArchiveLogFile (Log_type, Log, Directory_name, Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Archivtyp

Legt den Typ des Archivs fest:

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

0 (hmiTagArchive) = Variablenarchiv

1 (miAlarmArchive) = Meldearchiv

2 (hmiAudittrailArchive)= Audit-Trail-Archiv Verfügbar bei GMP-konformen Projekten. Nähere Hinweise hierzu finden Sie unter "GMP-konforme Projektierung aktivieren".

Archiv

Name des Archivs, das archiviert wird.

Verzeichnisname

Pfad, in den das Archiv gespeichert wird.

Modus

0 (hmiCopy) = Archiv kopieren

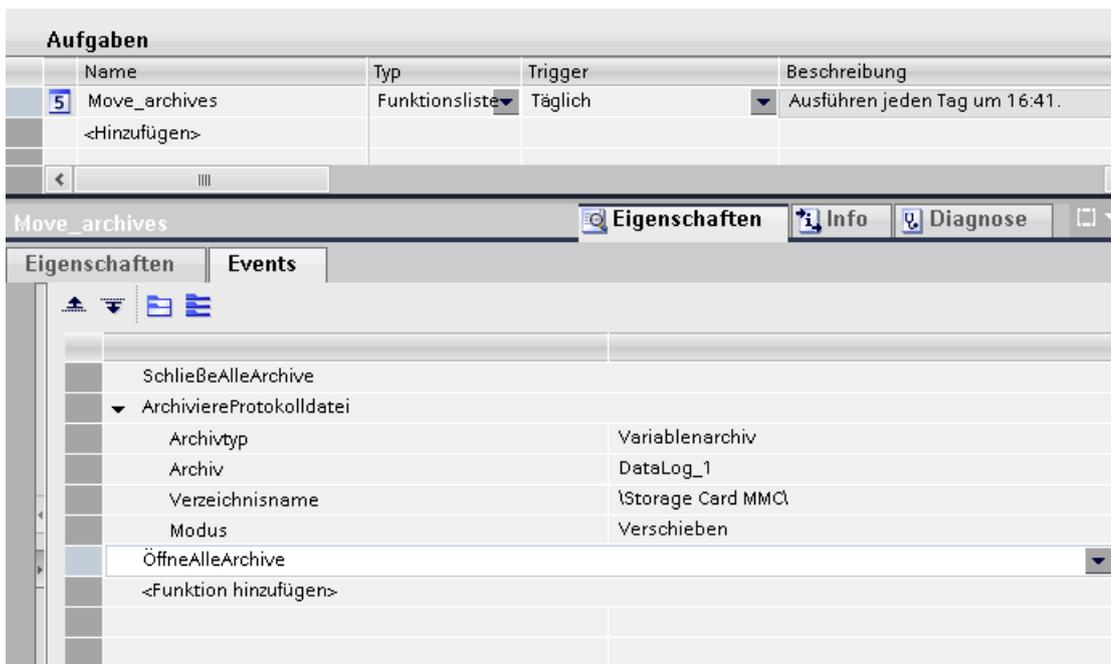
1 (hmiMove) = Archiv verschieben

Anwendungsbeispiel

Sie wollen ein Archiv vom lokalen Speichermedium auf den Server verschieben, um das Archiv in regelmäßigen Abständen zu sichern.

Hinweise zur Projektierung

Legen Sie im Aufgabenplaner eine Aufgabe an, die täglich zu einer bestimmten Uhrzeit ausgeführt wird. Projektieren Sie an der Aufgabe folgende Funktionsliste:



Ablauf am Bediengerät

- Alle geöffneten Archive werden geschlossen.
- Das angegebene Archiv wird auf den Server verschoben.
- Alle geschlossenen Archive werden wieder geöffnet.

3.1.7 BackupRAMFileSystem (Panels, Comfort Panels, RT Advanced)

Beschreibung

Sichert das RAM-Dateisystem auf dem Speichermedium des Bediengeräts.

Nach dem Neustart des Bediengeräts werden die Daten automatisch wieder in das RAM-Dateisystem geladen.

Anwendungen wie der Internet Explorer speichern Daten (z.B. zuletzt aufgerufene Webseiten) temporär im DRAM-Dateisystem des Bediengeräts.

Verwendung in der Funktionsliste

SichereRAMDateisystem

Verwendung in benutzerdefinierten Funktionen

BackupRAMFileSystem

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.8 CalibrateTouchScreen (Panels, Comfort Panels, RT Advanced)

Beschreibung

Ruft ein Programm zur Kalibrierung des Touch-Displays auf.

Während des Kalibriervorgangs werden Sie aufgefordert, fünf Positionen auf dem Touch-Display zu berühren. Um den Kalibriervorgang zu bestätigen, berühren Sie innerhalb von 30 Sekunden das Touch-Display. Wenn Sie das Touch-Display innerhalb der Zeitspanne nicht berühren, werden die Kalibriereinstellungen verworfen. Die Bedienerführung ist in englischer Sprache.

Verwenden Sie diese Systemfunktion, wenn Sie das Bediengerät zum ersten Mal in Betrieb nehmen.

Verwendung in der Funktionsliste

KalibriereTouchscreen

Verwendung in benutzerdefinierten Funktionen

CalibrateTouchScreen

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

Hinweis

Die Systemfunktion KalibriereTouchscreen kann nicht simuliert werden.

3.1.9 ChangeConnection (Panels, Comfort Panels, RT Advanced)

Beschreibung

Trennt die Verbindung zur gerade verwendeten Steuerung und stellt eine Verbindung zu einer Steuerung mit einer anderen Adresse her. Die neu verbundene Steuerung muss dabei zur gleichen Geräteklasse (S7-1200, S7-300, ...usw) gehören. Bei S7-1200 ist die Verwendung der Funktion darüber hinaus nur bei absoluter Adressierung zulässig.

Hinweis

Achten Sie beim Wechsel zu einer anderen Adresse darauf, dass diese Adresse nicht bereits von einem anderen Bediengerät verwendet wird.

Folgende Adresstypen werden unterstützt:

- IP-Adresse
- MPI-Adresse

Folgende Steuerungstypen werden unterstützt:

- SIMATIC S7 300/400
- SIMATIC S7 NC
- SIMOTION

Verwendung in der Funktionsliste

WechseleVerbindung (Verbindung, Adresse, Steckplatz, Baugruppenträger)

Verwendung in benutzerdefinierten Funktionen

ChangeConnection (Connection, Address, Slot, Rack)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Verbindung

Name der Verbindung, die getrennt wird. Den Namen legen Sie während der Projektierung z.B. im Editor "Verbindungen" fest.

Adresse

MPI/PROFIBUS- oder IP-Adresse der Steuerung, zu der die Verbindung hergestellt wird.

Hinweis

Sie legen die Adresse über eine Variable fest. Die Objektliste zeigt Ihnen Variablen aller Datentypen an. Wählen Sie ausschließlich Variablen folgender Datentypen:

- Ethernet-Verbindung: Datentyp "String"
 - MPI-Verbindung: Datentypen "Int"
-

Steckplatz

Steckplatz der Steuerung, zu der die Verbindung hergestellt wird.

Baugruppenträger

Baugruppenträger der Steuerung, zu der die Verbindung hergestellt wird.

Anwendungsbeispiel

Sie wollen ein Bediengerät an verschiedenen Maschinen betreiben. Dazu projektieren Sie im Projekt die benötigten Steuerungen, zu denen Sie mit Tastendruck wechseln wollen. Beim Wechsel der Steuerung wird die Verbindung mit der gerade verwendeten Steuerung getrennt. Danach wird die Verbindung zur neuen Steuerung mit anderen Adressparametern wieder aufgebaut. Um auf die Werte der neuen Steuerung zuzugreifen, projektieren Sie für die verwendete Steuerung dieselben Variablen.

Die Steuerung, die Sie beim Anlegen des Projekts angegeben haben, wird standardmäßig verwendet.

1. Geben Sie im Editor "Verbindungen" Namen und Adresse der Steuerung ein.
2. Projektieren Sie im Prozessbild eine Schaltfläche.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

3. Projektieren Sie am Ereignis "Drücken" die Systemfunktion "WechseleVerbindung".
4. Geben Sie als Parameter den Namen der Verbindung und die Adresse der Steuerung an.

3.1.10 ClearAlarmBuffer (Panels, Comfort Panels, RT Advanced)

Beschreibung

Löscht Meldungen aus dem Meldepuffer am Bediengerät.

Hinweis

Noch nicht quittierte Meldungen werden ebenfalls gelöscht.

Verwendung in der Funktionsliste

LöscheMeldepuffer (Nummer der Meldeklasse)

Verwendung in benutzerdefinierten Funktionen

ClearAlarmBuffer (Alarm_class_number)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Nummer der Meldeklasse

Legt fest, welche Meldungen aus dem Meldepuffer gelöscht werden:

0 (hmiAll) = Alle Meldungen

1 (hmiAlarms) = Meldungen der Meldeklasse "Errors"

2 (hmiEvents) = Meldungen der Meldeklasse "Warnings"

3 (hmiSystem) = Meldungen der Meldeklasse "System"

4 (hmiS7Diagnosis) = Meldungen der Meldeklasse "Diagnosis Events"

Hinweis

Geräteabhängigkeit

Meldungen der Meldeklasse "Diagnosis Events" stehen auf Basic Panels nicht zur Verfügung.

3.1.11 ClearAlarmBufferProtool (Panels, Comfort Panels, RT Advanced)

Beschreibung

Diese Systemfunktion existiert aus Kompatibilitätsgründen.

Sie hat die gleiche Funktionalität wie die Systemfunktion "LöscheMeldepuffer", verwendet aber die alte ProTool-Nummerierung.

Verwendung in der Funktionsliste

LöscheMeldepufferProTool (Nummer der Meldeklasse)

Verwendung in benutzerdefinierten Funktionen

ClearAlarmBufferProtoolLegacy (Alarm_class_number)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Nummer der Meldeklasse

Nummer der Meldeklasse, deren Meldungen gelöscht werden sollen:

-1 (hmiAllProtoolLegacy) = Alle Meldungen

0 (hmiAlarmsProtoolLegacy) = Meldungen der Meldeklasse "Errors"

1 (hmiEventsProtoolLegacy) = Meldungen der Meldeklasse "Warnings"

2 (hmiSystemProtoolLegacy) = Meldungen der Meldeklasse "System"

3 (hmiS7DiagnosisProtoolLegacy) = Meldungen der Meldeklasse "Diagnosis Events"

Hinweis

Geräteabhängigkeit

Meldungen der Meldeklasse "Diagnosis Events" stehen auf Basic Panels nicht zur Verfügung.

3.1.12 ClearDataRecord (Panels, Comfort Panels, RT Advanced)

Beschreibung

Löscht einen Rezepturdatensatz.

Mehrere Datensätze können in einer oder mehreren Rezepturen gelöscht werden.

Verwendung in der Funktionsliste

LöscheDatensatz (Rezepturnummer/-name, Datensatznummer/-name, Bestätigung, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

DeleteDataRecord

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, aus der Rezepturdatensätze gelöscht werden. Wenn Sie aus allen verfügbaren Rezepturen Rezepturdatensätze löschen wollen, geben Sie "0" an.

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der gelöscht wird. Wenn Sie alle Rezepturdatensätze löschen wollen, geben Sie "0" an.

Bestätigung

Legt fest, ob der Bediener das Löschen bestätigen soll:

0 (hmiOff) = Aus: Löschvorgang wird ohne Bestätigung gestartet.

1 (hmiOn) = Ein: Beginn des Löschvorgangs muss bestätigt werden.

Statusmeldung ausgeben

Legt fest, ob nach dem Löschen eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.13 ClearDataRecordMemory (Panels, Comfort Panels, RT Advanced)

Beschreibung

Löscht alle Rezepturdatensätze vom angegebenen Speichermedium.

Verwendung in der Funktionsliste

LöscheDatensatzSpeicher (Speicherort, Bestätigung, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

ClearDataRecordMemory (Storage_location, Confirmation, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Speicherort

Legt den Speicherort fest:

0 (hmiFlashMemory) = Flash-Speicher: Interner Flash-Speicher des Bediengeräts

1 (hmiStorageCard) = Storage Card

2 (hmiStorageCard2) = Storage Card 2

3 (hmiStorageCard3) = Storage Card MultiMediaCard

4 (hmiStorageCard4) = Storage Card USB

Bestätigung

Legt fest, ob der Bediener das Löschen bestätigen soll:

0 (hmiOff) = Aus: Löschvorgang wird ohne Bestätigung gestartet.

1 (hmiOn) = Ein: Beginn des Löschvorgangs muss bestätigt werden.

Statusmeldung ausgeben

Legt fest, ob nach dem Löschen eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.14 ClearLog (Panels, Comfort Panels, RT Advanced)

Beschreibung

Löscht alle Datensätze in dem angegebenen Archiv.

Verwendung in der Funktionsliste

LoescheArchiv (Archivtyp, Archiv)

Verwendung in benutzerdefinierten Funktionen

ClearLog (Log_type, Log)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Archivtyp

Legt den Typ des Archivs fest.

0 (hmiTagArchive) = Variablenarchiv

1 (hmiAlarmArchive) = Meldearchiv

2 (hmiAudittrailArchive) = Audit Trail-Archiv. Verfügbar bei GMP-konformen Projekten. Nähere Nähere Hinweise hierzu finden in "GMP-konforme Projektierung aktivieren".

Archiv

Name des Archivs, aus dem alle Einträge gelöscht werden.

3.1.15 CloseAllLogs (Panels, Comfort Panels, RT Advanced)

Beschreibung

Trennt die Verbindung zwischen WinCC und allen Archiven.

Hinweis

Bevor Sie ein Archiv schließen, muss erst die Archivierung in diesem Archiv gestoppt werden. Verwenden Sie dazu die Systemfunktion "StoppeArchivierung".

Verwendung in der Funktionsliste

SchlieÙeAlleArchive

Verwendung in benutzerdefinierten Funktionen

CloseAllLogs

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

Anwendungsbeispiel

Sie wollen in Runtime den Datenträger wechseln, auf dem die Prozesswerte archiviert werden.

Hinweise zur Projektierung

Projektieren Sie die Systemfunktionen "StoppeArchivierung" und "SchlieÙeAlleArchive" an die Schaltfläche "Close Archive".

Projektieren Sie die Systemfunktionen "ÖffneAlleArchive" und "StarteArchivierung" an die Schaltfläche "Open Archive".

Übergeben Sie als Parameter jeweils den Namen des Archivs, das gestoppt und gestartet werden soll.

Ablauf am Bediengerät

Wenn Sie die Schaltfläche "Close Archive" drücken, wird das angegebene Archiv gestoppt und die geöffneten Archive werden geschlossen. Sie können den Datenträger wechseln. Mit der Schaltfläche "Open Archive" öffnen Sie alle Archive. Die Archivierung wird in dem angegebenen Archiv fortgesetzt.

3.1.16 ControlSmartServer (Panels, Comfort Panels, RT Advanced)

Beschreibung

Startet oder stoppt den Sm@rtServer.

Verwendung in der Funktionsliste

SteuereSmartServer (Modus)

Verwendung in benutzerdefinierten Funktionen

ControlSmartServer (Mode)

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob der Sm@rtServer gestartet oder gestoppt wird.

-1 (hmiToggle) = Umschalten: Schaltet zwischen den beiden Zuständen hin und her

0 (hmiStop) = Stoppen: Sm@rtServer wird gestoppt

1 (hmiStart) = Starten: Sm@rtServer wird gestartet

3.1.17 ControlWebServer (Panels, Comfort Panels, RT Advanced)

Beschreibung

Startet oder stoppt den Webserver.

Verwendung in der Funktionsliste

SteuereWebServer (Modus)

Verwendung in benutzerdefinierten Funktionen

ControlWebServer (Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob der Webserver gestartet oder gestoppt wird.

-1 (hmiToggle) = Umschalten: Schaltet zwischen den beiden Zuständen hin und her

0 (hmiStop) = Stoppen: Webserver wird gestoppt

1 (hmiStart) = Starten: Webserver wird gestartet

3.1.18 CopyLog (Panels, Comfort Panels, RT Advanced)

Beschreibung

Kopiert den Inhalt eines Archivs in ein anderes Archiv. Dabei können Sie Variablenwerte nur in andere Variablenarchive und Meldungen nur in andere Meldearchive kopieren.

Hinweis

Wenn Sie ein Archiv mit der Systemfunktion "KopiereArchiv" kopieren, kann es vorkommen, dass in der Kopie des Archivs nationale Sonderzeichen in den archivierten Meldetexten durch Fremdapplikationen nicht lesbar sind. WinCC Runtime ist davon nicht betroffen. Die kopierten Archive können von WinCC Runtime ohne Fehler gelesen werden.

Hinweis

Beim Kopieren der Umlaufarchive werden 80 % der Archiveinträge kopiert. 20 % der Einträge werden nicht mitkopiert, weil der Speicherplatz standardmäßig für Pufferüberlauf reserviert ist.

Verwendung in der Funktionsliste

KopiereArchiv (Archivtyp, Zielarchiv, Quellarchiv, Modus, Quellarchiv löschen)

Verwendung in benutzerdefinierten Funktionen

CopyLog (Log_type, Destination log, Source_log, Mode, Delete_source_log)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Archivtyp

Legt den Typ des Archivs fest:

0 (hmiTagArchive) = Variablenarchiv

1 (hmiAlarmArchive) = Meldearchiv

Zielarchiv

Name des Archivs, in das die Einträge kopiert werden (Zielarchiv).

Quellarchiv

Name des Archivs, aus dem die Einträge kopiert werden (Quellarchiv).

Modus

Legt fest, wie im Zielarchiv mit den kopierten Einträgen verfahren wird:

0 (hmiOverwrite) = Überschreiben: Vorhandene Einträge werden überschrieben.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

2 (hmiAppend) = Anhängen: Die Einträge werden am Ende des Zielarchivs eingefügt. Wenn die projektierte Größe des Zielarchivs erreicht ist, wird das Zielarchiv wie ein Umlaufarchiv behandelt.

Quellarchiv löschen

Legt fest, ob das Quellarchiv nach dem Kopieren gelöscht wird.

0 (hmiNo) = Nein: Wird nicht gelöscht.

1 (hmiYes) = Ja: Wird gelöscht.

3.1.19 DecreaseTag (Panels, Comfort Panels, RT Advanced)

Beschreibung

Subtrahiert den angegebenen Wert vom Wert der Variablen.

$$X = X - a$$

Hinweis

Die Systemfunktion verwendet als Eingangswert und Ausgangswert dieselbe Variable. Wenn Sie diese Systemfunktion zum Umrechnen eines Wertes verwenden, müssen Sie mit einer Hilfsvariablen arbeiten. Der Hilfsvariablen weisen Sie mit der Systemfunktion "SetzeVariable" den Variablenwert zu.

Wenn Sie die Systemfunktion an Ereignisse einer Meldung projektieren und die Variable wird im aktuellen Bild nicht verwendet, ist nicht sichergestellt, dass der tatsächliche Wert der Variablen in der Steuerung verwendet wird. Dies kann durch Setzen der Erfassungsart "Zyklisch fortlaufend" verbessert werden.

Verwendung in der Funktionsliste

VerringereVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

DecreaseTag (Tag, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, von welcher der angegebene Wert subtrahiert wird.

Wert

Der Wert, der subtrahiert wird.

3.1.20 EditAlarm (Panels, Comfort Panels, RT Advanced)

Beschreibung

Löst das Ereignis "Bearbeiten" für alle gewählten Meldungen aus.

Wenn Sie die zu bearbeitenden Meldungen noch nicht quittiert haben, erfolgt mit Aufruf dieser Systemfunktion automatisch die Quittierung.

Diese Systemfunktion kann nur für Funktionstasten verwendet werden.

Verwendung in der Funktionsliste

BearbeiteMeldung

Verwendung in benutzerdefinierten Funktionen

EditAlarm

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.21 Encode (Panels, Comfort Panels, RT Advanced)

Beschreibung

Passt den Datentyp "String" einer Variablen für Übergabe an das Automatisierungssystem (AS) an. Die Variable vom WinCC-Datentyp "String" wird in den AS-Datentyp "Array of Byte" umgewandelt. Das Ergebnis wird in eine Variable geschrieben.

Verwendung in der Funktionsliste

Codieren (Byte-Array, Zeichenfolge, Codieren)

Verwendung in benutzerdefinierten Funktionen

Encode (Byte_array, String, Encoding)

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Byte-Array

Die Variable, die den umgewandelten Wert enthält.

Hinweis

Das Byte-Array muss die doppelte Größe der Stringlänge + 2 haben. Wenn der String eine Länge von 120 Zeichen hat, muss das Byte-Array 242 Arrayelemente enthalten.

Wenn die Größe nicht ausreichend ist, werden Zeichen abgeschnitten bzw. nicht konvertiert.

Zeichenfolge

Die Variable vom Datentyp "String", die umgewandelt wird.

Codieren

0 (hmiEncodeUTF16LE) - String wird in UTF16LE (Unicode 16 Little Endian) codiert.

3.1.22 EncodeEx (Panels, Comfort Panels, RT Advanced)

Beschreibung

Passt den Datentyp "String" einer Variablen für Übergabe an das Automatisierungssystem (AS) an. Die Variable vom WinCC-Datentyp "String" wird in den AS-Datentyp "Array of Byte" umgewandelt. Das Ergebnis wird in eine Variable geschrieben.

Im Gegensatz zur Systemfunktion Codieren ist der Parameter Zeilenumbruch definierbar. Mit dem Parameter Zeilenumbruch löschen Sie die Zeilenumbrüche oder ersetzen die Zeilenumbrüche durch vordefinierte Zeichen.

Verwendung in der Funktionsliste

CodierenEx (Byte-Array, Zeichenkette, Codieren, Zeilenumbruch)

Verwendung in benutzerdefinierten Funktionen

EncodeEx (Byte_array, String, Encoding, Line_break)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Byte-Array

Die Variable, die den umgewandelten Wert enthält.

Hinweis

Das Byte-Array muss die doppelte Größe der Stringlänge +2 haben. Wenn der String eine Länge von 120 Zeichen besitzt, muss das Byte-Array 242 Arrayelemente enthalten.

Wenn die Größe nicht ausreichend ist, werden Zeichen abgeschnitten bzw. nicht konvertiert.

Zeichenkette

Die Variable vom Datentyp "String", die umgewandelt wird.

Kodierung

0 (hmiEncodeUTF16LE) - String wird in UTF16LE (Unicode 16 Little Endian) codiert.

Zeilenumbruch

Alle Zeilenumbrüche werden durch vordefinierte Zeichen ersetzt oder gelöscht. Zeilenumbrüche nicht ersetzen ist als Standardwert gesetzt.

0 (Ersetzen mit "\r\n" (0x000D, 0x000A)) - Zeilenumbruch wird durch "\r\n" ersetzt.

1 (Ersetzen mit "\n" (0x000A)) - Zeilenumbruch wird durch "\n" ersetzt.

2 (Nicht ersetzen) - Zeilenumbruch wird nicht ersetzt.

3 (Entfernen der Zeilenumbrüche) - Zeilenumbrüche werden gelöscht.

3.1.23 EstablishPROFIsafe (Panels, Comfort Panels, RT Advanced)

Beschreibung

Baut die PROFIsafe-Verbindung für den fehlersicheren Betrieb zwischen einem KTP Mobile Panel und der Steuerung auf, wenn die Verbindung vorher über PROFIsafeTrennen getrennt wurde.

Hinweis

Eine Verbindung für die PROFIsafe-Kommunikation wird nur dann hergestellt, wenn Sie am Bediengerät die Betriebsart ONLINE eingestellt haben. Verwenden Sie dazu die Systemfunktion "SetzeBetriebsartGerät".

Verwendung in der Funktionsliste

PROFIsafeVerbinden

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Verwendung in benutzerdefinierten Funktionen

EstablishPROFIsafe

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.24 ExportDataRecords (Panels, Comfort Panels, RT Advanced)

Beschreibung

Exportiert einen oder alle Datensätze einer Rezeptur in eine CSV-Datei.

Pro Rezeptur wird eine Datei angelegt.

Verwendung in der Funktionsliste

ExportiereDatensätze (Rezepturnummer/-name, Datensatznummer/-name, Dateiname, Überschreiben, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

ExportDataRecords (Recipe_number/name, Data_record number/name, File_name, Overwrite, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, aus der Rezepturdatensätze exportiert werden. Geben Sie "0" an, wenn Sie aus allen verfügbaren Rezepturen Rezepturdatensätze exportieren wollen.

Dateiname

Name der CSV-Datei, in welche die Rezepturdatensätze exportiert werden. Geben Sie den Ablageort und die Dateierweiterung (*.csv) mit an, z.B. "C:\TEMP\Orange.csv", für Basic Panels "\USB_X61.1\Orange.csv".

Hinweis

Ablage der CSV-Datei

Gilt nicht für Basic Panels:

- Wenn Sie als Ablageort eine Speicherkarte verwenden, dann geben Sie den Ablageort wie folgt an: "\StorageCard\".
 - Wenn Sie nur einen Dateinamen und keinen Pfad festlegen, wird die Datei in einem Systemverzeichnis, z.B. "C:\Documents and Settings\[User]", abgelegt.
 - Wenn nur ein Pfad für den Export angegeben wird, wird der Dateiname jeweils automatisch aus dem Rezepturnamen erzeugt. Die Voraussetzung ist, dass der Ordner, z. B. "D:\Temp", angelegt ist. Wenn der Ordner "D:\Temp" nicht angelegt ist, wird der Ordnername als Präfix für den Dateinamen verwendet, Temp_Rezepturnamen.csv.
-

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der exportiert wird. Geben Sie "0" an, wenn Sie alle Rezepturdatensätze exportieren wollen.

Überschreiben

Legt fest, ob eine bereits vorhandene CSV-Datei gleichen Namens überschrieben wird:

0 (hmiOverwriteForbidden) = Nein: CSV-Datei wird nicht überschrieben. Der Exportvorgang wird nicht ausgeführt.

1 (hmiOverwriteAlways) = Ja: CSV-Datei wird ohne Nachfrage überschrieben.

2 (hmiOverwriteWithPrompting) = Nach Bestätigung: CSV-Datei wird erst nach Bestätigung überschrieben.

Statusmeldung ausgeben

Legt fest, ob nach dem Export eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

Exportformat der Rezepturdatensätze

Wenn für die Exportdatei die Dateierweiterung ".csv" gewählt wird, dann werden lediglich gültige Zeichen aus dem ANSI-Zeichensatz unterstützt. Dies gilt auch für Trennzeichen bei Dezimalzahlen und Listenelementen. Die verwendeten Trennzeichen werden in den Ländereinstellungen des Betriebssystems des exportierenden Computers festgelegt.

Gilt nicht für Basic Panels:

Für den Export kann auch das Dateiformat "Unicode Text" (".txt") gewählt werden. Dieses Dateiformat unterstützt alle in WinCC und WinCC Runtime verwendbaren Zeichen. Die verwendeten Trennzeichen werden auch hier in den Ländereinstellungen des Betriebssystems des exportierenden Computers festgelegt. Bei diesem Dateiformat wird als Trennzeichen für Listenelemente immer der Tabulator verwendet.

Die zugehörige Funktion zum Import der Daten unterstützt ebenfalls die beiden Dateiformate ".csv" und ".txt" (Unicode).

Anwendungsbeispiel

Sie wollen über eine Taste alle Datensätze exportieren.

Hinweise zur Projektierung

Projektieren Sie die Systemfunktion "ExportiereDatensätze" an das Ereignis "Drücken" der gewünschten Taste. Übergeben Sie folgende Parameter:

- Rezepturnummer/-name = 1
- Datensatznummer/-name = 0
- Dateiname = c:\temp\orange.csv, für Basic Panels "\\USB_X61.1\orange.csv"
- Überschreiben = 1
- Statusmeldung ausgeben = 1

An Stelle der Konstanten können Sie auch Variablen angeben. Je nach Projektierung kann der Bediener die gewünschten Werte in E/A-Felder eingeben oder aus der Steuerung lesen. Auf diese Weise kann der Bediener bestimmen, welche Rezepturdatensätze exportiert werden.

Ablauf am Bediengerät

Sobald die Taste betätigt wird, wird die Systemfunktion ausgelöst. Die Konstanten oder Variablen werden ausgewertet. Alle Datensätze der Rezeptur 1 werden in die Datei orange.csv exportiert. Wenn die Datei bereits vorhanden ist, wird sie überschrieben.

Nach dem Export der Datensätze wird eine Systemmeldung ausgegeben.

3.1.25 ExportDataRecordsWithChecksum (Panels, Comfort Panels, RT Advanced)

Beschreibung

Exportiert einen oder alle Datensätze einer Rezeptur in eine CSV-Datei und generiert für jede Zeile in der Datei eine Prüfsumme.

Pro Rezeptur wird eine Datei angelegt.

Verwendung in der Funktionsliste

ExportiereDatensätzeMitPrüfsumme (Rezepturnummer/-name, Datensatznummer/-name, Dateiname, Überschreiben, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

ExportDataRecordsWithChecksum (Recipe_number/name, Data_record number/name, File_name, Overwrite, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, aus der Rezepturdatensätze exportiert werden. Geben Sie "0" an, wenn Sie aus allen verfügbaren Rezepturen Rezepturdatensätze exportieren wollen.

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der exportiert wird. Geben Sie "0" an, wenn Sie alle Rezepturdatensätze exportieren wollen.

Dateiname

Name der CSV-Datei, in welche die Rezepturdatensätze exportiert werden. Geben Sie den Pfad und die Dateiendung mit an z.B. "C:\TEMP\Orange.CSV".

Wenn Sie als Ablageort eine Speicherkarte verwenden, dann geben Sie den Ablageort wie folgt an: "\StorageCard\".

Wenn Sie nur einen Dateinamen und keinen Pfad festlegen, wird die Datei in dem Verzeichnis abgelegt, aus dem die Runtime gestartet wurde. Wenn unter Betriebssystem Windows 7 keine Schreibrechte für das Verzeichnis bestehen, wird die Datei im Benutzerverzeichnis im Ordner "VirtualStore" abgelegt.

Wenn Sie nur den Pfad für den Export angeben, wird der Dateiname jeweils automatisch aus dem Rezepturnamen erzeugt. Die Voraussetzung ist, dass das Verzeichnis z. B. "D:\Temp\" angelegt ist. Wenn das Verzeichnis "D:\Temp" nicht angelegt ist, wird der Verzeichnisname als Präfix für den Dateinamen verwendet, Temp_Rezepturnamen.csv.

Überschreiben

Legt fest, ob eine bereits vorhandene CSV-Datei gleichen Namens überschrieben wird:

0 (hmiOverwriteForbidden) = Nein: CSV-Datei wird nicht überschrieben. Der Exportvorgang wird nicht ausgeführt.

1 (hmiOverwriteAlways) = Ja: CSV-Datei wird ohne Nachfrage überschrieben.

2 (hmiOverwriteWithConfirmation) = Nach Bestätigung: CSV-Datei wird erst nach Bestätigung überschrieben.

Statusmeldung ausgeben

Legt fest, ob nach dem Export eine Statusmeldung ausgegeben wird:

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

Exportformat der Rezepturdatensätze

Wenn für die Exportdatei die Dateierweiterung ".csv" gewählt wird, dann werden lediglich gültige Zeichen aus dem ANSI-Zeichensatz unterstützt. Dies gilt auch für Trennzeichen bei Dezimalzahlen und Listenelementen. Die verwendeten Trennzeichen werden in den Ländereinstellungen des Betriebssystems des exportierenden Computers festgelegt.

Für den Export kann auch das Dateiformat "Unicode Text" (".txt") gewählt werden. Dieses Dateiformat unterstützt alle in WinCC und WinCC Runtime verwendbaren Zeichen. Die verwendeten Trennzeichen werden auch hier in den Ländereinstellungen des Betriebssystems des exportierenden Computers festgelegt. Bei diesem Dateiformat wird als Trennzeichen für Listenelemente immer der Tabulator verwendet.

Die zugehörige Funktion zum Import der Daten unterstützt ebenfalls die beiden Dateiformate ".csv" und ".txt" (Unicode).

Anwendungsbeispiel

Sie wollen über eine Taste alle Datensätze exportieren und mit einer Prüfsumme versehen.

Hinweise zur Projektierung

Projektieren Sie die Systemfunktion "ExportiereDatensätzeMitPrüfsumme" an das Ereignis "Drücken" der gewünschten Taste. Übergeben Sie folgende Parameter:

- Rezepturnummer/-name = 1
- Datensatznummer/-name = 0
- Dateiname = c:\temp\orange.csv
- Überschreiben = 1
- Statusmeldung ausgeben = 1

An Stelle der Konstanten können Sie auch Variablen angeben. Je nach Projektierung kann der Bediener die gewünschten Werte in E/A-Felder eingeben oder aus der Steuerung lesen. Auf diese Weise kann der Bediener bestimmen, welche Rezepturdatensätze exportiert werden.

Ablauf am Bediengerät

Sobald die Taste betätigt wird, wird die Systemfunktion ausgelöst. Die Konstanten oder Variablen werden ausgewertet. Alle Datensätze der Rezeptur 1 werden in die Datei orange.csv

exportiert und mit Prüfsummen versehen. Wenn die Datei bereits vorhanden ist, wird sie überschrieben.

Nach dem Export der Datensätze wird eine Systemmeldung ausgegeben.

3.1.26 ExportImportUserAdministration (Panels, Comfort Panels, RT Advanced)

Beschreibung

Exportiert alle Benutzer der Benutzerverwaltung des gerade aktiven Projekts in die angegebene Datei oder importiert die Benutzer aus der angegebenen Datei in das gerade aktive Projekt.

In der Benutzerverwaltung werden Benutzer, deren Passwörter und Rechte gespeichert.

Beim Import werden alle Benutzer überschrieben. Die importierten Benutzer sind sofort gültig.

Verwendung in der Funktionsliste

ExportiereImportiereBenutzerverwaltung (Dateiname, Richtung)

Verwendung in benutzerdefinierten Funktionen

ExportImportUserAdministration (File_name, Direction)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Dateiname

Name der Datei, welche die Passwörter enthält oder in der die Passwörter geschrieben werden. Geben Sie den Ablageort und die Dateiergung (*.txt) mit an, z. B. "C:\TEMP\Passwords.txt".

Hinweis

Wenn Sie als Ablageort eine Storage Card verwenden, dann geben Sie den Ablageort wie folgt an: "\StorageCard\".

Richtung

Legt fest, ob Passwörter exportiert oder importiert werden:

0 (hmiExport) = Export: Passwörter werden exportiert.

1 (hmilImport) = Import: Passwörter werden importiert.

3.1.27 GetBrightness (Panels, Comfort Panels, RT Advanced)

Beschreibung

Liest den Wert für die Helligkeit aus.

Verwendung in der Funktionsliste

LeseHelligkeit (Helligkeit)

Verwendung in benutzerdefinierten Funktionen

GetBrightness (Brightness)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Helligkeit

Variable, in die der Wert geschrieben wird.

3.1.28 GetDataRecordFromPLC (Panels, Comfort Panels, RT Advanced)

Beschreibung

Überträgt den angegebenen Rezepturdatensatz aus der Steuerung auf das Speichermedium des Bediengeräts.

Verwendung in der Funktionsliste

LeseDatensatzAusSteuerung (Rezepturnummer/-name, Datensatznummer/-name, Überschreiben, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

GetDataRecordFromPLC (Recipe_number/name, Data_record_number/name, Overwrite, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, aus der Rezepturdatensätze übertragen werden.

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der aus der Steuerung auf den Datenträger des Bediengeräts übertragen wird.

Überschreiben

Legt fest, ob ein bereits vorhandener Rezepturdatensatz gleichen Namens überschrieben wird:

0 (hmiOverwriteForbidden) = Nein: Rezepturdatensatz wird nicht überschrieben. Der Übertragungsvorgang wird nicht ausgeführt.

1 (hmiOverwriteAlways) = Ja: Rezepturdatensatz wird ohne Nachfrage überschrieben.

2 (hmiOverwriteWithPrompting) = Nach Bestätigung: Rezepturdatensatz wird erst nach Bestätigung überschrieben.

Statusmeldung ausgeben

Legt fest, ob nach der Übertragung eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

Anwendungsbeispiel

Sie wollen über eine Taste einen Datensatz aus der Steuerung auf den Datenträger des Bediengeräts übertragen.

Hinweise zur Projektierung

Projektieren Sie die Systemfunktion "LeseDatensatzAusSteuerung" an das Ereignis "Drücken" der gewünschten Taste. Übergeben Sie folgende Parameter:

Rezepturnummer/-name = 1

Datensatznummer/-name = 1

Überschreiben = 1

Statusmeldung ausgeben = 1

An Stelle der Konstanten können Sie auch Variablen angeben. Je nach Projektierung kann der Bediener die gewünschten Werte in E/A-Felder eingeben oder aus der Steuerung lesen.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Auf diese Weise kann der Bediener bestimmen, welcher Rezepturdatensatz aus der Steuerung übertragen wird.

Ablauf am Bediengerät

Sobald die Taste betätigt wird, wird die Systemfunktion ausgelöst. Die Konstanten oder Variablen werden ausgewertet und der erste Datensatz der Rezeptur 1 wird aus der Steuerung auf den Datenträger des Bediengeräts übertragen. Ist der Rezepturdatensatz bereits vorhanden, wird er überschrieben.

Nach der Übertragung wird eine Systemmeldung ausgegeben.

3.1.29 GetDataRecordName (Panels, Comfort Panels, RT Advanced)

Beschreibung

Schreibt die Namen der angegebenen Rezeptur und des angegebenen Rezepturdatensatzes in die angegebenen Variablen.

Hinweis

Wenn die Rezeptur oder der Rezepturdatensatz nicht existiert, werden Platzhalter ("###") in die Variablen geschrieben.

Verwendung in der Funktionsliste

LeseDatensatzname (Rezepturnummer, Datensatznummer, Rezepturname, Datensatzname, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

GetDataRecordName (Recipe_number, Data_record_number, Recipe_name, Data_record_name, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer

Nummer der Rezeptur, deren Name in die angegebene Variable geschrieben wird.

Datensatznummer

Nummer des Rezepturdatensatzes, dessen Name in die angegebene Variable geschrieben wird.

Rezepturname

Die Variable, in die der Rezepturname geschrieben wird. Die Variable muss vom Typ STRING sein.

Datensatzname

Die Variable, in die der Name des Rezepturdatensatzes geschrieben wird. Die Variable muss vom Typ STRING sein.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

Anwendungsbeispiel

Sie wollen die Namen der angezeigten Rezeptur und den Namen des angezeigten Rezepturdatensatzes am Bediengerät ausgeben.

Projektieren Sie folgende Variablen:

- "RecNumber" vom Typ INTEGER
- "RecDataNumber" vom Typ INTEGER
- "RecName" vom Typ STRING
- "RecDataName" vom Typ STRING

Projektieren Sie eine Rezepturanzeige mit den Variablen "RecNumber" für die Rezepturnummer und "RecDataNumber" für die Datensatznummer.

Projektieren Sie an das Ereignis "Drücken" einer Schaltfläche die Systemfunktion "LeseDatensatzname" und übergeben Sie folgende Parameter:

- Rezepturnummer: RecNumber
- Datensatznummer: RecDataNumber
- Rezepturname: RecName
- Datensatzname: RecDataName

Projektieren Sie zwei Ausgabefelder und verbinden Sie diese mit den Variablen "RecName" und "RecDataName".

Wählen Sie in der Rezepturanzeige eine Rezeptur- und eine Datensatznummer aus. Sobald Sie die Schaltfläche betätigen, wird die Systemfunktion ausgelöst und die Namen der Rezeptur und des Rezepturdatensatzes werden in den beiden Ausgabefeldern angezeigt.

3.1.30 GetDataRecordTagsFromPLC (Panels, Comfort Panels, RT Advanced)

Beschreibung

Überträgt die Werte des Rezepturdatensatzes, der in der Steuerung geladen ist, in die Rezepturvariablen.

Verwenden Sie diese Systemfunktion z.B. beim Teach-In-Betrieb an einer Maschine.

Verwendung in der Funktionsliste

LeseDatensatzvariablenAusSteuerung (Rezepturnummer/-name, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

GetDataRecordTagsFromPLC (Recipe_number/name, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name des Rezepturdatensatzes, dessen Werte aus der Steuerung in die Variablen geschrieben werden.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.31 GetGroupNumber (Panels, Comfort Panels, RT Advanced)

Beschreibung

Liest die Nummer der Gruppe, welcher der am Bediengerät angemeldete Benutzer angehört, und schreibt sie in die angegebene Variable.

Verwendung in der Funktionsliste

LeseGruppennummer (Variable)

Verwendung in benutzerdefinierten Funktionen

GetGroupNumber (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in welche die Nummer der Gruppe geschrieben wird.

3.1.32 GetPassword (Panels, Comfort Panels, RT Advanced)

Beschreibung

Schreibt das Kennwort des gerade am Bediengerät angemeldeten Benutzers in die angegebene Variable.

Hinweis

Stellen Sie sicher, dass der Wert der angegebenen Variable nicht an anderer Stelle im Projekt angezeigt wird.

Hinweis

Kennwörter von SIMATIC Logon-Benutzern können nicht ausgelesen werden.

Verwendung in der Funktionsliste

LeseKennwort (Variable)

Verwendung in benutzerdefinierten Funktionen

GetPassword (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in die das Kennwort geschrieben wird.

3.1.33 GetUserName (Panels, Comfort Panels, RT Advanced)

Beschreibung

Schreibt den Benutzernamen des gerade am Bediengerät angemeldeten Benutzers in die angegebene Variable.

Wenn die angegebene Variable eine Steuerungsanbindung besitzt, steht der Benutzername auch in der Steuerung zur Verfügung. Mit dieser Systemfunktion können Sie z.B. eine benutzerabhängige Freischaltung bestimmter Funktionalitäten realisieren.

Verwendung in der Funktionsliste

LeseBenutzername (Variable)

Verwendung in benutzerdefinierten Funktionen

GetUserName (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in die der Benutzername geschrieben wird.

3.1.34 GoToEnd (Panels, Comfort Panels, RT Advanced)

Beschreibung

Führt die Tastenfunktion <Ende> am Bediengerät aus.

Verwenden Sie diese Systemfunktion, wenn das Bediengerät diese Funktionalität nicht standardmäßig besitzt.

Diese Systemfunktion ist nur für Funktionstasten verwendbar.

Verwendung in der Funktionsliste

GeheZumEnde

Verwendung in benutzerdefinierten Funktionen

GoToEnd

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.35 GoToHome (Panels, Comfort Panels, RT Advanced)

Beschreibung

Führt die Tastenfunktion <Pos1> am Bediengerät aus.

Verwenden Sie diese Systemfunktion, wenn das Bediengerät diese Funktionalität nicht standardmäßig besitzt.

Diese Systemfunktion ist nur für Funktionstasten verwendbar.

Verwendung in der Funktionsliste

GeheZumAnfang

Verwendung in benutzerdefinierten Funktionen

GoToHome

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.36 ImportDataRecords (Panels, Comfort Panels, RT Advanced)

Beschreibung

Importiert einen oder alle Datensätze einer Rezeptur aus einer CSV-Datei.

Wenn ein Pfad angegeben wird, werden alle Dateien des angegebenen Verzeichnisses importiert.

Verwendung in der Funktionsliste

ImportiereDatensätze (Dateiname, Datensatznummer/-name, Überschreiben, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

ImportDataRecords (File_name, Data_record number/name, Overwrite, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Dateiname

Name der CSV-Datei, aus der die Rezepturdatensätze importiert werden. Geben Sie den Ablageort und die Dateiergung (*.csv) mit an z. B. "C:\TEMP\Orange.csv", für Basic Panels "\USB_X61.1\Orange.csv".

Hinweis

Gilt nicht für Basic Panels: Wenn Sie als Ablageort eine Storage Card verwenden, dann geben Sie den Ablageort wie folgt an: "\StorageCard\".

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der importiert wird. Geben Sie "0" an, wenn Sie alle Rezepturdatensätze importieren wollen.

Überschreiben

Legt fest, ob bereits vorhandene Rezepturdatensätze überschrieben werden:

0 (hmiOverwriteForbidden) = Nein: Rezepturdatensätze werden nicht überschrieben. Der Importvorgang wird nicht ausgeführt.

1 (hmiOverwriteAlways) = Ja: Rezepturdatensätze werden ohne Nachfrage überschrieben.

2 (hmiOverwriteWithPrompting) = Nach Bestätigung: Rezepturdatensätze werden erst nach Bestätigung überschrieben.

Statusmeldung ausgeben

Legt fest, ob nach dem Import eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

Projektierbare Objekte

Objekt	Ereignis
Variable	Wertänderung Obergrenze überschritten Untergrenze unterschritten
Funktionstaste (global)	Loslassen Drücken
Funktionstaste (lokal)	Loslassen Drücken
Bild	Aufgebaut Abgebaut
Bildobjekt	Drücken Loslassen Klicken Ändern (oder Umschalten beim Schalter) Einschalten Ausschalten Aktivieren Deaktivieren
Aufgabenplaner	Zeit abgelaufen

3.1.37 ImportDataRecordsWithChecksum (Panels, Comfort Panels, RT Advanced)**Beschreibung**

Importiert einen oder alle Datensätze einer Rezeptur aus einer CSV-Datei mit Prüfsumme und verifiziert die Prüfsumme.

Verwendung in der Funktionsliste

ImportiereDatensätzeMitPrüfsumme (Dateiname, Datensatznummer/-name, Überschreiben, Statusmeldung ausgeben, Bearbeitungsstatus, Prüfsumme verifizieren)

Verwendung in benutzerdefinierten Funktionen

ImportDataRecordsWithChecksum (File_name, Data_record number/name, Overwrite, Output_status_message, Processing_status, Verify_Checksum)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Dateiname

Name der CSV-Datei, aus der die Rezepturdatensätze importiert werden. Geben Sie den Pfad und die Dateierweiterung mit an, z. B. "C:\TEMP\Orange.CSV".

Wenn Sie als Speichermedium eine Storage Card verwenden, dann geben Sie den Ablageort wie folgt an: "\StorageCard\".

Wenn Sie ein Verzeichnis statt einer einzelnen CSV-Datei angeben, werden alle Dateien des angegebenen Verzeichnisses importiert.

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der importiert wird. Geben Sie "0" an, wenn Sie alle Rezepturdatensätze importieren wollen.

Überschreiben

Legt fest, ob bereits vorhandene Rezepturdatensätze überschrieben werden:

0 (hmiOverwriteForbidden) = Nein: Rezepturdatensätze werden nicht überschrieben. Der Importvorgang wird nicht ausgeführt.

1 (hmiOverwriteAlways) = Ja: Rezepturdatensätze werden ohne Nachfrage überschrieben.

2 (hmiOverwriteWithConfirmation) = Nach Bestätigung: Rezepturdatensätze werden erst nach Bestätigung überschrieben.

Statusmeldung ausgeben

Legt fest, ob nach dem Import eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z. B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

Prüfsumme verifizieren

Legt fest, ob die Prüfsumme beim Import verifiziert wird:

0 (hmiFalse) = Nein: Prüfsumme wird nicht verifiziert.

1 (hmiTrue) = Ja: Prüfsumme wird verifiziert.

3.1.38 IncreaseTag (Panels, Comfort Panels, RT Advanced)

Beschreibung

Addiert den angegebenen Wert zum Wert der Variablen.

$$X = X + a$$

Hinweis

Die Systemfunktion verwendet als Eingangswert und Ausgangswert dieselbe Variable. Wenn Sie diese Systemfunktion zum Umrechnen eines Werts verwenden, müssen Sie mit einer Hilfsvariablen arbeiten. Der Hilfsvariablen können Sie mit der Systemfunktion "SetzeVariable" den Variablenwert zuweisen.

Wenn Sie die Systemfunktion an Ereignisse einer Meldung projektieren und die Variable wird im aktuellen Bild nicht verwendet, ist nicht sichergestellt, dass der tatsächliche Wert der Variablen in der Steuerung verwendet wird. Dies kann durch Setzen der Erfassungsart "Zyklisch fortlaufend" verbessert werden.

Verwendung in der Funktionsliste

ErhöheVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

IncreaseTag (Tag, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, zu welcher der angegebene Wert addiert wird.

Wert

Der Wert, der addiert wird.

3.1.39 InverseLinearScaling (Panels, Comfort Panels, RT Advanced)

Beschreibung

Weist der Variablen X einen Wert zu, der aus dem Wert der angegebenen Variablen Y über die lineare Funktion $X = (Y - b) / a$ berechnet wird.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Die Variablen X und Y dürfen nicht identisch sein. Diese Systemfunktion ist die Umkehrfunktion zur Systemfunktion "LineareSkalierung".

Hinweis

Die Variablen X und Y dürfen nicht identisch sein. Wenn Sie eine Variable auf sich selbst umrechnen wollen, müssen Sie eine Hilfsvariable verwenden.

Mit der Systemfunktion "SetzeVariable" können Sie den Wert der umzurechnenden Variablen der Hilfsvariablen zuweisen.

Verwendung in der Funktionsliste

InvertiereLineareSkalierung (X, Y, b, a)

Verwendung in benutzerdefinierten Funktionen

InvertLinearScaling (X, Y, b, a)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

X

Die Variable, welcher der berechnete Wert aus der linearen Gleichung zugewiesen wird.

Y

Die Variable, deren Wert zur Berechnung verwendet wird.

b

Der Wert, der subtrahiert wird.

a

Der Wert, durch den dividiert wird.

Beispiel

Der folgende Programmcode weist der Variable varX einen Wert mit der Funktion InverseLinearScaling zu.

```
{
BYTE varX;
BYTE Yvalue = 10;
BYTE bvalue = 3;
BYTE avalue = 4;

//Inverse linear scaling
InverseLinearScaling (varX, Yvalue, bvalue, avalue);

printf ("varX = %d\r\n, varX);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.1.40 InvertBit (Panels, Comfort Panels, RT Advanced)

Beschreibung

Invertiert den Wert der angegebenen Variable vom Typ "Bool":

- Hat die Variable den Wert 1 (TRUE), wird sie auf 0 (FALSE) gesetzt.
- Hat die Variable den Wert 0 (FALSE), wird sie auf 1 (TRUE) gesetzt.

Verwendung in der Funktionsliste

InvertiereBit (Variable)

Verwendung in benutzerdefinierten Funktionen

InvertBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, dessen Bit gesetzt wird.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Beispiel

Der folgende Programmcode invertiert den Wert der booleschen Variable `b_value` und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `b_saved` aus.

```
{
BOOL b_value = 0;
BOOL b_saved = b_value;

//Invert variable
invertBit(b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

3.1.41 InvertBitInTag (Panels, Comfort Panels, RT Advanced)

Beschreibung

Invertiert ein Bit in der angegebenen Variablen:

- Hat das Bit in der Variablen den Wert 1 (TRUE), wird es auf 0 (FALSE) gesetzt.
- Hat das Bit in der Variablen den Wert 0 (FALSE), wird es auf 1 (TRUE) gesetzt.

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie statt dessen die Systemfunktion "InvertiereBit".

Verwendung in der Funktionsliste

InvertiereBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

InvertBitInTag (Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Tag

Die Variable, in der das angegebene Bit gesetzt wird.

Bit

Die Nummer des Bits, das gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in einer Variablen von rechts nach links gezählt. Die Zählung beginnt mit 0.

Beispiel

Der folgende Programmcode invertiert ein Bit an der angegebenen Position `bitposition` in der Variable `bvalue` und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `bsaved` aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Invert bit in bitposition
InvertBitInTag (bvalue, bitposition);
//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

3.1.42 LinearScaling (Panels, Comfort Panels, RT Advanced)

Beschreibung

Weist der Variablen `Y` einen Wert zu, der aus dem Wert der angegebenen Variablen `X` über die lineare Funktion $Y = (a * X) + b$ berechnet wird.

Die Umkehrfunktion ist die Systemfunktion "InvertiereLineareSkalierung".

Hinweis

Die Variablen `X` und `Y` dürfen nicht identisch sein. Wenn Sie eine Variable auf sich selbst umrechnen wollen, müssen Sie eine Hilfsvariable verwenden.

Mit der Systemfunktion "SetzeVariable" können Sie den Wert der umzurechnenden Variablen der Hilfsvariablen zuweisen.

Verwendung in der Funktionsliste

LineareSkalierung (Y, a, X, b)

Verwendung in benutzerdefinierten Funktionen

LinearScaling (Y, a, X, b)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Y

Die Variable, welcher der berechnete Wert aus der linearen Gleichung zugewiesen wird.

a

Der Wert, mit dem multipliziert wird.

X

Die Variable, deren Wert zur Berechnung verwendet wird.

b

Der Wert, der addiert wird.

Beispiel

Der folgende Programmcode weist mit der Funktion LinearScaling der Variable Yvar einen Wert zu.

```
{
BYTE Yvar;
BYTE Xvalue = 10;
BYTE bvalue = 3;
BYTE avalue = 4;

// linear scaling
LinearScaling ( Yvar, avalue, Xvalue, bvalue);

printf ("Yvar = %d\r\n", Yvar);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.1.43 LoadDataRecord (Panels, Comfort Panels, RT Advanced)

Beschreibung

Lädt den angegebenen Rezepturdatensatz vom Speichermedium des Bediengeräts in die Rezepturvariablen. Verwenden Sie diese Systemfunktion, um z. B. einen Rezepturdatensatz im Rezepturbild anzuzeigen.

Aktivieren Sie in den Einstellungen zur Synchronisation der Rezeptur die Option "Rezepturanzeige und Rezepturvariablen abgleichen". Wenn diese Option deaktiviert ist, hat die Systemfunktion keine Auswirkung.

Verwendung in der Funktionsliste

LadeDatensatz (Rezepturnummer/-name, Datensatznummer/-name, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

LoadDataRecord (Recipe_number/name, Data_record_number/name, Confirmation, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, aus der ein Rezepturdatensatz geladen wird.

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der geladen wird.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.44 Logoff (Panels, Comfort Panels, RT Advanced)

Beschreibung

Meldet den aktuellen Benutzer am Bediengerät ab.

Verwendung in der Funktionsliste

Abmelden

Verwendung in benutzerdefinierten Funktionen

Logoff

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.45 Logon (Panels, Comfort Panels, RT Advanced)

Beschreibung

Meldet den aktuellen Benutzer am Bediengerät an.

Verwendung in der Funktionsliste

Anmelden (Kennwort, Benutzername)

Verwendung in benutzerdefinierten Funktionen

Logon (Password, User_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Kennwort

Die Variable, aus der das Kennwort des anzumeldenden Benutzers gelesen wird.

Wenn der Benutzer angemeldet ist, wird das Kennwort in der Variablen gelöscht.

Benutzername

Die Variable, aus welcher der Benutzername des anzumeldenden Benutzers gelesen wird.

3.1.46 LookupText (Panels, Comfort Panels, RT Advanced)

Beschreibung

Ermittelt den Listeneintrag aus einer Textliste. Das Ergebnis ist abhängig vom Wert und von der gewählten Runtime-Sprache. Das Ergebnis wird in eine Variable vom Datentyp "String" gespeichert.

Verwendung in der Funktionsliste

TextAbrufen (Ausgabertext, Wert, Sprache, Textliste)

Verwendung in benutzerdefinierten Funktionen

LookupText (Output_text, Index, Language, Text_list)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Ausgabertext

Die Variable, in der das Ergebnis geschrieben wird.

Wert

Die Variable, die den Wert des Listeneintrags festlegt.

Sprache

Legt fest, in welcher Runtime-Sprache der Listeneintrag ermittelt wird.

- Runtime-Sprache
Sprachkürzel gemäß der VBScript-Referenz , z. B. "de-DE" für Deutsch (Deutschland) oder "en-US" für Englisch (Vereinigte Staaten von Amerika). Die Auswahl ist abhängig von den aktivierten Runtime-Sprachen.
- Variable
Die Variable, welche die Sprache enthält. Die Runtime-Sprache geben Sie als Dezimalwert der Länder-Kennung an, z. B. 1031 für Deutsch - Standard, 1033 für Englisch - USA. Nähere Hinweise hierzu finden Sie in den Grundlagen von VBScript bei "Gebietsschema-ID (LCID)-Diagramm.
- Integer
Die Zahl, die der Reihenfolge der Runtime-Sprachen bei Sprachumschaltung entspricht. Die Auswahl ist abhängig von den aktivierten Runtime-Sprachen, z. B. "0" für die Sprache, die beim ersten Runtime-Start erscheint. Nähere Hinweise hierzu finden Sie im Thema "Sprachen in Runtime".

Textliste

Legt die Textliste fest. Aus der Textliste wird der Listeneintrag ausgelesen.

3.1.47 NotifyUserAction (Panels, Comfort Panels, RT Advanced)

Beschreibung

Mit dieser Systemfunktion erfassen Sie Benutzeraktionen im Audit Trail, die nicht automatisch im Audit Trail eingetragen werden. Außerdem können Sie mithilfe dieser Systemfunktion vom

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Benutzer eine Quittierung oder eine elektronische Unterschrift für die Bedienaktion sowie einen Kommentar einfordern. Voraussetzung für die Verwendung der Systemfunktion ist, dass unter "Runtime-Einstellungen > GMP" die GMP-konforme Projektierung aktiviert ist.

Wenn Sie in einer Funktion die Systemfunktion "NotifyUserAction" verwenden und die Eingabe über die Schaltfläche "Abbrechen" beenden, kann der Debugger aufgerufen werden. Um dieses Verhalten abzufangen, können Sie in einer Funktion die Anweisung "On Error Resume Next" verwenden. Diese Anweisung bewirkt, dass nach einem Laufzeitfehler die Folgeanweisung ausgeführt wird. Wenn Sie die Anweisung "On Error Resume Next" verwenden, wird auch die Ausgabe von Systemmeldungen unterdrückt.

Verwendung in der Funktionsliste

ErfasseBenutzeraktion (Bestätigungstyp, Kommentar erforderlich, Kategorie, Objektname, Beschreibung)

Verwendung in benutzerdefinierten Funktionen

NotifyUserAction (Confirmation_type, Mandatory_commenting, Category, Object_name, Description)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bestätigungstyp

Legt fest, wie die Aktion bestätigt werden muss

0 = (None): Keine Bestätigung erforderlich, ein Eintrag im Audit Trail wird erzeugt

1 = (Acknowledgement): Quittierung, der Benutzer muss die Aktion quittieren; ein Eintrag im Audit Trail wird erzeugt

2 = (Digital Signature): elektronische Unterschrift, ein Dialogfenster wird eingeblendet, in dem der Benutzer seine elektronische Unterschrift eingeben muss; ein Eintrag im Audit Trail wird erzeugt

Kommentar erforderlich

Legt fest, ob der Benutzer einen Kommentar eingeben muss. Der Kommentar wird im Audit Trail archiviert.

0 = (True): Wahr; ein Dialogfenster wird eingeblendet, in dem der Benutzer einen Kommentar eingeben muss

1 = (False): Falsch; kein Kommentar erforderlich

Kategorie

Kategorie oder Klassenname des geänderten Objekts

Objektname

Name des geänderten Objekts

Beschreibung

Text, der die zu archivierende Benutzeraktion beschreibt.

3.1.48 OpenAllLogs (Panels, Comfort Panels, RT Advanced)

Beschreibung

Stellt die Verbindung zwischen WinCC und den Archiven wieder her. Die Archivierung kann fortgesetzt werden.

Hinweis

Um die Archivierung wieder zu starten, führen Sie die Systemfunktion "StarteArchivierung" aus.

Verwendung in der Funktionsliste

ÖffneAlleArchive

Verwendung in benutzerdefinierten Funktionen

OpenAllLogs

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

Anwendungsbeispiel

Sie wollen in Runtime den Datenträger wechseln, auf dem die Prozesswerte archiviert werden.

Hinweise zur Projektierung

Projektieren Sie die Systemfunktionen "StoppeArchivierung" und "SchlieÙeAlleArchive" an die Schaltfläche "Close Archive".

Projektieren Sie die Systemfunktionen "ÖffneAlleArchive" und "StarteArchivierung" an die Schaltfläche "Open Archive".

Übergeben Sie als Parameter jeweils den Namen des Archivs, das gestoppt und gestartet werden soll.

Ablauf am Bediengerät

Wenn Sie die Schaltfläche "Close Archive" drücken, wird das angegebene Archiv gestoppt und alle geöffneten Archive geschlossen. Sie können den Datenträger wechseln. Mit der

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Schaltfläche "Open Archive" öffnen Sie alle Archive. Die Archivierung in das angegebene Archiv wird fortgesetzt.

3.1.49 OpenCommandPrompt (Panels, Comfort Panels, RT Advanced)

Beschreibung

Öffnet eine Windows-Eingabeaufforderung.

Verwenden Sie diese Systemfunktion, wenn Sie z.B. Dateien kopieren wollen oder eine andere Anwendung aufrufen wollen.

Verwendung in der Funktionsliste

ÖffneEingabeaufforderung

Verwendung in benutzerdefinierten Funktionen

OpenCommandPrompt

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.50 OpenControlPanel (Panels, Comfort Panels, RT Advanced)

Beschreibung

Öffnet einen Dialog über den ausgewählte Einstellungen der Systemsteuerung verändert werden können.

Mit dieser Systemfunktion stellen Sie unter anderem am Bediengerät Folgendes ein:

- Eigenschaften und Wert der IP-Adresse
- Identifikation des Benutzers im Netzwerk
- WinCC Internet Settings

Hinweis

Sicherheit im Projekt

Mit der Systemfunktion „ÖffneDialogSystemsteuerung“ umgehen Sie den SecureMode am Bediengerät. Um die Sicherheit Ihres Projekts zu gewähren, treffen Sie die entsprechenden Vorkehrungen.

Verwendung in der Funktionsliste

ÖffneDialogSystemsteuerung (Dialog)

Verwendung in benutzerdefinierten Funktionen

-

Parameter

Dialog

Legt den zu öffnenden Dialog der Systemsteuerung fest.

- PROFINET_X1: Einstellung der IP-Adresse und der Ethernet-Parameter
- PROFINET_X3: Einstellung der IP-Adresse und der Ethernet-Parameter; nur bei Comfort Panel KP 1500, TP 1500; TP1900, TP2200
- WinCC Internet Settings: Einstellung für Web Server, E-Mail-Benachrichtigung, sofern das Bediengerät diese Funktionen unterstützt
- Network ID: Einstellung zur Identifikation im Netzwerk, sofern das Bediengerät diese Funktionen unterstützt

3.1.51 OpenInternetExplorer (Panels, Comfort Panels, RT Advanced)

Beschreibung

Öffnet den Internet Explorer am Bediengerät.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Wenn der Internet Explorer bereits geöffnet ist, wird der Internet Explorer mit Aufruf der Systemfunktion geschlossen und erneut geöffnet.

Hinweis

Der Internet Explorer speichert Daten temporär im DRAM-Dateisystem des Bediengeräts, z.B. zuletzt aufgerufene Web-Seiten.

Diese Daten können Sie mit der Systemfunktion "SichereRAMDateisystem" sichern, damit sie auch nach dem Neustart des Bediengeräts zur Verfügung stehen.

Verwendung in der Funktionsliste

ÖffneInternetExplorer (Startseite)

Verwendung in benutzerdefinierten Funktionen

OpenInternetExplorer (Start_page)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Startseite

Die Seite, die beim Aufruf des Internet Explorer geladen wird, z.B. "www.siemens.com".

3.1.52 OpenScreenKeyboard (Panels, Comfort Panels, RT Advanced)

Beschreibung

Blendet die Bildschirmtastatur ein oder aus.

Die Bildschirmtastatur bleibt so lange geöffnet, bis Sie die Bildschirmtastatur explizit schließen. Damit können Sie die Bildschirmtastatur auch in anderen Applikationen verwenden.

Verwendung in der Funktionsliste

ÖffneBildschirmtastatur (Darstellung)

Verwendung in benutzerdefinierten Funktionen

OpenScreenKeyboard (Display_mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Darstellung

Legt fest, ob das Fenster mit der Bildschirmtastatur minimiert oder maximiert geöffnet wird:

0 (hmiScreenKeyboardMinimized) = Minimiert

1 (hmiScreenKeyboardMaximized) = Maximiert

3.1.53 OpenTaskManager (Panels, Comfort Panels, RT Advanced)

Beschreibung

Blendet den Task-Manager ein.

Mit dem Task-Manager können Sie am Bediengerät zu anderen geöffneten Anwendungen wechseln.

Hinweis

Das Aussehen des Task-Managers ist abhängig vom installierten Betriebssystem.

Verwendung in der Funktionsliste

ÖffneTaskManager

Verwendung in benutzerdefinierten Funktionen

OpenTaskManager

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Skripte unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.54 PageDown (Panels, Comfort Panels, RT Advanced)

Beschreibung

Führt die Tastenfunktion <Bild Ab> am Bediengerät aus.

Diese Systemfunktion kann nur für Funktionstasten verwendet werden.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Verwendung in der Funktionsliste

SeiteAbwärts

Verwendung in benutzerdefinierten Funktionen

PageDown

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

-

3.1.55 PageUp (Panels, Comfort Panels, RT Advanced)

Beschreibung

Führt die Tastenfunktion <Bild Auf> am Bediengerät aus.

Diese Systemfunktion kann nur für Funktionstasten und für Aufgaben mit zeitlichen Trigger verwendet werden.

Verwendung in der Funktionsliste

SeiteAufwärts

Verwendung in benutzerdefinierten Funktionen

PageUp

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

-

3.1.56 PrintReport (Panels, Comfort Panels, RT Advanced)

Beschreibung

Druckt das angegebene Protokoll auf dem Drucker aus, der am Bediengerät angeschlossen ist. Das Protokoll wird in der Sprache gedruckt, die Sie am Bediengerät eingestellt haben.

Hinweis

Wenn Runtime beendet wird, während ein mit Hilfe der Systemfunktion Archivdaten gedruckt werden, so wird das Protokoll ab dem Zeitpunkt zu dem Runtime beendet ist, nicht mehr mit Daten versorgt.

Verwendung in der Funktionsliste

DruckeProtokoll (Protokoll)

Verwendung in benutzerdefinierten Funktionen

PrintReport (Report)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Protokoll

Name des Protokolls, das ausgedruckt wird.

Hinweis

Wenn Sie für die Funktion "DruckeProtokoll" ein neues Protokoll über den Dialog "Funktionsliste" anlegen, erscheint beim Kompilieren folgende Warnung: "Das Protokoll "Protokoll_1" hat keine Druckseiten."

Um die Warnung zu beheben, öffnen Sie "Protokoll_1" über das Projektfenster und kompilieren Sie das Projekt neu.

3.1.57 PrintScreen (Panels, Comfort Panels, RT Advanced)

Beschreibung

Druckt das gerade am Bediengerät angezeigte Bild auf dem Drucker aus, der am Bediengerät angeschlossen ist.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Geöffnete Fenster werden ebenfalls mitgedruckt.

Hinweis

Die Druckereinstellungen werden aus den aktuellen Einstellungen im Windows-Betriebssystem übernommen.

Verwendung in der Funktionsliste

DruckeBild

Verwendung in benutzerdefinierten Funktionen

PrintScreen

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.58 ResetBit (Panels, Comfort Panels, RT Advanced)

Beschreibung

Setzt den Wert einer Variablen vom Typ "Bool" auf 0 (FALSE).

Verwendung in der Funktionsliste

RücksetzeBit (Variable)

Verwendung in benutzerdefinierten Funktionen

ResetBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable vom Typ BOOL, die auf 0 (FALSE) gesetzt wird.

Beispiel

Der folgende Programmcode setzt den Wert der booleschen Variable `b_value` mit der Funktion `ResetBit` auf 0 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `b_saved` aus.

```
{
BOOL b_value = 1;
BOOL b_saved = b_value;

//Reset bit
ResetBit (b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

3.1.59 ResetBitInTag (Panels, Comfort Panels, RT Advanced)

Beschreibung

Setzt ein Bit in der angegebenen Variablen auf 0 (FALSE).

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie stattdessen die Systemfunktion "RücksetzeBit".

Verwendung in der Funktionsliste

RücksetzeBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

ResetBitInTag (Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in der ein Bit auf 0 (FALSE) gesetzt wird.

Bit

Die Nummer des Bits, das auf 0 (FALSE) gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in der angegebenen Variablen unabhängig von der verwendeten Steuerung von rechts nach links gezählt. Die Zählung beginnt mit 0.

Beispiel

Der folgende Programmcode setzt ein Bit an der angegebenen Position `bitposition` der Variablen `bvalue` auf 0 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `bsaved` aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Reset bit in bitposition
ResetBitInTag (bvalue, bitposition);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

3.1.60 SafelyRemoveHardware (Panels, Comfort Panels)

Beschreibung

Prüft ob auf dem externen Speichermedium schreibende oder lesende Zugriffe ausgeführt werden. Wenn keine Zugriffe ausgeführt werden, lässt sich das externe Speichermedium ohne Datenverlust entfernen.

Verwendung in der Funktionsliste

HardwareSicherEntfernen(Pfad, Ergebnis)

Verwendung in benutzerdefinierten Funktionen

SafelyRemoveHardware(Path, Result)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Pfad

Pfad des Speichermediums , z. B. \Storage Card USB\

Ergebnis

Variable, in der das Ergebnis eingetragen wird.

TRUE : Speichermedium kann sicher entfernt werden. Eine entsprechende Systemmeldung wird ausgegeben.

FALSE: Speichermedium kann nicht entfernt werden. Eine entsprechende Systemmeldung wird ausgegeben.

3.1.61 SaveDataRecord (Panels, Comfort Panels, RT Advanced)

Beschreibung

Speichert die aktuellen Werte der Rezepturvariablen als Datensatz auf dem Speichermedium des Bediengeräts.

Verwenden Sie diese Systemfunktion, um z.B. einen Rezepturdatensatz im Rezepturbild zu speichern.

Verwendung in der Funktionsliste

SpeichereDatensatz (Rezepturnummer/-name, Datensatznummer/-name, Überschreiben, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

SaveDataRecord (Recipe_number/name, Data_record_number/name, Overwrite, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, in der ein Rezepturdatensatz gespeichert wird.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der gespeichert wird. Wenn in der Rezeptur ein Datensatz mit dieser Nummer oder diesem Namen nicht existiert, wird unabhängig vom Wert des Parameters "Überschreiben" ein neuer Datensatz angelegt.

Überschreiben

Legt fest, ob ein bereits vorhandener Datensatz überschrieben wird:

0 (hmiOverwriteForbidden) = Nein: Der Rezepturdatensatz wird nicht überschrieben, der Datensatz wird nicht gespeichert.

1 (hmiOverwriteAlways) = Ja: Der Rezepturdatensatz wird ohne Nachfrage überschrieben.

2 (hmiOverwriteWithConfirmation) = Nach Bestätigung: Der Rezepturdatensatz wird erst nach Bestätigung durch den Benutzer überschrieben.

Statusmeldung ausgeben

Legt fest, ob nach dem Speichern eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.62 SendEMail (Panels, Comfort Panels, RT Advanced)

Beschreibung

Schickt eine E-Mail vom Bediengerät aus an den angegebenen Empfänger.

Verwenden Sie diese Systemfunktion, wenn z.B. im Servicefall die Meldung direkt an den zuständigen Servicetechniker weitergeleitet werden soll.

Hinweis

Um Meldungen zusätzlich als E-Mail zu versenden, muss das HMI-System über einen E-Mail-Client verfügen.

Verwendung in der Funktionsliste

SendEMail (Adresse, Betreff, Text, Antwortadresse)

Verwendung in benutzerdefinierten Funktionen

SendEMail (Address, Subject, Text, Return_address)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Adresse

Die E-Mail-Adresse des Empfängers.

Betreff

Die Betreffzeile der E-Mail.

Text

Der Text, der in der E-Mail versendet wird.

Antwortadresse

Die E-Mail-Adresse, an die der Empfänger die Antwort auf diese E-Mail senden soll.

3.1.63 SetAcousticSignal (Panels, Comfort Panels, RT Advanced)

Beschreibung

Konfiguriert die akustische Rückmeldung des Bediengeräts bei der Bedienung des Touch-Displays.

Hinweis

Beim Neustart des Bediengeräts wird die Konfiguration wieder hergestellt, die beim Ausschalten eingestellt war.

Verwendung in der Funktionsliste

SetzeAkustischesSignal (Lautstärke)

Verwendung in benutzerdefinierten Funktionen

SetAcousticSignal (Volume)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Lautstärke

Legt fest, ob und wie laut ein akustisches Signal ausgegeben wird:

-1 (hmiToggle) = Umschalten: Schaltet die Ausgabe des akustischen Signals, wie folgt um:
Stumm > Leise > Laut.

0 (hmiMuted) = Stumm: kein akustisches Signal

1 (hmiQuiet) = Leise: leises akustisches Signal

2 (hmiLoud) = Laut: lautes akustisches Signal

3.1.64 SetAlarmReportMode (Panels, Comfort Panels, RT Advanced)

Beschreibung

Schaltet das automatische Protokollieren von Meldungen auf dem Drucker ein oder aus.

Verwendung in der Funktionsliste

SetzeProtokolliermodusMeldung (Modus)

Verwendung in benutzerdefinierten Funktionen

SetAlarmReportMode (Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob Meldungen automatisch auf dem Drucker protokolliert werden:

0 (hmiDisablePrinting) = Protokollierung Aus: Meldungen werden nicht automatisch gedruckt.

1 (hmiEnablePrinting) = Protokollierung Ein: Meldungen werden automatisch gedruckt.

-1 (hmiToggle) = Umschalten: Schaltet zwischen den beiden Zuständen hin und her.

3.1.65 SetAndGetBrightness (Panels, Comfort Panels, RT Advanced)

Beschreibung

Legt die Helligkeit des Displays für ein MP 377 Touch daylight readable fest. Dabei kann der Wert für die Helligkeit absolut oder relativ zum aktuellen Wert interpretiert werden.

Hinweis

Beim Neustart des Bediengeräts wird die Konfiguration wieder hergestellt, die im Control Panel eingestellt ist.

Verwendung in der Funktionsliste

SetzeUndLeseHelligkeit (Helligkeit, Modus, Aktueller Wert)

Verwendung in benutzerdefinierten Funktionen

SetAndGetBrightness (Brightness, Mode, Actual brightness)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Helligkeit

Neuer Wert für die Helligkeit.

Modus

Legt fest ob der neue Wert für die Helligkeit absolut oder relativ zum aktuellen Wert gesetzt wird.

Aktueller Wert

Variable in welche der aktuelle Wert der Helligkeit abgelegt wird.

3.1.66 SetBit (Panels, Comfort Panels, RT Advanced)

Beschreibung

Setzt den Wert einer Variablen vom Typ "Bool" auf 1 (TRUE).

Verwendung in der Funktionsliste

SetzeBit (Variable)

Verwendung in benutzerdefinierten Funktionen

SetBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable vom Typ BOOL, die auf 1 (TRUE) gesetzt wird.

Beispiel

Der folgende Programmcode setzt den Wert der booleschen Variable b_value mit der Funktion SetBit auf 1 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert b_saved aus.

```
{
BOOL b_value = 0;
BOOL b_saved = b_value;

//Set bit
  SetBit (b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

3.1.67 SetBitInTag (Panels, Comfort Panels, RT Advanced)

Beschreibung

Setzt ein Bit in der angegebenen Variablen auf 1 (TRUE).

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie statt dessen die Systemfunktion "SetzeBit".

Verwendung in der Funktionsliste

SetzeBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

SetBitInTag(Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in der ein Bit auf 1 (TRUE) gesetzt wird.

Bit

Die Nummer des Bits, das auf 1 (TRUE) gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in der angegebenen Variablen unabhängig von der verwendeten Steuerung von rechts nach links gezählt. Die Zählung beginnt mit 0.

Hinweis

Eine notwendige Bedingung für eine zuverlässige Funktionalität ist eine garantierte Aktualisierung der verwendeten Variablen mit den aktuellen Prozesswerten. Projektieren Sie die Variable deswegen in einem E/A-Feld oder die Systemfunktion an einem Bildobjekt z. B. einer Schaltfläche.

Wenn Sie die Systemfunktion an ein kurzzeitiges Ereignis z. B. Kommen einer Meldung projiziert haben, dann erreichen Sie aktuelle Prozesswerte nur, wenn Sie die Variable auf dauerhaft lesen setzen.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Beispiel

Der folgende Programmcode setzt ein Bit an der angegebenen Position bitposition der Variablen bvalue auf 1 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert bsaved aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Reset bit in bitposition
SetBitInTag (bvalue, bitposition);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

3.1.68 SetBrightness (Panels, Comfort Panels, RT Advanced)

Beschreibung

Legt die Helligkeit des Displays fest.

Hinweis

Beim Neustart des Bediengeräts wird die Konfiguration wieder hergestellt, die im Control Panel/ Start Center eingestellt ist.

Gilt für Basic Panels 2nd Generation, Mobile Panels und Comfort Panels:

Der Wert für die Systemfunktion "SetzeHelligkeit" ist zwischen 0 % und 100 % einstellbar. Der eingestellte Wert wird auf das Bediengerät übertragen. Die Helligkeitseinstellungen auf dem Bediengerät sind unter "Start Center > Settings > Display" einsehbar und editierbar. Die Bediengeräte unterstützen eine Helligkeitseinstellung zwischen 10 % und 100 %.

Wenn der Systemfunktion „SetzeHelligkeit“ ein Wert von 0 % zugewiesen ist, ist das Display des Bediengeräts in Runtime standardmäßig ausgeschaltet. Wenn der Bediener das Display berührt, wechselt das Display zur vorherigen Helligkeitseinstellung.

Wenn der Systemfunktion „SetzeHelligkeit“ ein Wert zwischen 1 % und 10 % zugewiesen ist und der Bediener die Display-Einstellungen im Start Center öffnet, stellt sich die Helligkeit auf 10 % um.

Verwendung in der Funktionsliste

SetzeHelligkeit (Wert)

Verwendung in benutzerdefinierten Funktionen

SetBrightness (Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Wert

Neuer Wert für die Helligkeit.

3.1.69 SetConnectionMode (Panels, Comfort Panels, RT Advanced)

Beschreibung

Baut die angegebene Verbindung auf oder ab.

Hinweis

Eine Verbindung zur Steuerung kann nur dann hergestellt werden, wenn Sie am Bediengerät die Betriebsart ONLINE eingestellt haben. Verwenden Sie dazu die Systemfunktion "SetzeBetriebsartGerät".

Verwendung in der Funktionsliste

SetzeVerbindungsmodus (Modus, Verbindung)

Verwendung in benutzerdefinierten Funktionen

SetConnectionMode (Mode, Connection)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob die Verbindung mit der Steuerung hergestellt wird oder getrennt wird:

0 (hmiOnline) = Online: Verbindung wird hergestellt.

1 (hmiOffline) = Offline: Verbindung wird getrennt.

Verbindung

Die Steuerung, die mit dem Bediengerät verbunden wird. Den Namen der Steuerung legen Sie im Verbindungs-Editor fest.

Mehrfache Verwendung der Systemfunktion in einer benutzerdefinierten Funktion

Wenn Sie die Systemfunktion "SetzeVerbindungsmodus" für verschiedene Verbindungen verwenden, kann es vorkommen, dass nicht alle Systemfunktionen korrekt ausgeführt werden. Um dies zu vermeiden, gehen Sie wie folgt vor:

1. Legen Sie eine Variable vom Datentyp "BOOL" mit dem Startwert "0" an.
2. Projektieren Sie an das Ereignis "Wertänderung" der HMI-Variablen die Systemfunktion "SetzeVerbindungsmodus". Wenn Sie z.B. 3 Verbindungen abbauen wollen, dann müssen die Systemfunktion dreimal projektieren.
3. Wenden Sie in der benutzerdefinierten Funktion die Systemfunktion "InvertBit" auf die HMI-Variable an.

Anwendungsbeispiel

Typische Anwendungsbeispiele für diese Systemfunktion sind die beiden folgenden:

- Test
Solange keine Steuerung am Bediengerät angeschlossen ist, werden während des Tests am Bediengerät keine Fehlermeldungen ausgegeben. Wenn das Bediengerät an eine Steuerung angeschlossen ist, können Sie die Verbindung zur Steuerung auf Tastendruck herstellen.
- Inbetriebnahme
Für eine Anlage sollen mehrere Steuerungen projiziert werden. Dazu projektieren Sie zunächst alle Steuerungen bis auf eine "offline". Nach der Inbetriebnahme der ersten Steuerung stellen Sie die Verbindung zu jeder weiteren Steuerung über Tastendruck her. Auf diese Weise nehmen Sie die anderen Steuerungen nacheinander in Betrieb.

3.1.70 SetDataRecordTagsToPLC (Panels, Comfort Panels, RT Advanced)

Beschreibung

Überträgt die Werte der Rezepturvariablen in die Steuerung. Die Rezepturvariablen enthalten die Werte des Datensatzes, der am Bediengerät angezeigt wird

Verwendung in der Funktionsliste

SchreibeDatensatzvariablenInSteuerung (Rezepturnummer/-name, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

SetDataRecordTagsToPLC (Recipe_ number/name, Processing_ status)

Verwendbar, sofern das projizierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, aus der ein Rezepturdatensatz an die Steuerung übertragen wird.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.71 SetDataRecordToPLC (Panels, Comfort Panels, RT Advanced)

Beschreibung

Überträgt den angegebenen Rezepturdatensatz direkt vom Datenträger des Bediengeräts an die Steuerung, die mit dem Bediengerät verbunden ist.

Hinweis

Die Werte des Rezepturdatensatzes müssen dabei nicht am Bediengerät angezeigt werden.

Verwendung in der Funktionsliste

SchreibeDatensatzInSteuerung (Rezepturnummer/-name, Datensatznummer/-name, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

SetDataRecordToPLC (Recipe_number/name, Data_record_number/name, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, aus der ein Rezepturdatensatz an die Steuerung übertragen wird.

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Datensatznummer/-name

Nummer oder Name des Rezepturdatensatzes, der an die Steuerung übertragen wird.

Statusmeldung ausgeben

Legt fest, ob nach der Übertragung eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.72 SetDaylightSavingTime (Panels, Comfort Panels, RT Advanced)

Beschreibung

Die Systemfunktion "SetzeSommerzeit" schaltet die Uhrzeit des Bediengeräts auf Sommerzeit bzw. Winterzeit um.

Die Zeitumstellung wird unmittelbar nach Ausführung der Systemfunktion vorgenommen.

Hinweis

Die Systemfunktion "SetzeSommerzeit" unterstützt keine Zeitzonen ohne Sommerzeit.

Hinweis

Windows 7

Die Systemfunktion "SetzeSommerzeit" wird auf PC-basierten Bediengeräten unter Windows 7 nicht unterstützt.

Verwendung in der Funktionsliste

SetzeSommerzeit(Sommerzeit)

Verwendung in benutzerdefinierten Funktionen

SetDaylightSavingTime (Daylight_saving_time)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Sommerzeit

Legt fest, ob am Bediengerät die Sommerzeit eingestellt wird:

0 = Sommerzeit ist nicht aktiviert

1 = Sommerzeit ist aktiviert

3.1.73 SetDeviceMode (Panels, Comfort Panels, RT Advanced)

Beschreibung

Schaltet die Betriebsart am Bediengerät um. Folgende Betriebsarten sind möglich: "Online", "Offline" und "Laden".

Verwendung in der Funktionsliste

SetzeBetriebsartGerät (Betriebsart)

Verwendung in benutzerdefinierten Funktionen

SetDeviceMode (Operating_mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Betriebsart

Legt die Betriebsart des Bediengeräts fest:

0 (hmiOnline) = Online: Die Verbindung zur Steuerung wird hergestellt. Dabei wird immer der projektierte Verbindungsstatus eingestellt. Die Stati, die zuletzt in Runtime verwendet wurden, werden nicht berücksichtigt.

1 (hmiOffline) = Offline: Die Verbindung zur Steuerung wird getrennt.

2 (hmiTransfer) = Laden: Sie können ein Projekt vom Projektierungsrechner auf das Bediengerät übertragen.

Hinweis

Wenn Sie als Bediengerät einen PC verwenden, wird beim Umschalten der Betriebsart nach "Laden" die Runtime-Software beendet.

3.1.74 SetDisplayMode (Panels, Comfort Panels, RT Advanced)

Beschreibung

Ändert die Einstellungen des Fensters, in dem die Runtime-Software läuft.

Standardmäßig läuft die Runtime-Software im Vollbildmodus. Die Taskumschaltung von Windows ist deaktiviert.

Verwendung in der Funktionsliste

SetzeAnzeigemodus (Darstellung)

Verwendung in benutzerdefinierten Funktionen

SetDisplayMode (Display mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Darstellung

Legt die Einstellungen für das Fenster fest, in dem die Runtime-Software läuft:

1 (hmiScreenFull): Vollbild: Titelzeile des Fensters ist nicht sichtbar

2 (hmiScreenMaximized): Maximiert

3 (hmiScreenRestore): Wiederherstellen: Zuletzt verwendete Fenstereinstellung wird verwendet. Diese Darstellung kann nur verwendet werden, wenn das Fenster minimiert oder maximiert angezeigt wird.

4 (hmiScreenMinimized): Minimiert

5 (hmiScreenAutoAdjust): Automatisch: Die Größe des Fensters wird so eingestellt, dass alle enthaltenen Bildobjekte sichtbar sind

6 (hmiScreenOnTop): Vordergrund; abhängig von der Windows-Einstellung wird entweder das Fenster im Vordergrund eingeblendet oder das zum Fenster gehörige Programmsymbol blinkt in der Taskleiste. Die Einstellung kann in der Windows-Konfiguration geändert werden und gilt für alle Windows-Anwendungen.

3.1.75 SetLanguage (Panels, Comfort Panels, RT Advanced)

Beschreibung

Schaltet die Sprache am Bediengerät um. Alle projektierten Texte und Systemmeldungen werden am Bediengerät in der neu eingestellten Sprache dargestellt.

Verwendung in der Funktionsliste

SetzeSprache (Sprache)

Verwendung in benutzerdefinierten Funktionen

SetLanguage (Language)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Sprache

Legt fest, welche Sprache am Bediengerät eingestellt wird. Es sind folgende Angaben möglich

- -1 (hmiToggle) = Umschalten: Wechselt zur nächsten Sprache. Die Reihenfolge legen Sie während der Projektierung im Editor "Projektsprachen" fest.
- Nummer, die Sie im Editor "Runtime-Einstellungen" unter "Sprache & Schriftart" festgelegt haben. Wechselt zu der Sprache mit der angegebenen Nummer.
- Sprache, die Sie im Editor "Runtime-Einstellungen" unter "Sprache & Schriftart" festgelegt haben.
- Sprachkürzel gemäß der VBScript5-Referenz: Wechselt zur Sprache, die dem angegebenen Sprachkürzel entspricht, z. B. "de-DE" für Deutsch (Deutschland) oder "en-US" für Englisch (Vereinigte Staaten von Amerika).
Eine Übersicht aller Sprachkürzel finden Sie in den Grundlagen von VBScript unter dem Thema "Gebietsschema-ID (LCID)-Diagramm".

3.1.76 SetPLCDateTime (Panels, Comfort Panels, RT Advanced)

Beschreibung

Ändert das Datum und die Uhrzeit der verbundenen Steuerung.

Sie Systemfunktion "SetzePLCDatumUhrzeit" ist nur für folgende Steuerungen projektierbar:

- SIMATIC S7 1200
- SIMATIC S7 1500

Verwendung in der Funktionsliste

SetzePLCDatumUhrzeit (Verbindung, Zeit)

Verwendung in benutzerdefinierten Funktionen

SetPLCDateTime (Connection, Time)

Parameter

Verbindung

Verbindung von Steuerung und Bediengerät.

Zeit

Übergibt das Datum und die Uhrzeit des Bediengeräts an die Steuerung. Die Steuerung übernimmt das Datum und die Uhrzeit des Bediengeräts.

3.1.77 SetRecipeTags (Panels, Comfort Panels, RT Advanced)

Beschreibung

Ändert den Status der Rezepturvariablen von "Online" nach "Offline" und umgekehrt

Verwenden Sie diese Systemfunktion, wenn Sie z.B. bei der Inbetriebnahme einer Maschine noch die Rezepturdatensatzwerte fein abstimmen.

Verwendung in der Funktionsliste

SetzeRezepturvariablen (Rezepturnummer/-name, Status, Statusmeldung ausgeben, Bearbeitungsstatus)

Verwendung in benutzerdefinierten Funktionen

SetRecipeTags (Recipe_number/name, Status, Output_status_message, Processing_status)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Rezepturnummer/-name

Nummer oder Name der Rezeptur, in welcher der Rezepturdatensatz gespeichert ist.

Status

Legt den Status der Rezepturvariablen fest:

0 (hmiOnline) = Online: Wertänderungen an den Rezepturvariablen werden sofort an die mit dem Bediengerät verbundene Steuerung übertragen.

1 (hmiOffline) = Offline: Wertänderungen an den Rezepturvariablen werden erst an die mit dem Bediengerät verbundene Steuerung übertragen, wenn Sie z.B. die Systemfunktion "SchreibeDatensatzvariablenInSteuerung" ausführen.

Statusmeldung ausgeben

Legt fest, ob nach dem Speichern eine Statusmeldung ausgegeben wird:

0 (hmiOff) = Aus: Statusmeldung wird nicht ausgegeben.

1 (hmiOn) = Ein: Statusmeldung wird ausgegeben.

Bearbeitungsstatus

Gibt den Bearbeitungsstatus der Systemfunktion zurück. Verwenden Sie den Rückgabewert, um z.B. andere Systemfunktionen erst dann auszuführen, wenn diese Systemfunktion erfolgreich beendet wurde:

2 = Systemfunktion wird gerade ausgeführt.

4 = Systemfunktion wurde erfolgreich beendet.

12 = Systemfunktion wurde nicht ausgeführt, weil ein Fehler aufgetreten ist.

3.1.78 SetScreenKeyboardMode (Panels, Comfort Panels, RT Advanced)

Beschreibung

Aktiviert oder deaktiviert das automatische Einblenden der Bildschirmtastatur am Bediengerät.

Verwenden Sie diese Systemfunktion, wenn Sie das Einblenden der Bildschirmtastatur verhindern wollen, weil Sie z.B. eine externe Tastatur am Bediengerät angeschlossen haben.

Hinweis

Wenn Sie die Systemfunktion "SetzeModusBildschirmtastatur" ("SetScreenKeyboardMode") auf einem nicht Touch-Bediengerät einsetzen, aktivieren Sie in den Geräteeinstellungen im Bereich "Einstellungen für Runtime" die Option "Bildschirmtastatur verwenden".

Verwendung in der Funktionsliste

SetzeModusBildschirmtastatur (Modus)

Verwendung in benutzerdefinierten Funktionen

SetScreenKeyboardMode (Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob die Bildschirmtastatur eingeblendet oder ausgeblendet wird:

0 (hmiOff) = Aus: Bildschirmtastatur wird ausgeblendet

1 (hmiOn) = Ein: Bildschirmtastatur wird eingeblendet

-1 (hmiToggle) = Umschalten: Schaltet zwischen den beiden Zuständen hin und her.

3.1.79 SetTag (Panels, Comfort Panels, RT Advanced)

Beschreibung

Weist der angegebenen Variablen einen neuen Wert zu.

Hinweis

Abhängig vom Variablentyp können Sie mit dieser Systemfunktion Zeichenketten und Zahlen zuweisen.

Verwendung in der Funktionsliste

SetzeVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

SetTag (Tag, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, welcher der angegebene Wert zugewiesen wird.

Wert

Der Wert, welcher der angegebenen Variablen zugewiesen wird.

Hinweis

Die Systemfunktion "SetzeVariable" wird nur nach Aufbau einer Verbindung ausgeführt.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTag den Wert der Variablen gs_tag_bit auf TRUE und speichert den Rückgabewert in der Variable ok.

```
{
BOOL ok;
BOOL bvalue;

//Set the tag to true
ok = SetTag("gs_tag_bit", TRUE);
//error handling
if(ok)
{
    // succeeded
    printf ( "Function has run through.\r\n" );
bvalue = GetTagBit("gs_tag_bit");
    printf ("Value of gs_tag_bit: %d\r\n", bvalue);
}
else
{
    // failed
    printf ( "Error - function failed." );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.1.80 ShiftAndMask (Panels, Comfort Panels, RT Advanced)

Beschreibung

Die Systemfunktion wandelt das Eingangs-Bitmuster der Quellvariablen in ein Ausgangs-Bitmuster der Zielvariablen. Dabei werden Bits verschoben und maskiert.

Hinweis

Wenn Quellvariable und Zielvariable eine unterschiedliche Anzahl von Bits haben, kann die Anwendung der Systemfunktion in der Zielvariablen zu einer Wertebereichsüberschreitung führen.

Verwendung in der Funktionsliste

SchiebenUndMaskieren (Quellvariable, Zielvariable, Zu schiebende Bits, Zu maskierende Bits)

Verwendung in benutzerdefinierten Funktionen

ShiftAndMask (Source_tag, Target_tag, Bits_to_shift, Bits_to_mask)

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Quellvariable

Die Variable enthält das Eingangs-Bitmuster, erlaubt sind Variablen von ganzzahligem Typ z. B. "Byte", "Char", "Int", "UInt", "Long" und "ULong".

Beispiel: Die Quellvariable vom Typ 16 Bit Integer hat den aktuellen Wert 72:
000000001001000.

Zielvariable

In die Variable wird das Ausgangs-Bitmuster gespeichert. Erlaubt sind Variablen von ganzzahligem Typ z. B. "Byte", "Char", "Int", "UInt", "Long" und "ULong".

Beispiel: Das verschobene Eingangs-Bitmuster wird mit der Bit-Maske multipliziert, Bit für Bit UND-verknüpft: 000000000001001. Das Ergebnis hat den Dezimalwert "8" und wird in der Zielvariablen gespeichert.

Achten Sie auf Folgendes:

- Quell- und Zielvariable haben die gleiche Anzahl Bits.
- Die Anzahl der zu schiebenden Bits ist kleiner als die Anzahl der Bits der Quell- und Zielvariablen.
- Zu maskierende Bits hat nicht mehr Bits als Quell- und Zielvariable.

Zu schiebende Bits

Anzahl der Bits, um die das Eingangs-Bitmuster nach rechts verschoben wird. Ein negativer Wert verschiebt das Eingangs-Bitmuster nach links.

Beispiel: "Zu schiebende Bits" hat den Wert "+3". Dann wird beim Aufruf der Systemfunktion das Eingangs-Bitmuster um 3 Bits nach rechts verschoben: 000000000001001.
Links wird mit "0" aufgefüllt. Rechts werden 3 Bits abgeschnitten. Neuer Dezimalwert ist "9".

Hinweis

Wenn die Quellvariable einen vorzeichenbehafteten Datentyp Integer besitzt mit dem Vorzeichen "-", ist das linke Bit "1". Beim Verschieben nach rechts wird dieses Vorzeichen-Bit mit "0" aufgefüllt. Das Vorzeichen wechselt nach "+".

Zu maskierende Bits

Eine Integer-Zahl dient als Bit-Maske. Mit deren Bitmuster wird das verschobene Eingangs-Bitmuster multipliziert. Beispiel: Zahl "2478" mit dem Bitmuster "0000100110101110".

Die Bit-Maske geben Sie auf 3 verschiedene Arten ein:

- Hexadezimal: Geben Sie zuerst als Präfix "0h" oder "0H" ein, gefolgt von einem optionalen Leerzeichen zur besseren Lesbarkeit. Anschließend gruppieren Sie das Bitmuster in Viererblöcken (0000)(1001)(1010)(1110) und stellen jeden Block im 16er System dar: (0)(9)(A)(E). Bei der Eingabe sind nur die Zeichen 0-9, A-F, a-f erlaubt: "0h 09AE".
- Binär: Geben Sie zuerst als Präfix "0b" oder "0B" ein, gefolgt von einem optionalen Leerzeichen zur besseren Lesbarkeit. Anschließend gruppieren Sie das binäre Bitmuster zur Kontrolle in Viererblöcken 0000 1001 1010 1110 mit Leerzeichen dazwischen. Bei der Eingabe sind nur die Zeichen "0" oder "1" erlaubt: "0b 0000 1001 1010 1110".
- Dezimal: Geben Sie direkt ohne Präfix den Wert "2478" ein.

3.1.81 ShowAlarmWindow (Panels, Comfort Panels, RT Advanced)

Beschreibung

Blendet das Meldefenster am Bediengerät ein oder aus.

Verwendung in der Funktionsliste

ZeigeMeldefenster (Objektname, Darstellung)

Verwendung in benutzerdefinierten Funktionen

ShowAlarmWindow (Object_name, Display_mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Objektname

Name der Meldeanzeige, die eingeblendet oder ausgeblendet wird.

Darstellung

Legt fest, ob das Meldefenster eingeblendet oder ausgeblendet wird:

0 (hmiOff) = Aus: Meldeanzeige wird ausgeblendet

1 (hmiOn) = Ein: Meldeanzeige wird eingeblendet

-1 (hmiToggle) = Umschalten: Schaltet zwischen den beiden Zuständen hin und her

3.1.82 ShowOperatorNotes (Panels, Comfort Panels, RT Advanced)

Verwendung

Zeigt den projizierten Tooltip des ausgewählten Objekts an.

Wenn die Systemfunktion an eine Funktionstaste projiziert ist, wird der Tooltip zu dem Bildobjekt angezeigt, das gerade den Fokus besitzt. Wenn gleichzeitig ein Tooltip zum Bild selbst projiziert ist, kann zu diesem mit der Taste <Enter> oder durch Doppelklick auf das Hilfenfenster umgeschaltet werden.

Wenn die Systemfunktion an eine Schaltfläche projiziert ist, wird nur der Tooltip zum aktuellen Bild angezeigt. Wenn gleichzeitig ein Tooltip an die Schaltfläche selbst projiziert ist, wird zuerst der Tooltip zur Schaltfläche angezeigt. Mit der Taste <Enter> oder durch Doppelklick auf das Hilfenfenster wird zum Tooltip des aktuellen Bilds umgeschaltet.

Hinweis

Solange das Hilfenfenster geöffnet ist, lässt sich kein anderes Bildobjekt bedienen. Um die Bildobjekte zu bedienen, schließen Sie das Hilfenfenster.

Schließen des Hilfenfensters

Es gibt folgende Möglichkeiten das Hilfenfenster zu schließen:

Bei Tasten-Bedienung:

- Erneutes Drücken der Taste <HELP>
- Drücken der Taste <ESC>

Bei Touch-Bedienung:

- Bedienen der Schaltfläche 

Verwendung in der Funktionsliste

ZeigeHilfetext (Darstellung)

Verwendung in benutzerdefinierten Funktionen

ShowOperatorNotes (Display_mode)

Verwendbar, sofern das projizierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Darstellung

Legt fest, ob der projizierte Tooltip eingeblendet oder ausgeblendet wird:

0 (hmiOff) = Aus: Projizierter Tooltip wird ausgeblendet

1 (hmiOn) = Ein: Projektierter Tooltip wird eingeblendet

-1 (hmiToggle) = Umschalten: Schaltet zwischen den beiden Zuständen hin und her

3.1.83 ShowPopupScreen (Panels, Comfort Panels, RT Advanced)

Beschreibung

Ruft das Pop-up-Bild auf, z. B. beim Betätigen einer Schaltfläche.

Als Koordinaten können Sie einen konstanten Wert eingeben oder eine Variable zuweisen. Falls das projektierte Pop-up-Bild nicht sichtbar oder nur teilweise sichtbar ist, werden die Koordinaten auf 0.0 gesetzt.

Verwendung in der Funktionsliste

ZeigePopupBild (Bildobjekt)

Verwendung in benutzerdefinierten Funktionen

ShowPopupScreen (Name_of_the_screen, Coordinate_X, Coordinate_Y, Layout)

Parameter

Name des Bildes

Legt den Namen des Bildes fest, das in Runtime beim Betätigen der Schaltfläche erscheint.

Koordinate X

Position des Bildes im aktuellem Bild auf der X-Achse

Koordinate Y

Position des Bildes im aktuellem Bild auf der Y-Achse

Darstellung

Legt den Modus für das Slide-in-Bild fest:

Umschalten

Aus

Ein

3.1.84 ShowSlideInScreen (Panels, Comfort Panels, RT Advanced)

Beschreibung

Ruft das Slide-in-Bild auf, z. B. beim Betätigen einer Schaltfläche.

Verwendung in der Funktionsliste

ZeigeSlideInBild (Bildname, Modus)

Verwendung in benutzerdefinierten Funktionen

ShowSlideInScreen (SlideInScreen_name, Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bildname

Legt das Slide-in-Bild fest, das in Runtime beim Betätigen der Schaltfläche erscheint:

Slide-in-Bild oben

Slide-in-Bild unten

Slide-in-Bild links

Slide-in-Bild rechts

Modus

Legt den Modus für das Slide-in-Bild fest:

Umschalten

Aus

Ein

3.1.85 ShowSoftwareVersion (Panels, Comfort Panels, RT Advanced)

Beschreibung

Blendet die Versionsnummer der Runtime-Software ein oder aus.

Verwenden Sie diese Systemfunktion, wenn Sie z.B. in einem Servicefall die verwendete Version der Runtime-Software benötigen.

Verwendung in der Funktionsliste

ZeigeSoftwareVersion (Darstellung)

Verwendung in benutzerdefinierten Funktionen

ShowSoftwareVersion (Display_mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Darstellung

Legt fest, ob die Versionsnummer eingeblendet wird:

0 (hmiOff) = Aus: Versionsnummer wird nicht eingeblendet

1 (hmiOn) = Ein: Versionsnummer wird eingeblendet

-1 (hmiToggle) = Umschalten: Schaltet zwischen den beiden Zuständen hin und her

3.1.86 ShowSystemAlarm (Panels, Comfort Panels, RT Advanced)

Beschreibung

Zeigt den Wert des übergebenen Parameters als Systemmeldung am Bediengerät an.

Verwendung in der Funktionsliste

ZeigeSystemmeldung (Text/Wert)

Verwendung in benutzerdefinierten Funktionen

ShowSystemAlarm (Text/value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Text/Wert

Der Text oder der Wert, der als Systemmeldung ausgegeben wird.

3.1.87 ShowSystemDiagnosticsWindow (Panels, Comfort Panels)

Beschreibung

Blendet das System-Diagnosefenster am Bediengerät ein oder aus. Das System-Diagnosefenster ist nur im Globalen Bild für Comfort Panels und für WinCC Runtime Advanced verfügbar.

Verwendung in der Funktionsliste

ZeigeSystemDiagnosefenster (Bildobjekt)

Verwendung in benutzerdefinierten Funktionen

ShowSystemDiagnosticsWindow (Target_Object_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bildobjekt

Name des System-Diagnosefensters, das eingeblendet oder ausgeblendet wird.

3.1.88 StartLogging (Panels, Comfort Panels, RT Advanced)

Beschreibung

Startet die Archivierung von Prozesswerten oder Meldungen in dem angegebenen Archiv. Die Funktion kann auch auf Audit-Trails angewendet werden.

Sie können die Archivierung in Runtime mit der Systemfunktion "StoppeArchivierung" unterbrechen.

Verwendung in der Funktionsliste

StarteArchivierung (Archivtyp, Archiv)

Verwendung in benutzerdefinierten Funktionen

StartLogging (Log_type, Log)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Archivtyp

Legt den Typ des Archivs fest:

0 (hmiTagArchive) = Variablenarchiv

1 (hmiAlarmArchive) = Meldearchiv

2 (hmiAudittrailArchive) = Audit-Trail

Archiv

Name des Archivs, das gestartet wird.

3.1.89 StartNextLog (Panels, Comfort Panels, RT Advanced)

Beschreibung

Stoppt die Archivierung von Prozesswerten oder Meldungen für das angegebene Archiv.

Die Archivierung wird in dem nächsten Archiv des segmentierten Umlaufarchivs fortgesetzt, das Sie für das angegebene Archiv projiziert haben.

Wenn Sie für das angegebene Archiv kein segmentiertes Umlaufarchiv projiziert haben, hat die Systemfunktion keine Wirkung.

Verwendung in der Funktionsliste

StarteNächstesArchiv (Archivtyp, Archiv)

Verwendung in benutzerdefinierten Funktionen

StartNextLog (Log_type, Log)

Verwendbar, sofern das projizierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Archivtyp

Legt den Typ des Archivs fest:

0 (hmiTagArchive) = Variablenarchiv

1 (hmiAlarmArchive) = Meldearchiv

Archiv

Name des Archivs, für das die Archivierung gestoppt und im nächsten Archiv fortgesetzt wird.

3.1.90 StartProgram (Panels, Comfort Panels, RT Advanced)

Beschreibung

Startet das angegebene Programm auf dem Bediengerät.

Die Runtime-Software läuft im Hintergrund weiter. Meldungen werden weiterhin ausgegeben und Prozesswerte werden weiterhin aktualisiert.

Wenn Sie die angegebene Anwendung beenden, wird am Bediengerät das Bild angezeigt, das beim Ausführen der Systemfunktion aktiv war.

Verwenden Sie diese Systemfunktion, wenn Sie z.B. Rezepturdatensätze am Bediengerät in MS Excel bearbeiten wollen.

Hinweis

Wenn auf dem Bediengerät Windows CE installiert ist, prüfen Sie während der Projektierung, ob sich die gewünschte Anwendung mit dieser Systemfunktion starten lässt.

Sie können mit dieser Systemfunktion alle Anwendungen starten, die Sie im Dialog "Ausführen" von Windows CE starten können.

Die zu startende Anwendung muss auf dem Bediengerät installiert sein.

Verwendung in der Funktionsliste

StarteProgramm (Programmname, Programmparameter, Darstellung, Auf Programmende warten)

Verwendung in benutzerdefinierten Funktionen

StartProgram (Program_name, Program_parameters, Display_mode, Wait_for_program_to_end)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Programmname

Name und Pfad des Programms, das gestartet wird. Bei diesem Parameter wird die Groß- und Kleinschreibung beachtet.

Hinweis

Wenn der Pfad Leerzeichen enthält, wird das Programm nur dann korrekt gestartet, wenn der Pfad in Anführungszeichen gesetzt ist z. B. "C:\Program Files\START\start.exe".

Programmparameter

Die Parameter, die Sie bei Start des Programms übergeben z.B. eine Datei, die nach Programmstart geöffnet wird.

Die Beschreibung der benötigten Parameter finden Sie in der Dokumentation des zu startenden Programms.

Darstellung

Legt fest, wie das Programmfenster am Bediengerät dargestellt wird:

0 (hmiShowNormal) = Normal

1 (hmiShowMinimized) = Minimiert

2 (hmiShowMaximized) = Maximiert

3 (ShowMinimizedAndInactive) = Minimiert und inaktiv

Auf Programmende warten

Legt fest, ob nach Beenden des aufgerufenen Programms zurück zum Projekt gewechselt wird:

0 (hmiNo) = Nein: Es wird nicht zum Projekt gewechselt

1 (hmiYes) = Ja: Es wird zum Projekt gewechselt

Hinweis

Der Parameter "Auf Programmende warten" steht nur für Runtime Advanced und Panels zur Verfügung.

3.1.91 StopLogging (Panels, Comfort Panels, RT Advanced)

Beschreibung

Stoppt die Archivierung von Prozesswerten oder Meldungen in dem angegebenen Archiv. Die Funktion kann auch auf Audit-Trails angewendet werden.

Sie können die Archivierung in Runtime mit der Systemfunktion "StarteArchivierung" wieder fortsetzen.

Hinweis

Wenn die Archivierung gestoppt ist, besteht immer noch eine Verbindung zwischen WinCC und den Archivdateien oder der Archivdatenbank. Um diese Verbindung zu trennen, verwenden Sie die Systemfunktion "SchlieÙeAlleArchive".

Verwendung in der Funktionsliste

StoppeArchivierung (Archivtyp, Archiv)

Verwendung in benutzerdefinierten Funktionen

StopLogging (Log_type, Log)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Archivtyp

Legt den Typ des Archivs fest:

0 (hmiTagArchive) = Variablenarchiv

1 (hmiAlarmArchive) = Meldearchiv

2 (hmiAudittrailArchive) = Audit-Trail

Archiv

Name des Archivs, das gestoppt wird.

Anwendungsbeispiel

Sie wollen in Runtime den Datenträger wechseln, auf dem die Prozesswerte archiviert werden.

Hinweise zur Projektierung

Projektieren Sie die Systemfunktionen "StoppeArchivierung" und "SchlieÙeAlleArchive" an die Schaltfläche "Close Archive".

Projektieren Sie die Systemfunktionen "ÖffneAlleArchive" und "StarteArchivierung" an die Schaltfläche "Open Archive".

Übergeben Sie als Parameter jeweils den Namen des Archivs, das gestoppt und gestartet werden soll.

Ablauf am Bediengerät

Wenn Sie die Schaltfläche "Close Archive" drücken, wird das angegebene Archiv gestoppt und alle geöffneten Archive geschlossen. Sie können den Datenträger wechseln. Mit der Schaltfläche "Open Archive" öffnen Sie alle Archive und setzen die Archivierung in dem angegebenen Archiv wieder fort.

3.1.92 StopRuntime (Panels, Comfort Panels, RT Advanced)

Beschreibung

Beendet die Runtime-Software und damit das laufende Projekt am Bediengerät.

Verwendung in der Funktionsliste

StoppeRuntime (Modus)

Verwendung in benutzerdefinierten Funktionen

StopRuntime (Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob nach dem Beenden von Runtime auch das Betriebssystem heruntergefahren wird:

0 (hmiStopRuntime) = Runtime: Betriebssystem wird nicht heruntergefahren

1 (hmiStopRuntimeAndOperatingSystem) = Runtime und Betriebssystem: Betriebssystem wird heruntergefahren (bei WinCE nicht möglich)

Beispiel

Der folgende Programmcode fährt Runtime und Betriebssystem herunter.

```
{  
  
//Stop runtime and shutdown  
StopRuntime (hmiStopRuntimeAndOperationSystem);  
  
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.1.93 TerminatePROFIsafe (Panels, Comfort Panels, RT Advanced)

Beschreibung

Trennt die PROFIsafe-Verbindung für den fehlersicheren Betrieb zwischen einem KTP Mobile Panel und der Steuerung.

Nach dem Ausführen der Systemfunktion "PROFIsafeTrennen" kann der Stecker des KTP Mobile Panels von der Steuerung getrennt werden ohne dass die Anlage einen Fehler meldet.

Verwendung in der Funktionsliste

PROFIsafeTrennen

Verwendung in benutzerdefinierten Funktionen

TerminatePROFIsafe

3.1 Systemfunktionen (Panels, Comfort Panels, RT Advanced)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

--

3.1.94 WinACMPGetVersion (Panels, Comfort Panels)

Beschreibung

Liest den Wert der Versionsnummer von WinAC MP aus.

Verwendung in der Funktionsliste

WinACMPLeaseVersion (Version, Aktualisieren)

Verwendung in benutzerdefinierten Funktionen

WinACMPGetVersion (Version, Action)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter AUTOHOTSPOT.

Parameter

Version

Die Variable, die den Wert enthält.

Aktualisieren

Legt fest, ob die Versionsnummer ausgelesen wird:

0 (SwitchOff) = Aus: Versionsnummer wird nicht ausgelesen.

1 (SwitchOn) = Ein: Versionsnummer wird ausgelesen:

3.1.95 WinACMPSetStartAtBoot (Panels, Comfort Panels)

Beschreibung

Stellt ein, ob WinAC MP nach dem Anlauf des Bediengeräts automatisch gestartet wird oder nicht.

Verwendung in der Funktionsliste

WinACMPSetzeAnlaufverhalten (Anlaufverhalten)

Verwendung in benutzerdefinierten Funktionen

WinACMPSetStartAtBoot (Start at boot)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter AUTOHOTSPOT.

Parameter

Anlaufverhalten

Legt fest, ob WinAC MP automatisch gestartet wird.

0 (StartAtBootOff) = Aus: WinAC MP wird bei Anlauf des Bediengeräts nicht gestartet.

1 (StartAtBootOn) = Ein: WinAC MP wird bei Anlauf des Bediengeräts automatisch gestartet.

3.1.96 WinACSetStartMode (Panels, Comfort Panels)

Beschreibung

Stellt die Betriebsart nach Anlauf des Bediengeräts von WinAC MP ein.

Verwendung in der Funktionsliste

WinACMPSetzeStartModus (Autostart)

Verwendung in benutzerdefinierten Funktionen

WinACSetStartMode (Autostart)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter AUTOHOTSPOT.

Parameter

Startverhalten

Legt fest, ob die Autostart-Funktion von WinAC MP aktiviert wird.

0 (AutoStartOff) = Aus: WinAC MP bleibt nach dem Start im Betriebszustand STOP.

1 (AutoStartOn) = Ein: Die WinAC MP wechselt nach dem Start in den Betriebszustand, indem es vor dem Schließen war.

3.2 Systemfunktionen (RT Professional)

3.2.1 ActivateScreen (RT Professional)

Beschreibung

Führt einen Bildwechsel zu dem angegebenen Bild aus.

Um vom Grundbild in das Permanentfenster oder umgekehrt zu wechseln, verwenden Sie die Systemfunktion "AktiviereBildMitNummer".

Verwendung in der Funktionsliste

AktiviereBild (Bildname, Objektnummer)

Verwendung in benutzerdefinierten Funktionen

ActivateScreen (Screen_name, Object_number)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Bildname

Name des Bildes, zu dem gewechselt wird.

Objektnummer

Das Bedienelement, das nach dem Bildwechsel im angegebenen Bild den Fokus erhält. Die Nummer des Bedienelements legen Sie während der Projektierung durch die Tab-Reihenfolge fest.

Wenn Sie "0" angeben:

- Ist der Fokus beim Aufruf der Systemfunktion im Permanentfenster, behält das Permanentfenster den Fokus.
- Ist der Fokus beim Aufruf der Systemfunktion im Grundbild, erhält das erste Bedienelement im angegebenen Bild den Fokus.

Hinweis

Wenn die Systemfunktion "AktiviereBild" an das Ereignis "Rand erreicht" projiziert wird, so ist für den Parameter "Objektnummer" nur der Wert 0 zulässig. Das aktive Objekt wird nicht anhand der Objektnummer festgelegt, sondern anhand der X-Position des aktiven Objekts vor dem Bildwechsel.

Beispiel

Der folgende Programmcode aktiviert mit der Funktion ActivateScreen beim Klicken auf eine Taste das Bild "Screen_2".

```
{  
  
// User defined code  
// i.e. when pressing a button  
ActivateScreen ("Screen_2", 0);  
...  
}
```

3.2.2 ActivateScreenInScreenWindow (RT Professional)

Beschreibung

Führt im angegebenen Bildfenster einen Bildwechsel zu dem angegebenen Bild aus.

Verwendung in der Funktionsliste

AktiviereBildInBildfenster (Bildname, Bildfenster, Neuer Bildname)

Verwendung in benutzerdefinierten Funktionen

ActivateScreenInScreenWindow (Screen_name, Screen_window, New_screen_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Bildname

Name des Bildes, das im Bildfenster angezeigt wird.

Bildfenster

Name des Bildfensters, in dem das neue Bild angezeigt wird.

Neuer Bildname

Name des neuen Bildes, das im Bildfenster angezeigt wird.

Beispiel

Der folgende Programmcode aktiviert mit der Funktion `ActivateScreenInScreenWindow` beim Klicken auf eine Taste das Bild "Screen_2".

```
{
// User defined code
// i.e. when pressing a button
ActivateScreenInScreenWindow (GetParentScreen(screenName),
GetParentScreenWindow(screenName), "Screen_2");
...
}
```

3.2.3 DecreaseTag (RT Professional)

Beschreibung

Subtrahiert den angegebenen Wert vom Wert der Variablen.

$X = X - a$

Hinweis

Die Systemfunktion verwendet als Eingangswert und Ausgangswert dieselbe Variable. Wenn Sie diese Systemfunktion zum Umrechnen eines Wertes verwenden, müssen Sie mit einer Hilfsvariablen arbeiten. Der Hilfsvariablen können Sie mit der Systemfunktion "SetzeVariable" den Variablenwert zuweisen.

Wenn Sie die Systemfunktion an Ereignisse einer Meldung projektieren und die Variable wird im aktuellen Bild nicht verwendet, ist nicht sichergestellt, dass der tatsächliche Variablenwert in der Steuerung verwendet wird. Dies kann durch Setzen der Erfassungsart "Zyklisch fortlaufend" verbessert werden.

Verwendung in der Funktionsliste

VerringereVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

DecreaseTag (Tag, Value)

Verwendbar, sofern das projizierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variable

Die Variable, von welcher der angegebene Wert subtrahiert wird.

Wert

Der Wert, der subtrahiert wird.

Beispiel

Der folgende Programmcode erniedrigt den Wert der Variablen varX um den Wert in der Variable value. Der eingegebene Wert wird in der Variable old_value gesichert und zusammen mit dem neuen Wert varX ausgegeben.

```
{
BYTE varX;
BYTE value;

//user input
...
BYTE old_value = varX;

//Decrease tag
DecreaseTag(varX, value);

//print original value and function result
printf ("User input: %i\r\n, Result of function DecreaseTag: %i\r\n", old_value, varX);
...
}
```

3.2.4 ExportImportUserAdministration (RT Professional)**Beschreibung**

Exportiert alle Benutzer der Benutzerverwaltung des gerade aktiven Projekts in die angegebene Datei oder importiert die Benutzer aus der angegebenen Datei in das gerade aktive Projekt.

In der Benutzerverwaltung werden Benutzer, deren Passwörter und Rechte gespeichert.

Beim Import werden alle Benutzer überschrieben. Die importierten Benutzer sind sofort gültig.

Verwendung in der Funktionsliste

ExportiereImportiereBenutzerverwaltung (Dateiname, Richtung)

Verwendung in benutzerdefinierten Funktionen

ExportImportUserAdministration (File_name, Direction)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Dateiname

Name der Datei, welche die Passwörter enthält oder in der die Passwörter geschrieben werden. Geben Sie den Ablageort und die Dateierweiterung (*.txt) mit an, z. B. "C:\TEMP\Passwords.txt".

Hinweis

Wenn Sie als Ablageort eine Storage Card verwenden, dann geben Sie den Ablageort wie folgt an: "\StorageCard\".

Richtung

Legt fest, ob Passwörter exportiert oder importiert werden:

0 (hmiExport) = Export: Passwörter werden exportiert.

1 (hmImport) = Import: Passwörter werden importiert.

3.2.5 IncreaseTag (RT Professional)

Beschreibung

Addiert den angegebenen Wert zum Wert der Variablen.

$X = X + a$

Hinweis

Die Systemfunktion verwendet als Eingangswert und Ausgangswert dieselbe Variable. Wenn Sie diese Systemfunktion zum Umrechnen eines Wertes verwenden, müssen Sie mit einer Hilfsvariablen arbeiten. Der Hilfsvariablen können Sie mit der Systemfunktion "SetzeVariable" den Variablenwert zuweisen.

Wenn Sie die Systemfunktion an Ereignisse einer Meldung projektieren und die Variable wird im aktuellen Bild nicht verwendet, ist nicht sichergestellt, dass der tatsächliche Variablenwert in der Steuerung verwendet wird. Dies kann durch Setzen der Erfassungsart "Zyklisch fortlaufend" verbessert werden.

Verwendung in der Funktionsliste

ErhöheVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

IncreaseTag (Tag, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variable

Die Variable, zu welcher der angegebene Wert addiert wird.

Wert

Der Wert, der addiert wird.

Beispiel

Der folgende Programmcode erhöht den Wert der Variablen varX um den Wert in der Variable value. Der eingegebene Wert wird in der Variable old_value gesichert und zusammen mit dem neuen Wert varX ausgegeben.

```
{
BYTE varX;
BYTE value;

//user input
...
BYTE old_value = varX;

//Increase tag
IncreaseTag(varX, value);

//print original value and function result
printf ("User input: %i\r\n, Result of function IncreaseTag: %i\r\n", old_value, varX);
...
}
```

3.2.6 InverseLinearScaling (RT Professional)

Beschreibung

Weist der Variablen X einen Wert zu, der aus dem Wert der angegebenen Variablen Y über die lineare Funktion $X = (Y - b) / a$ berechnet wird.

Die Variablen X und Y dürfen nicht identisch sein. Diese Systemfunktion ist die Umkehrfunktion zur Systemfunktion "LineareSkalierung".

Hinweis

Die Variablen X und Y dürfen nicht identisch sein. Wenn Sie eine Variable auf sich selbst umrechnen wollen, müssen Sie eine Hilfsvariable verwenden.

Mit der Systemfunktion "SetzeVariable" können Sie den Wert der umzurechnenden Variablen der Hilfsvariablen zuweisen.

Verwendung in der Funktionsliste

InvertiereLineareSkalierung (X, Y, b, a)

Verwendung in benutzerdefinierten Funktionen

InvertLinearScaling (X, Y, b, a)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

X

Die Variable, welcher der berechnete Wert aus der linearen Gleichung zugewiesen wird.

Y

Die Variable, deren Wert zur Berechnung verwendet wird.

b

Der Wert, der subtrahiert wird.

a

Der Wert, durch den dividiert wird.

Beispiel

Der folgende Programmcode weist der Variable varX einen Wert mit der Funktion InverseLinearScaling zu.

```
{
BYTE varX;
BYTE Yvalue = 10;
BYTE bvalue = 3;
BYTE avalue = 4;

//Inverse linear scaling
InverseLinearScaling (varX, Yvalue, bvalue, avalue);

printf ("varX = %d\r\n", varX);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.2.7 InvertBit (RT Professional)

Beschreibung

Invertiert den Wert der angegebenen Variable vom Typ "Bool":

- Hat die Variable den Wert 1 (TRUE), wird sie auf 0 (FALSE) gesetzt.
- Hat die Variable den Wert 0 (FALSE), wird sie auf 1 (TRUE) gesetzt.

Verwendung in der Funktionsliste

InvertiereBit (Variable)

Verwendung in benutzerdefinierten Funktionen

InvertBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, dessen Bit gesetzt wird.

Beispiel

Der folgende Programmcode invertiert den Wert der boolschen Variable b_value und gibt das Ergebnis zusammen mit dem ursprünglichen Wert b_saved aus.

```
{
BOOL b_value = 0;
BOOL b_saved = b_value;

//Invert variable
invertBit(b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

3.2.8 InvertBitInTag (RT Professional)

Beschreibung

Invertiert ein Bit in der angegebenen Variablen:

- Hat das Bit in der Variablen den Wert 1 (TRUE), wird es auf 0 (FALSE) gesetzt.
- Hat das Bit in der Variablen den Wert 0 (FALSE), wird es auf 1 (TRUE) gesetzt.

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie statt dessen die Systemfunktion "InvertiereBit".

Verwendung in der Funktionsliste

InvertiereBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

InvertBitInTag (Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Tag

Die Variable, in der das angegebene Bit gesetzt wird.

Bit

Die Nummer des Bits, das gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in einer Variablen von rechts nach links gezählt. Die Zählung beginnt mit 0.

Beispiel

Der folgende Programmcode invertiert ein Bit an der angegebenen Position `bitposition` in der Variable `bvalue` und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `bsaved` aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Invert bit in bitposition
InvertBitInTag (bvalue, bitposition);
//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

3.2.9 LinearScaling (RT Professional)

Beschreibung

Weist der Variablen `Y` einen Wert zu, der aus dem Wert der angegebenen Variablen `X` über die lineare Funktion $Y = (a * X) + b$ berechnet wird.

Die Umkehrfunktion ist die Systemfunktion "InvertiereLineareSkalierung".

Hinweis

Die Variablen `X` und `Y` dürfen nicht identisch sein. Wenn Sie eine Variable auf sich selbst umrechnen wollen, müssen Sie eine Hilfsvariable verwenden.

Mit der Systemfunktion "SetzeVariable" können Sie den Wert der umzurechnenden Variablen der Hilfsvariablen zuweisen.

Verwendung in der Funktionsliste

LineareSkalierung (Y, a, X, b)

Verwendung in benutzerdefinierten Funktionen

LinearScaling (Y, a, X, b)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Y

Die Variable, welcher der berechnete Wert aus der linearen Gleichung zugewiesen wird.

a

Der Wert, mit dem multipliziert wird.

X

Die Variable, deren Wert zur Berechnung verwendet wird.

b

Der Wert, der addiert wird.

Beispiel

Der folgende Programmcode weist mit der Funktion LinearScaling der Variable Yvar einen Wert zu.

```
{
BYTE Yvar;
BYTE Xvalue = 10;
BYTE bvalue = 3;
BYTE avalue = 4;

// linear scaling
LinearScaling ( Yvar, avalue, Xvalue, bvalue);

printf ("Yvar = %d\r\n", Yvar);
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.2.10 ResetBit (RT Professional)

Beschreibung

Setzt den Wert einer Variablen vom Typ "Bool" auf 0 (FALSE).

Verwendung in der Funktionsliste

RücksetzeBit (Variable)

Verwendung in benutzerdefinierten Funktionen

ResetBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable vom Typ BOOL, die auf 0 (FALSE) gesetzt wird.

Beispiel

Der folgende Programmcode setzt den Wert der booleschen Variable b_value mit der Funktion ResetBit auf 0 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert b_saved aus.

```
{
BOOL b_value = 1;
BOOL b_saved = b_value;

//Reset bit
ResetBit (b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

3.2.11 ResetBitInTag (RT Professional)

Beschreibung

Setzt ein Bit in der angegebenen Variablen auf 0 (FALSE).

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie stattdessen die Systemfunktion "RücksetzeBit".

Verwendung in der Funktionsliste

RücksetzeBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

ResetBitInTag (Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in der ein Bit auf 0 (FALSE) gesetzt wird.

Bit

Die Nummer des Bits, das auf 0 (FALSE) gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in der angegebenen Variablen unabhängig von der verwendeten Steuerung von rechts nach links gezählt. Die Zählung beginnt mit 0.

Beispiel

Der folgende Programmcode setzt ein Bit an der angegebenen Position bitposition der Variablen bvalue auf 0 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert bsaved aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Reset bit in bitposition
ResetBitInTag (bvalue, bitposition);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

3.2.12 SetBit (RT Professional)

Beschreibung

Setzt den Wert einer Variablen vom Typ "Bool" auf 1 (TRUE).

Verwendung in der Funktionsliste

SetzeBit (Variable)

Verwendung in benutzerdefinierten Funktionen

SetBit (Tag)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable vom Typ BOOL, die auf 1 (TRUE) gesetzt wird.

Beispiel

Der folgende Programmcode setzt den Wert der booleschen Variable b_value mit der Funktion SetBit auf 1 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert b_saved aus.

```
{
BOOL b_value = 0;
BOOL b_saved = b_value;

//Set bit
SetBit (b_value);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",b_value, b_saved);
...
}
```

3.2.13 SetBitInTag (RT Professional)

Beschreibung

Setzt ein Bit in der angegebenen Variablen auf 1 (TRUE).

Die Systemfunktion überträgt nach der Änderung des angegebenen Bits die gesamte Variable wieder an die Steuerung. Es wird nicht geprüft, ob sich zwischenzeitlich andere Bits in der Variablen geändert haben. Bediener und Steuerung dürfen auf die angegebene Variable nur lesend zugreifen, bis die Variable wieder an die Steuerung übertragen wurde.

Hinweis

Verwenden Sie diese Systemfunktion nicht, wenn die Steuerung BOOL-Variablen unterstützt. Verwenden Sie statt dessen die Systemfunktion "SetzeBit".

Verwendung in der Funktionsliste

SetzeBitInVariable (Variable, Bit)

Verwendung in benutzerdefinierten Funktionen

SetBitInTag(Tag, Bit)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, in der ein Bit auf 1 (TRUE) gesetzt wird.

Bit

Die Nummer des Bits, das auf 1 (TRUE) gesetzt wird.

Wenn Sie diese Systemfunktion in einer benutzerdefinierten Funktion verwenden, werden die Bits in der angegebenen Variablen unabhängig von der verwendeten Steuerung von rechts nach links gezählt. Die Zählung beginnt mit 0.

Hinweis

Eine notwendige Bedingung für eine zuverlässige Funktionalität ist eine garantierte Aktualisierung der verwendeten Variablen mit den aktuellen Prozesswerten. Projektieren Sie die Variable deswegen in einem E/A-Feld oder die Systemfunktion an einem Bildobjekt z. B. einer Schaltfläche.

Wenn Sie die Systemfunktion an ein kurzzeitiges Ereignis z. B. Kommen einer Meldung projiziert haben, dann erreichen Sie aktuelle Prozesswerte nur, wenn Sie die Variable auf dauerhaft lesen setzen.

Beispiel

Der folgende Programmcode setzt ein Bit an der angegebenen Position `bitposition` der Variablen `bvalue` auf 1 und gibt das Ergebnis zusammen mit dem ursprünglichen Wert `bsaved` aus.

```
{
BYTE bvalue;
BYTE bsaved = bvalue;
BYTE bitposition = 2;

//Reset bit in bitposition
SetBitInTag (bvalue, bitposition);

//print current and saved value
printf ("Current value: %d\r\n, Saved value: %d\r\n",bvalue, bsaved);
...
}
```

3.2.14 SetPropertyByConstant (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft als Zeichenkette fest.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchKonstante (Bildname, Objekt, Name der Eigenschaft, Wert)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByConstant

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes ändern wollen, muss der Parameter "Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

SetPropertyByConstant ("**Bildname**", "", "<Eigenschaftsname>", "<Wert>")

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft geändert wird.

3.2 Systemfunktionen (RT Professional)

Name der Eigenschaft

Name der Eigenschaft, die geändert wird.

Wert

Der Wert, welcher der Eigenschaft zugewiesen wird.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByConstant die Eigenschaft eines Objekts: Im Bild "Trends" wird die Eigenschaft "ToolBarButtonClick" des darin enthaltenen Objekts "Control_1" auf den Wert 26 gesetzt.

```
{  
  
//Name of the picture: Trends  
//Name of the f(t) trend view control: Control_1  
  
SetPropertyByConstant ("Trends", "Control_1", "ToolBarButtonClick", "26");  
  
// User defined code  
...  
}
```

Beispiel: Bildeigenschaft ändern

Der folgende Programmcode ändert mit der Funktion SetPropertyByConstant die Eigenschaft eines Bilds: Im Bild "Trends" wird die Eigenschaft "BackColor" auf den Wert 255 gesetzt.

```
{  
  
//Name of the picture: Trends  
  
SetPropertyByConstant ("Trends", "Trends", "BackColor", "255");  
  
// User defined code  
...  
}
```

Alternativ verwenden Sie anstelle des zweiten Parameters (Objekt) das Kennwort NULL oder einen Leerzeichen-String.

3.2.15 SetPropertyByProperty (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft durch eine andere Objekteigenschaft fest.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchEigenschaft (Bildname, Objekt, Name der Eigenschaft, Ziel Bildname, Ziel Objekt, Ziel Eigenschaftsname)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByProperty (Screen_name, Screen_object, Property_name, Source_screen_name, Source_screen_object, Source_property_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes durch eine andere Bildeigenschaft festlegen wollen, müssen die Parameter "Objekt" und "Ziel Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

```
SetPropertyByProperty  
("<Bildname>", "", "<Eigenschaftsname>", "<Ziel_Bildname>", "", "<Ziel_Eigenschaftsname>")
```

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft an das Zielobjekt übertragen wird.

Name der Eigenschaft

Name der Eigenschaft, an das Zielobjekt übertragen wird.

Ziel Bildname

Name des Bildes, in dem das Zielobjekt enthalten ist.

Ziel Objekt

Name des Zielobjekts, an das die Eigenschaft übertragen wird.

Ziel Eigenschaftsname

Name der Eigenschaft, die geändert wird.

Beispiel

Der folgende Programmcode überträgt mit der Funktion SetPropertyByProperty die Eigenschaft "ToolBarButtonClick" des Objekts "Control_1" im Ausgangsbild "Trend_1" : auf die entsprechende Eigenschaft im Zielbild "Trend_2".

```
{  
  
//Name of source picture: Trend_1  
//Name of target picture: Trend_2  
//Name of the f(t) trend view control: Control_1  
  
SetPropertyByProperty ("Trend_1", "Control_1", "ToolBarButtonClick", "Trend_2",  
"Control_2", "ToolBarButtonClick");  
  
// User defined code  
...  
}
```

3.2.16 SetPropertyByTag (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft durch einen Variablenwert fest.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchVariable (Bildname, Objekt, Name der Eigenschaft, Variablenname)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByTag (Screen_name, Screen_object, Property_name, Tag_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes durch einen Variablenwert festlegen wollen, muss der Parameter "Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

SetPropertyByTag ("<Bildname>", "", "<Eigenschaftsname>", "<Variablenname>")

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft durch den Variablenwert gesetzt wird.

Name der Eigenschaft

Name der Eigenschaft, die durch den Variablenwert gesetzt wird.

Variablenname

Name der Variablen, die den Wert der Eigenschaft enthält.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByTag die Eigenschaft eines Objekts: Beim Klicken auf das Objekt wird der Objektname und das Bild, in dem sich das Objekt befindet übergeben. Die Überschrift CaptionText im Bildfenster erhält den Wert der Variable HMI_value_1.

```
{  
  
SetPropertyByTag (screenName, objectName, "CaptionText", "HMI_value_1");  
  
// User defined code  
...  
}
```

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByTag die Eigenschaft eines Objekts: Im Bild "Trends" wird die Eigenschaft "ToolBarButtonClick" des darin enthaltenen Objekts "Control_1" auf den Wert 26 gesetzt.

```
{  
  
//Name of the picture: Trends  
//Name of the f(t) trend view control: Control_1  
  
SetPropertyByConstant ("Trends", "Control_1", "ToolBarButtonClick", "26");  
  
// User defined code  
...  
}
```

3.2.17 SetPropertyByTagIndirect (RT Professional)

Beschreibung

Legt den Wert einer Objekteigenschaft durch eine Variable fest. Die Variable enthält den Variablennamen, der die Objekteigenschaft festlegt.

Verwendung in der Funktionsliste

SetzeEigenschaftDurchVariableIndirekt (Bildname, Objekt, Name der Eigenschaft, Variablenname)

Verwendung in benutzerdefinierten Funktionen

SetPropertyByTagIndirect (Screen_name, Screen_object, Property_name, Tag_name)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Wenn Sie die Eigenschaft eines Bildes indirekt durch eine Variable festlegen wollen, muss der Parameter "Objekt" leer sein. Verwenden Sie dazu z. B. folgende Syntax:

```
SetPropertyByTagIndirect ("<Bildname>", "", "<Eigenschaftsname>", "<Variablenname>")
```

Parameter

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft durch den Variablenwert gesetzt wird.

Name der Eigenschaft

Name der Eigenschaft, die durch den Variablenwert gesetzt wird.

Variablenname

Name der Variablen, die wiederum den Namen der Variablen enthält, welche die Objekteigenschaft festlegt.

Beispiel

Der folgende Programmcode ändert mit der Funktion SetPropertyByTagIndirect die Eigenschaft eines Objekts: .

```
{  
  
SetPropertyByTagIndirect (GetParentScreen(screenName), GetParentScreenWindow(screenName),  
"ScreenName", "HMI_value_1");  
  
// User defined code  
...  
}
```

3.2.18 SetTag (RT Professional)

Beschreibung

Weist der angegebenen Variablen einen neuen Wert zu.

Hinweis

Abhängig vom Variablentyp können Sie mit dieser Systemfunktion Zeichenketten und Zahlen zuweisen.

Verwendung in der Funktionsliste

SetzeVariable (Variable, Wert)

Verwendung in benutzerdefinierten Funktionen

SetTag (Tag, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Variable

Die Variable, welcher der angegebene Wert zugewiesen wird.

Wert

Der Wert, welcher der angegebenen Variablen zugewiesen wird.

Hinweis

Die Systemfunktion "SetzeVariable" wird nur nach Aufbau einer Verbindung ausgeführt.

Beispiel

Der folgende Programmcode setzt mit der Funktion SetTag den Wert der Variablen gs_tag_bit auf TRUE und speichert den Rückgabewert in der Variable ok.

```
{
BOOL ok;
BOOL bvalue;

//Set the tag to true
ok = SetTag("gs_tag_bit", TRUE);
//error handling
if(ok)
{
    // succeeded
    printf ( "Function has run through.\r\n" );
bvalue = GetTagBit("gs_tag_bit");
    printf ("Value of gs_tag_bit: %d\r\n", bvalue);
}
else
{
    // failed
    printf ( "Error - function failed." );
}
...
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.2.19 SetTagByProperty (RT Professional)

Beschreibung

Legt einen Variablenwert durch den Wert einer Objekteigenschaft fest. Die Änderung wird zusätzlich im Meldesystem protokolliert

Verwendung in der Funktionsliste

SetzteVariableDurchEigenschaft (Variablenname, Bildname, Objekt, Name der Eigenschaft, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagByProperty (Tag_name, Screen_name, Screen_object, Property_name, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der Variablen, deren Wert durch die Objekteigenschaft festgelegt wird.

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Objekt

Name des Objekts, dessen Eigenschaft den Variablenwert liefert.

Name der Eigenschaft

Name der Eigenschaft, die den Variablenwert liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode gibt beim Klick auf ein Kombinationsfeld den Wert des ausgewählten Textes zurück.

```
{  
char* rt_value;  
  
SetTagByProperty (rt_value, screenName, objectName, "SelectedText",  
hmiWithoutOperatorEvent);  
  
...  
}
```

3.2.20 SetTagByTagIndirect (RT Professional)

Beschreibung

Legt den Variablenwert durch den Wert einer indirekten Variablen fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableDurchVariableIndirekt (Variablenname, Variablenname, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagByTagIndirect (Tag_name, Source_tag_name, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der Variable, deren Wert durch eine indirekte Variable festgelegt wird.

Variablenname

Name der indirekten Variablen, die den Variablenwert liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode gibt beim Klick auf eine Taste den Wert der Variable "@LocalMachineName" zurück.

```
{
char* rt_value;

SetTagByTagIndirect (rt_value, "@LocalMachineName", hmiWithoutOperatorEvent);

...
}
```

3.2.21 SetTagIndirect (RT Professional)

Beschreibung

Legt den indirekten Namen für eine Variable fest.

Verwendung in der Funktionsliste

SetzeVariableIndirekt (Variablenname (Ausgabe), LpValue)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirect (Tag_name, Value, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname (Ausgabe)

Name der Variablen, in die der Variablenname geschrieben wird.

LpValue

Name der Variablen, der in die Variable geschrieben wird.

Beispiel

Der folgende Programmcode überträgt beim Klick auf eine Taste den Wert der Variable "value" auf die Variable "result" .

```
{  
BYTE result;  
BYTE value;  
  
SetTagIndirect ("result", "value", hmiWithoutOperatorEvent);  
...  
}
```

3.2.22 SetTagIndirectByProperty (RT Professional)

Beschreibung

Legt einen Variablennamen durch den Wert einer Objekteigenschaft fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableIndirektDurchEigenschaft (Variablenname, Bildname, Bildobjekt, Name der Eigenschaft, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirectByProperty (Tag_name, Screen_name, Screen_object, Property_name, With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der Variablen, deren Variablenname durch die Objekteigenschaft festgelegt wird.

Bildname

Name des Bildes, in dem das Objekt enthalten ist.

Bildobjekt

Name des Objekts, dessen Eigenschaft den Variablennamen liefert.

Name der Eigenschaft

Name der Eigenschaft, die den Variablennamen liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode setzt beim Klick auf die Taste `objectName` die Variable `"rt_value_property"` auf den Wert der Eigenschaft `"FlashingEnabled"` .

```
{  
  Int rt_value_property;  
  SetTagIndirectByProperty ("rt_value_property", screenName, objectName, "FlashingEnabled",  
  hmiWithoutOperatorEvent);  
  ...  
}
```

3.2.23 SetTagIndirectByTagIndirect (RT Professional)

Beschreibung

Legt den Variablenwert durch den Wert einer indirekten Variablen fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableIndirektDurchVariableIndirekt (Variablenname, Variablenname, Mit oder ohne Bedienmeldung)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirectByTagIndirect (Tag_name, Source_tag_name,
With_or_without_operator_event)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname

Name der indirekten Variable, deren Wert durch eine indirekte Variable festgelegt wird.

Variablenname

Name der indirekten Variablen, die den Variablenwert liefert.

Mit oder ohne Bedienmeldung

0 (hmiWithoutOperatorEvent) = Ohne Bedienmeldung

1 (hmiWithOperatorEvent) = Mit Bedienmeldung

Beispiel

Der folgende Programmcode setzt beim Klick auf die Taste objectName die Variable "rt_value" auf den Wert der Variable "value" .

```
{  
Int rt_value;  
Int value;  
  
SetTagIndirectByTagIndirect ("rt_value", "value", hmiWithoutOperatorEvent);  
...  
}
```

3.2.24 SetTagIndirectWithOperatorInputAlarm (RT Professional)

Beschreibung

Legt den indirekten Namen für eine Variable fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableIndirektMitBedienmeldung (Variablenname (Ausgabe), LpValue)

3.2 Systemfunktionen (RT Professional)

Verwendung in benutzerdefinierten Funktionen

SetTagIndirectWithOperatorEvent

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname (Ausgabe)

Name der Variablen, in die der Variablenname geschrieben wird.

LpValue

Name der Variablen, der in die Variable geschrieben wird.

3.2.25 SetTagWithOperatorEvent (RT Professional)

Beschreibung

Legt den Wert für eine Variable fest. Die Änderung wird zusätzlich im Meldesystem protokolliert.

Verwendung in der Funktionsliste

SetzeVariableMitBedienmeldung (Variablenname (Ausgabe), LpValue)

Verwendung in benutzerdefinierten Funktionen

SetTagWithOperatorEvent (Tag_name, Value)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT".

Parameter

Variablenname (Ausgabe)

Name der Variablen, deren Wert festgelegt wird.

LpValue

Der Wert, der in die Variable geschrieben wird.

Beispiel

Der folgende Programmcode überträgt beim Klick auf eine Taste den Wert der Variable "value" auf die Variable "result" .

```
{  
BYTE result;  
BYTE value;  
  
SetTagWithOperatorEvent ("result", "value");  
...  
}
```

3.2.26 ShowLogonDialog (RT Professional)

Beschreibung

Öffnet einen Dialog am Bediengerät, mit dem sich der Benutzer am Bediengerät anmelden kann.

Verwendung in der Funktionsliste

ZeigeAnmeldedialog

Verwendung in benutzerdefinierten Funktionen

ShowLogonDialog

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "AUTOHOTSPOT"

Parameter

--

3.2.27 StopRuntime (RT Professional)

Beschreibung

Beendet die Runtime-Software und damit das laufende Projekt am Bediengerät.

Verwendung in der Funktionsliste

StoppeRuntime (Modus)

Verwendung in benutzerdefinierten Funktionen

StopRuntime (Mode)

Verwendbar, sofern das projektierte Gerät benutzerdefinierte Funktionen unterstützt. Weitere Informationen finden Sie unter "Geräteabhängigkeit".

Parameter

Modus

Legt fest, ob nach dem Beenden von Runtime auch das Betriebssystem heruntergefahren wird:

0 (hmiStopRuntime) = Runtime: Betriebssystem wird nicht heruntergefahren

1 (hmiStopRuntimeAndOperatingSystem) = Runtime und Betriebssystem: Betriebssystem wird heruntergefahren (bei WinCE nicht möglich)

Beispiel

Der folgende Programmcode fährt Runtime und Betriebssystem herunter.

```
{  
  
//Stop runtime and shutdown  
StopRuntime (hmiStopRuntimeAndOperationSystem);  
  
}
```

Der gespeicherte Rückgabewert kann im nachfolgenden Code verarbeitet werden.

3.3 VBScript für Windows (Panels, Comfort Panels, RT Advanced, RT Professional)

3.3.1 VBScript für Windows (Panels, Comfort Panels, RT Advanced, RT Professional)

VBScript

Wenn Sie Visual Basic oder Visual Basic für Applikationen bereits kennen, wird Ihnen VBScript bekannt vorkommen. Und wenn Sie Visual Basic nicht kennen und sich darin einarbeiten, erlernen Sie gleichzeitig alle Visual Basic-Programmiersprachen. Eine gute Einführung in das Programmieren erhalten Sie in den von Microsoft Press verfügbaren Step-by-Step-Büchern.

Grundlegende Informationen zu VBScript Details zu den Sprachelementen finden Sie auf der Homepage von Microsoft:

<http://msdn.microsoft.com/en-us/library/t0aew7h6.aspx>

Gebietsschema-ID (LCID)

Eine Übersicht aller Sprachkürzel finden Sie auf der Homepage von Microsoft:

<http://msdn.microsoft.com/en-us/goglobal/bb964664>

3.4 VBScript für Windows CE (Panels, Comfort Panels)**3.4.1 VBScript für Windows CE (Panels, Comfort Panels)****3.4.1.1 Attr (Panels, Comfort Panels)****Function**

This property of the File control returns a number indicating the file mode that was used to open the file.

Syntax

file.Attr

Parameters**File**

Reference to a File control.

Return Values

The return values listed in the following table indicate the file access mode. If the return value is 0, the file is closed.

Constant	Value
None	0
fsModeInput	1
fsModeOutput	2
fsModeRandom	4
fsModeAppend	8
fsModeBinary	32

Remarks

The Attr property is read-only. Use the Open method of the File control to set the file mode.

3.4.1.2 Close (Panels, Comfort Panels)

Function

This method closes an open File control.

Syntax

```
file.Close
```

Parameters

File

Name of a File control.

Return Values

None.

Remarks

Use the Open method to open a file.

3.4.1.3 CreateObject (Panels, Comfort Panels)

Function

This function creates a reference to an Automation object.

Syntax

```
CreateObject (Object)
```

Parameters

Object

A string containing the ProgID of the object to create.

Return Values

Returns a reference to an Automation object.

Remarks

Use CreateObject to create non-visible ActiveX controls at runtime. You cannot use CreateObject to create graphical objects such as a TreeView control or a ListView control. CreateObject produces objects that cannot respond to events. To produce objects that can respond to events, use the CreateObjectWithEvents function. The following table lists the ProgIDs for the ActiveX controls without events.

Control	ProgID
Microsoft CE File control 6.0	.file
Microsoft CE FileSystem control 6.0	.filesystem
Microsoft CE ImageList control 6.0	CEImageList.imagelistctrl

Example

```
Dim f, fwModeAppend
Set f = CreateObject("FileCtl.File")
fwModeAppend=8
f.Open "\Storage Card\testfile.txt", fwModeAppend
f.Close
```

3.4.1.4 Dir (Panels, Comfort Panels)

Function

This method returns the name of a file, directory, or folder that matches a specified pattern or file attribute.

Syntax

```
File.Dir (Pathname,[Attributes])
```

Parameters

File

Reference to a FileSystem control.

Pathname

Optional. String expression that specifies a file name or path.

Attributes

Optional. Numeric expression whose sum specifies file attributes. If omitted, all files that match pathname are returned.

The following table describes the parameter settings of attributes.

Constant	Value	Description
fsAttrNormal	0	Normal
fsAttrReadOnly	1	Read-only
fsAttrHidden	2	Hidden
fsAttrSystem	4	System file
fsAttrVolume	8	Volume label. If specified, all other attributes are ignored.
fsAttrDirectory	16	Directory or folder
fsAttrArchive	32	Archive

Return Values

String. File name that matches pathname and attributes. Dir returns a zero-length string ("") if pathname is not found.

Remarks

Dir supports the use of multiple-character (*) and single-character (?) wildcards to specify multiple files. You must specify pathname the first time you call the Dir method. In addition, if you specify file attributes you must include pathname.

The Dir method returns the first file name that matches pathname. To get any additional file names that match pathname, call Dir again with no parameters. When no more file names match, Dir returns a zero-length string (" "). Once a zero-length string is returned, you must specify pathname in subsequent calls.

3.4.1.5 EOF (Panels, Comfort Panels)

Function

This property returns True when the end of a file opened for random or sequential input is reached.

Syntax

File.EOF

Parameters

File

Reference to a File control.

Remarks

Use the EOF property to avoid the error generated by attempting to read past the end of a file.

The EOF property returns False until the end of the file has been reached. For files opened with a fsModeRandom or fsModeBinary file mode, EOF returns False until the last executed Get statement is unable to read an entire record.

For files opened with a fsModeBinary file mode, an attempt to read through the file using the Input function until EOF returns True generates an error. Use the LOF and LOC properties instead of EOF when reading binary files with Input, or use Get when using the EOF property. For files opened with a fsModeOutput file mode, EOF always returns True.

3.4.1.6 FileCopy (Panels, Comfort Panels)

Function

This method copies an existing file to a new file.

Syntax

FileSystem.FileCopy PathName, NewPathName

Parameters

FileSystem

Reference to a FileSystem object.

PathName

String that contains the path and file name.

NewPathName

String that contains the file name and path of the new file.

Return Values

None.

Remarks

FileCopy returns an error if the new file does not exist.

3.4.1.7 FileLen (Panels, Comfort Panels)

Function

This method returns a value specifying the length, in bytes, of a file.

Syntax

FileSystem.FileLen(pathname)

Parameters

FileSystem

Reference to a FileSystem control.

Pathname

Required. String expression that specifies a file. The pathname can include a directory or folder.

Return Values

Returns the number of bytes in a file.

Remarks

If the specified file is open when the FileLen method is called, the value returned represents the size of the file immediately before it was opened.

3.4.1.8 FileDateTime (Panels, Comfort Panels)

Function

This method returns a variant (Date) that indicates the date and time when a file was created or last modified.

Syntax

filesystem.FileDateTime(pathname)

Parameters

FileSystem

Reference to a FileSystem control.

Pathname

Required. String expression that specifies a file name. The pathname can include a directory or folder.

Return Values

Returns the date the file was last modified.

Remarks

FileDateTime returns an error if the new file does not exist.

3.4.1.9 Get (Panels, Comfort Panels)

Function

This method reads data from an open disk file into a variable.

Syntax

file.Get Data, [Recnumber]

Parameters

File

Reference to a File control.

Data

Required. Variant variable into which data is read.

Recnumber

Optional. Variant. Record number at which reading begins. For files opened in binary mode, Recnumber specifies the byte position.

Return Values

None

Remarks

Data read with the Get method usually is written to a file with the Put method. The first record or byte in a file is at position 1, the second record or byte is at position 2, and so on. If you omit Recnumber, the next record or byte following the last Get or Put method (or pointed to by the last Seek function) is read.

For files opened in Random mode, the following rules apply:

- If the length of the data being read is less than the length specified in the Len clause of the Open method, Get reads subsequent records on record-length boundaries. The space between the end of one record and the beginning of the next record is padded with the existing contents of the file buffer. Because the amount of padding data cannot be determined with any certainty, it is generally advisable to match the record length with the length of the data being read.
- If Data is a Variant of numeric type, Get reads 2 bytes identifying the VarType of the Variant and then reads the data that goes into the variable. For example, when reading a Variant of VarType 3, Get reads 6 bytes: 2 bytes identifying the Variant as VarType 3 (Long) and 4 bytes containing the Long data. The record length specified by the Len clause in the Open method must be at least 2 bytes greater than the actual number of bytes required to store the variable.
- You can use the Get method to read a Variant array from a disk, but you cannot use Get to read a scalar Variant containing an array. You also cannot use Get to read objects from a disk.
- If the variable being read into is a Variant of VarType 8 (String), Get reads 2 bytes identifying the VarType and 2 bytes indicating the length of the string. Then it reads the string data. The record length specified by the Len clause in the Open method must be at least 4 bytes greater than the actual length of the string.
- If the variable being read into is a dynamic array, Get reads a descriptor whose length equals 2 plus 8 times the number of dimensions, that is, $2 + 8 * \text{NumberOfDimensions}$. The record length specified by the Len clause in the Open method must be greater than or equal to the sum of all the bytes required to read the array data and the array descriptor.

For files opened in Binary mode, the Len clause in the Open method has no effect. Get reads all variables from a disk contiguously; that is, with no padding between records.

3.4.1.10 GetAttr (Panels, Comfort Panels)

Function

This method returns a number representing the attributes of a file, directory, or folder.

Syntax

```
filesystem.GetAttr(pathname)
```

Parameters

Filesystem

Reference to a FileSystem control.

Pathname

Required. String expression that specifies a file name or directory or a folder name. The pathname can include the directory or folder.

Return Values

Sum of attribute values. The following table shows the sums that can be returned.

Constant	Value	Description
vbNormal	0	Normal
VbReadOnly	1	Read-only
VbHidden	2	Hidden
VbSystem	4	System
VbDirectory	16	Directory or folder
VbArchive	32	File has changed since last backup

Remarks

To determine which attributes are set, use the And operator to perform a bitwise comparison of the value returned by the GetAttr method and the value of the individual file attribute you want. If the result is not zero, that attribute is set for the named file.

3.4.1.11 Input (Panels, Comfort Panels)**Function**

This method returns a string containing characters from a file opened in Input or Binary mode.

Syntax

```
file.Input(number)
```

Parameters**File**

Reference to a File control.

Number

Any valid numeric expression that specifies the number of characters to return.

Return Values

String containing characters read from file.

Remarks

Data read with the Input method usually is written to a file with the LinePrint or Put functions. Use this method only with files opened in Input or Binary mode.

Unlike the LineInputString method, the Input method returns all the characters it reads, including commas, carriage returns, line feeds, quotation marks, and leading spaces.

With files opened for Binary access, an attempt to read through the file using the Input method until the EOF function returns True generates an error. To avoid an error, use the LOF and Loc functions instead of EOF when reading binary files with the Input method or use Get when using the EOF function.

3.4.1.12 InputFields (Panels, Comfort Panels)

Function

This method reads data from an open sequential file and returns a single dimension Variant array.

Syntax

```
file.InputFields(number)
```

Parameters

File

Reference to a File control.

Number

Number of comma-delimited fields to read from the file.

Return Values

Array containing the fields read from the file.

Remarks

Data read with the InputFields method usually is written to a file with WriteFields. Use this method only with files opened in Input or Binary mode.

InputFields reads standard string or numeric data without modification. The following table shows how InputFields reads other input data.

Data	Value Assigned to Variable
Delimiting comma or blank line	Empty
#NULL#	Null

Data	Value Assigned to Variable
#TRUE# or #FALSE#	True or False
#yyyy-mm-dd hh:mm:ss#	The date and/or time represented by the expression

Double quotation marks ("") within input data are discarded.

If you reach the end of the file while you are inputting a data item, the input is terminated and an error occurs.

To correctly read data from a file into variables using InputFields, use the WriteFields method instead of the LinePrint method to write the data to the files. Using WriteFields ensures each separate data field is properly delimited.

3.4.1.13 InputB (Panels, Comfort Panels)

Function

This method returns bytes from a file opened in Input or Binary mode.

Syntax

```
file.InputB(number)
```

Parameters

File

Reference to a File control.

Number

Any valid numeric expression that specifies the number of bytes to return.

Return Values

Array containing bytes read from file.

Remarks

Data read with the InputB method usually is written to a file with the LinePrint or Put functions. Use this method only with files opened in Input or Binary mode.

3.4.1.14 Kill (Panels, Comfort Panels)

Function

This method deletes files from a disk.

Syntax

filesystem.Kill pathname

Parameters

Filesystem

Reference to a FileSystem control.

Pathname

Required. String expression that specifies one or more file names to be deleted. The pathname can include the directory or folder.

Return Values

None.

Remarks

The Kill method supports the use of multiple-character (*) and single-character (?) wildcards to specify multiple files.

An error occurs if you try to use Kill to delete an open file.

3.4.1.15 LineInputString (Panels, Comfort Panels)

Function

This method reads a single line from an open sequential file and assigns it to a string variable.

Syntax

file.LineInputString

Parameters

File

Reference to a File control.

Return Values

None.

Remarks

Data read with LineInputString usually is written from a file with LinePrint. The LineInputString method reads from a file one character at a time until it encounters a carriage return (Chr(13)) or carriage return/line feed (Chr(13) + Chr(10)) sequence. Carriage return/line feed sequences are skipped rather than appended to the character string.

3.4.1.16 LinePrint (Panels, Comfort Panels)

Function

This method writes a single line to an open sequential file.

Syntax

file.LinePrint output

Parameters

File

Reference to a File control.

Output

String expression to write to a file.

Return Values

None.

Remarks

Data written with LinePrint is usually read from a file with LineInputString. A carriage return/line feed (Chr(13) + Chr(10)) sequence is appended to the end of the string.

3.4.1.17 Loc (Panels, Comfort Panels)

Function

This property returns a number specifying the current read/write position.

Syntax

file.Loc

Parameters

File

Reference to a File control.

Remarks

For files opened with the fsModeRandom file mode, Loc returns the number of the last record read or written. For files opened with all other modes, Loc returns the position of the last byte read or written.

3.4.1.18 LOF (Panels, Comfort Panels)

Function

This property returns a number representing the size, in bytes, of a file.

Syntax

file.LOF

Parameters

File

Reference to a File control.

Remarks

The LOF property can be used with the Loc property to guarantee that a read operation does not continue past the end of a file.

3.4.1.19 Mkdir (Panels, Comfort Panels)

Function

This method creates a new directory.

Syntax

filesystem.Mkdir PathName

Parameters

Filesystem

Reference to a FileSystem control.

Pathname

String expression that contains the directory name.

Return Values

None.

Remarks

MkDir generates an error if the directory already exists.

3.4.1.20 MoveFile (Panels, Comfort Panels)

Function

This method renames an existing file or a directory, including all its subdirectories.

Syntax

```
filesystem.MoveFile PathName, NewPathName
```

Parameters

Filesystem

Reference to a FileSystem control.

PathName

String that contains the file name.

NewPathName

String that contains the file name to copy to.

Return Values

None.

3.4.1.21 Open (Panels, Comfort Panels)

Function

This method opens a file in either the Input (1), Output (2), Random (4), Append (8), or Binary mode (32).

Syntax

file.Open pathname, mode, [access], [lock], [reclength]

Parameters

File

Reference to a File control.

Pathname

String expression that specifies a file name.

Mode

Specifies the file mode: Input (1), Output (2), Random (4) , Append (8), or Binary (32).

Access

Operation permitted on the open file: Read, Write, or ReadWrite [Default]. (1, 2, 3)

Lock

Operations permitted on the open file by other processes: Shared, LockRead, LockWrite [Default], and LockReadWrite. (1, 2, 3, 0)

Reclength

Number, in bytes, that is less than 32,767. For files opened for random access, this value is the record length. For sequential files, this value is the number of characters buffered.

Return Values

None.

Remarks

The reclength parameter is ignored if the mode is Binary. When opening a file in Random mode, you must specify a record size of greater than zero or an error will occur.

3.4.1.22 Put (Panels, Comfort Panels)

Function

This method writes data from a variable to a disk file.

Syntax

file.Put data, [recnumber]

Parameters

Data

Required. Variant variable that contains data to be written to disk.

Recnumber

Optional. Variant (Long). Record number (Random mode files) or byte number (Binary mode files) at which writing begins.

Return Values

None.

Remarks

Data written with Put usually is read from a file with Get.

The first record or byte in a file is at position 1, the second record or byte is at position 2, and so on. If you omit recnumber, the next record or byte after the last Get or Put method or pointed to by the last Seek function is written.

For files opened in Random mode, the following rules apply:

- If the length of the data being written is less than the length specified in the Len clause of the Open method, Put writes subsequent records on record-length boundaries. The space between the end of one record and the beginning of the next record is padded with the existing contents of the file buffer. Because the amount of padding data cannot be determined with any certainty, it generally is a good idea to have the record length match the length of the data being written. If the length of the data being written is greater than the length specified in the Len clause of the Open method, an error occurs.
- If the variable being written is a Variant of a numeric type, Put writes 2 bytes identifying the VarType of the Variant and then writes the variable. For example, when writing a Variant of VarType 3, Put writes 6 bytes: 2 bytes identifying the Variant as VarType 3 (Long) and 4 bytes containing the Long data. The record length specified by the Len clause in the Open method must be at least 2 bytes greater than the actual number of bytes required to store the variable.

You can use the Put method to write a Variant array to disk, but you cannot use Put to write a scalar Variant containing an array to disk. You also cannot use Put to write objects to disk. If the variable being written is a Variant of VarType 8 (String), Put writes 2 bytes identifying

3.4 VBScript für Windows CE (Panels, Comfort Panels)

the VarType and 2 bytes indicating the length of the string. It then writes the string data. The record length specified by the Len clause in the Open method must be at least 4 bytes greater than the actual length of the string.

If the variable being written is a dynamic array, Put writes a descriptor whose length equals 2 plus 8 times the number of dimensions, that is, $2 + 8 * \text{NumberOfDimensions}$. The record length specified by the Len clause in the Open method must be greater than or equal to the sum of all the bytes required to write the array data and the array descriptor. For example, the following array declaration requires 118 bytes when the array is written to disk.

For files opened in Binary mode, the Len clause in the Open method has no effect. Put writes all variables to disk contiguously; that is, with no padding between records.

3.4.1.23 Rmdir (Panels, Comfort Panels)

Function

This method deletes an existing empty directory.

Syntax

filesystem.Rmdir PathName

Parameters

Filesystem

Reference to a FileSystem control.

PathName

String that contains the directory name.

Return Values

None.

Remarks

The directory must be empty before it can be removed. You must specify a complete file path.

3.4.1.24 Seek (Panels, Comfort Panels)

Function

This property returns and sets the next position in a file that will be read or written.

Syntax

file.Seek [= position]

Parameters

File

Reference to a File control.

Position

Numeric expression that specifies a position within a file.

Remarks

The Seek property specifies the next file position, whereas the Loc property specifies the current position. Seek always will be one more than Loc, except when a file is first opened and Seek and Loc are both 1.

Negative Seek or 0 causes an error.

3.4.1.25 SetAttr (Panels, Comfort Panels)

Function

This method sets attribute data for a file.

Syntax

filesystem.SetAttr pathname, attributes

Parameters

Filesystem

Reference to a FileSystem control.

Pathname

Required. String expression that specifies a file name.

Attributes

Required. Numeric expression whose sum specifies file attributes. The following table shows the parameter settings of attributes.

Constant	Value	Description
vbNormal	0	Normal (default)
vbReadOnly	1	Read-only
vbHidden	2	Hidden
VbSystem	4	System file
VbArchive	32	File has changed since last backup

Return Values

None.

Remarks

A run-time error occurs if you try to set the attributes of an open file.

3.4.1.26 WriteFields (Panels, Comfort Panels)

Function

This method writes data to a sequential file.

Syntax

file.WriteFields [data]

Parameters

File

Reference to a File control.

Data

Variant or Variant array of numeric or string expressions to write to a file.

Prinzip

Return Values

None.

Remarks

Data written with WriteFields is usually read from a file with InputFields. If you omit data, a blank line is printed to the file.

- Numeric data is always written using the period as the decimal separator.
- For Boolean data, either #TRUE# or #FALSE# is printed. The True and False keywords are not translated, regardless of locale.
- Date data is written to the file using the universal date format. When either the date or the time component is missing or is zero, only the component provided gets written to the file.
- Nothing is written to the file if Data is Empty. However, for Null data, #NULL# is written.
- If data is Null, #NULL# is written to the file.

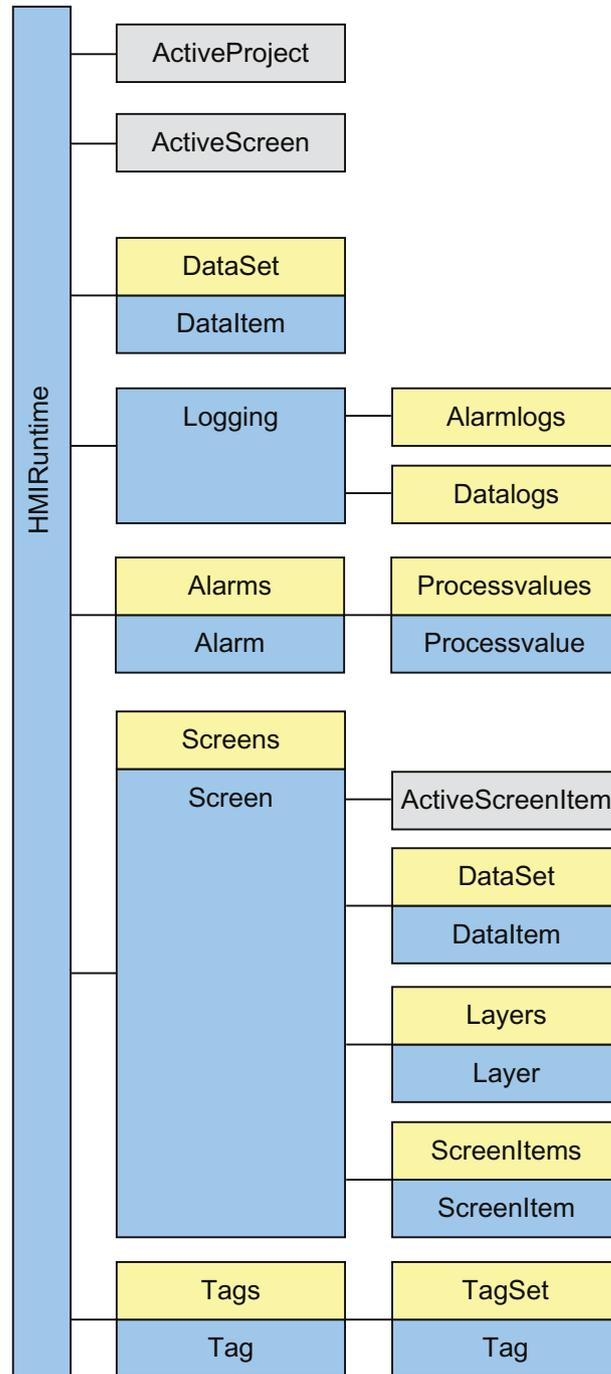
The WriteFields method inserts commas between items and quotation marks around strings as they are written to the file. You do not have to put explicit delimiters in the list. WriteFields inserts a newline character—that is, a carriage return/line feed (Chr(13) + Chr(10))—after it has written the final character in data to the file.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

3.5.1 VBS Objektmodell (RT Professional)

Das VBS-Objektmodell in WinCC

Das folgende Bild zeigt das VBS-Objektmodell in WinCC:



Über das WinCC Objektmodell des grafischen Runtime-Systems greifen Sie auf Objekte und Variablen in Runtime zu.

Objekte

Über die Objekte und Auflistungen erhalten Sie Zugriff auf alle Objekte des grafischen Runtime-Systems:

- Anzeige- und Bedienobjekte
- Bilder
- Ebenen
- Variablen

Eigenschaften

Über die Eigenschaften der einzelnen Objekte verändern Sie gezielt Anzeige- und Bedienobjekte und Variablen in Runtime. Beispielsweise schalten Sie mit Klicken eine Schaltfläche frei oder lösen bei Änderung eines Variablenwerts einen Farbumschlag aus.

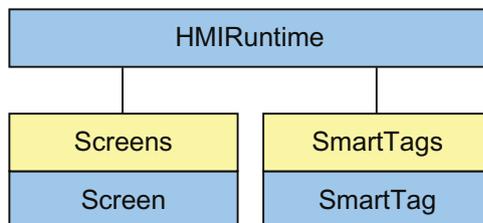
Methoden

Mit den Methoden, die Sie auf die einzelnen Objekte anwenden, lesen Sie z. B. Variablenwerte zur weiteren Verarbeitung aus oder geben Meldungen in Runtime aus.

3.5.2 Objekte (Panels, Comfort Panels, RT Advanced)

3.5.2.1 HMIRuntime (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das grafische Runtime-System dar.

Das HMIRuntime-Objekt enthält Eigenschaften und Methoden, die Objekte der Hauptebene zurückgeben, z. B. gibt die ActiveScreen-Eigenschaft ein Screen-Objekt zurück.

Verwendung

Sie verwenden das Objekt "HMIRuntime" z. B. wie folgt:

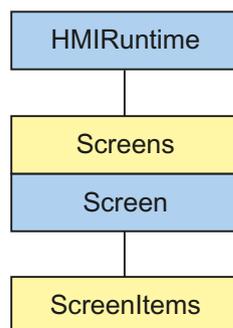
- Die aktuelle Runtime-Sprache lesen oder setzen (Eigenschaft "Language")
- Den Namen des aktuellen Grundbildes lesen oder über das Setzen eines neuen Bildnamens einen Grundbildwechsel auslösen (Eigenschaft "BaseScreenName")
- Auf Variablen zugreifen (Auflistung "SmartTags")
- Runtime beenden (Methode "Stop")
- Ausgaben zur Ablaufverfolgung ausgeben (Methode "Trace")
- Die in Runtime angezeigten Bilder ansprechen (Auflistung "Screens")

Siehe auch

ActiveScreen (Seite 1548)

3.5.2.2 Screens (Auflistung) (Panels, Comfort Panels, RT Advanced)

Beschreibung



Eine Auflistung von Screen-Objekten.

Die Auflistung enthält die folgenden zwei Elemente:

- Das 1. Element mit dem Index 0 stellt das Permanentfenster dar.
- Das 2. Element mit dem Index 1 stellt das Grundbild dar.

Alternativ können Sie die beiden Elemente folgendermaßen über ihre Namen ansprechen:

- Permanentfenster: "Overview"
- Grundbild: Name des im Grundbild angezeigten Bildes

Wenn das benannte Bild nicht dargestellt wird, kommt es beim Zugriff zu einem Fehler.

Das Permanentfenster "Overview" wird in der Objektliste und bei der Autovervollständigung angezeigt

Hinweis

Das Meldefenster und der Meldeindikator sind nicht in der Screens-Auflistung enthalten, auch wenn sie in Runtime den Fokus haben.

Verwendung

Verwenden Sie die Screens-Eigenschaft, um die Screens-Auflistung zurückzugeben. Im folgenden Beispiel wird die Hintergrundfarbe von Schwarz auf Grün geändert:

Als Index verwenden Sie den Objektnamen.

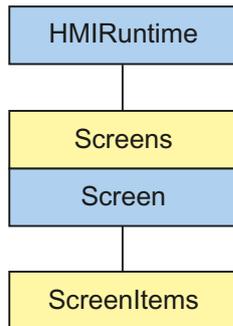
```
'VBS_Example_BackColor  
HMIRuntime.Screens("Rootscreen").BackColor = vbGreen
```

Hinweis

Wenn Sie einen Bildwechsel ausführen, werden die geöffneten Referenzen auf das nicht mehr vorhandene Bild ungültig. Sie können mit diesen Referenzen dann nicht mehr arbeiten.

3.5.2.3 Screen (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Prozessbild dar, das momentan am Bediengerät angezeigt wird, oder das Permanentfenster in Runtime. Das Screen-Objekt wird Ihnen als Ergebnis des Zugriffs auf die Screens-Auflistung zurückgegeben.

Des Weiteren enthält das Screen-Objekt eine Auflistung aller in dem angesprochenen Bild enthaltenen Grafikobjekte, die über die Auflistung "ScreenItems" angesprochen werden können.

Verwendung

Verwenden Sie das Screen-Objekt, um z.B.:

- Die Breite und Höhe eines Bildes zu lesen (Eigenschaften "Width" und "Height").
- Die Hintergrundfarbe zu ändern (Eigenschaften "BackColor").

Als Index verwenden Sie den Objektnamen.

Im folgenden Beispiel wird die Hintergrundfarbe von Schwarz auf Grün geändert:

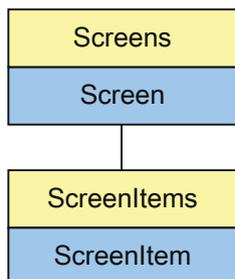
```
'VBS_Example_BackColor  
HMIRuntime.Screens("Rootscreen").BackColor = vbGreen
```

Siehe auch

ScreenItem (Seite 1298)

3.5.2.4 ScreenItem (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt ein Objekt im angegebenen Bild dar. Das ScreenItem-Objekt ist Element der ScreenItems-Auflistung.

Verwendung

Über das ScreenItem-Objekt können Sie abhängig von bestimmten Ereignissen auf die Eigenschaften von Grafikobjekten innerhalb eines Bildes zugreifen.

Sie verwenden das Objekt "ScreenItem" z. B. folgendermaßen:

- Eigenschaft "Visible"
Die Sichtbarkeit eines Objekts einschalten oder ausschalten
- Eigenschaften "Height" und "Width"
Die Breite und Höhe eines Objekts abfragen
- Eigenschaften "Top" und "Left"
Die Position eines Objekts verändern
- Eigenschaft "ObjectName"
Den Namen eines Grafikobjekts auslesen
- Eigenschaft "Parent"
Eine Referenz auf das übergeordnete Bild setzen

Verwenden Sie die ScreenItems-Eigenschaft, um ein Objekt im Bild zurückzugeben. Als Index verwenden Sie den Objektnamen.

Beispiel

Im folgenden Beispiel wird die Hintergrundfarbe des Kreises "myCircle" im Bild "RootScreen" auf grün gesetzt.

```
'VBS_Example_ScreenItems  
  
Dim objCircle  
Set objCircle = HMIRuntime.Screens("RootScreen").ScreenItems("myCircle")  
objCircle.BackColor = vbGreen
```

Hinweis

Um Speicherplatz auf dem Bediengerät zu sparen, werden beim Laden des Projekts keine Objektnamen übertragen. Wenn Sie dennoch Objektnamen übertragen wollen, rufen Sie in WinCC die Runtime-Einstellungen für das entsprechende Bediengerät auf. Unter "Allgemein" können Sie die Einstellung ändern. Die Objektnamen benötigen Sie, wenn Sie die Objekte über den Objektnamen ansprechen wollen, oder zum Debuggen eines Projekts.

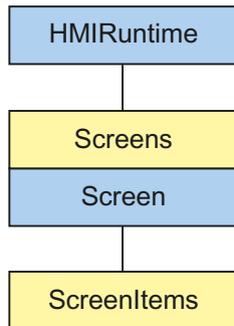
Das Objekt "ScreenItem" verfügt je nach Ausprägung über unterschiedliche Eigenschaften. Folgende Eigenschaften finden Sie an jedem Objekt vom Typ "ScreenItem":

- Enabled
- Height
- Left
- ObjectName
- Parent
- Top
- Type
- Visible
- Width

Wenn Sie einen bestimmten Objekttyp ansprechen, kommen weitere Eigenschaften zu den Standardeigenschaften hinzu. Die zusätzlichen Eigenschaften finden Sie bei der Beschreibung der einzelnen Objekttypen.

3.5.2.5 ScreenItems (Panels, Comfort Panels, RT Advanced)

Beschreibung



Eine Auflistung von ScreenItem-Objekten mit allen Bildobjekten des angegebenen Prozessbildes. Die Auflistung besitzt eine Eigenschaft Parent: Über diese Eigenschaft erhalten Sie eine Referenz auf das Prozessbild, in dem das Bildobjekt liegt.

Verwendung

Sie können über die Auflistung "ScreenItems"

- Alle Objekte innerhalb der Auflistung (also alle Objekte innerhalb eines Bildes) ausgeben oder bearbeiten
- Die Objekte eines Bildes aufzählen (Eigenschaft "Count").
- Ein bestimmtes Objekt der Auflistung bearbeiten (Methode "Item").

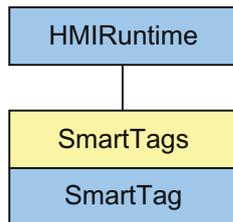
Verwenden Sie die ScreenItems-Eigenschaft, um ein Objekt aus dem Prozessbild zurückzugeben. Als Index verwenden Sie den Objektnamen.

Im folgenden Beispiel wird die Hintergrundfarbe des Kreises "myCircle" im Bild "RootScreen" auf grün gesetzt.

```
Dim objCircle
Set objCircle = HMIRuntime.Screens("RootScreen").ScreenItems("myCircle")
objCircle.BackColor = vbGreen
```

3.5.2.6 SmartTags (Panels, Comfort Panels, RT Advanced)

Beschreibung



Eine Auflistung von SmartTag-Objekten, die alle Variablen in WinCC Runtime darstellen.

Hinweis

Die SmartTags-Auflistung hat einen eingeschränkten Funktionsumfang: Sie können nur den Variablennamen verwenden, um auf ein SmartTag-Objekt zuzugreifen. Ein Zugriff über den Index oder die Verwendung der For-Each-Anweisung wird nicht unterstützt.

Hinweis

Wenn Sie mit der SmartTags-Auflistung auf eine Variable zugreifen wollen, die im Projekt noch nicht angelegt ist, wird kein Wert zurückgeliefert. Eine Zuweisung an eine nicht vorhandene Variable wird nicht ausgeführt:

```
Dim intVar  
intVar = SmartTags("FillLevel")
```

"intVar" bleibt leer, wenn die Variable "FillLevel" nicht angelegt ist.

Verwendung

Verwenden Sie die SmartTags-Auflistung, um ein SmartTag-Objekt zurückzugeben. Verwenden Sie den Variablennamen, um das SmartTag-Objekt zu referenzieren:

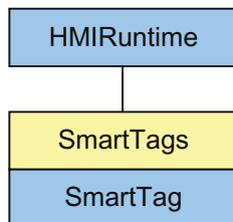
```
'VBS_Example_SmartTags
'Writes tag value to local tag and returns a user-defined text through the
operating system channel for debug alarms.
Dim strAirPressure
strAirPressure = "Current air pressure: " + CStr(SmartTags("AirPressure"))
HMIRuntime.Trace strAirPressure
```

In Runtime Advanced und Panels sprechen Sie die Variable direkt über ihren Namen an. Wenn der Variablenname den VBS-Namenskonventionen entspricht, brauchen Sie daher die SmartTags-Auflistung nicht zu verwenden. Beachten Sie folgendes Beispiel.

```
Dim strAirPressure
strAirPressure = "Current air pressure: " + CStr(AirPressure)
HMIRuntime.Trace strAirPressure
```

3.5.2.7 SmartTag (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt den Wert der angegebenen Prozessvariablen dar. Das SmartTag-Objekt ist Element der SmartTags-Auflistung.

Verwendung

Das SmartTag-Objekt ermöglicht es Ihnen, auf den Wert der angegebenen Prozessvariablen lesend und schreibend zuzugreifen. Das SmartTag-Objekt gibt keine Objektreferenz zurück. Verwenden Sie die SmartTags-Auflistung, um den Wert einer Prozessvariablen zurückzugeben. Als Index verwenden Sie den Variablennamen.

Hinweis

Mit der Einstellung "SmartTags liest Werte aus dem Cache" werden die Werte nicht direkt aus der Steuerung, sondern aus dem Prozessabbild (Cache) gelesen.

Das SmartTag-Objekt kann Werte auch direkt aus der Steuerung lesen. Dabei ist jedoch eine deutlich höhere Kommunikationslast zwischen HMI-Gerät und Steuerung zu erwarten.

Beispiel

```
'VBS_Example_SmartTags
'Writes tag value to local tag and returns a user-defined text through the
operating system channel for debug alarms.
Dim strAirPressure
strAirPressure = "Current air pressure: " + CStr(SmartTags("AirPressure"))
HMIRuntime.Trace strAirPressure
```

Hinweis

Wenn Sie mit der SmartTags-Auflistung auf eine Variable zugreifen wollen, die im Projekt noch nicht angelegt ist, wird kein Wert zurückgeliefert. Eine Zuweisung an eine nicht vorhandene Variable wird nicht ausgeführt:

```
Dim intVar
intVar = SmartTags("FillLevel")
```

"intVar" bleibt leer, wenn die Variable "FillLevel" nicht angelegt ist.

Hinweis

Wenn Sie mit der VBS-Funktion "TypeName" den Datentyp eines SmartTag-Objekts zurückgeben wollen, verwenden Sie folgende Syntax:

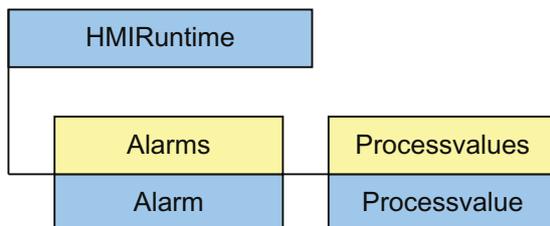
```
TypeName(SmartTags("FillLevel").value)
```

Um auf den Wert eines Arrayelements zuzugreifen, verwenden Sie "SmartTags("<Variable>"(Index)". Setzen Sie für "Index" die Nummer des gewünschten Arrayelements ein, z. B. "SmartTags("AirPressure")(2)".

3.5.3 Objekte (RT Professional)

3.5.3.1 Alarm (RT Professional)

Beschreibung



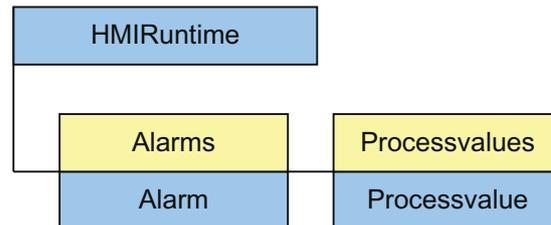
Das Alarm-Objekt wird verwendet um auf die Alarms-Objekt-Auflistung zuzugreifen.

Hinweis

Die Eigenschaften des Alarm-Objekts werden nicht automatisch aktualisiert, wenn sich die Werte der Eigenschaften ändern.

3.5.3.2 Alarms (Auflistung) (RT Professional)

Beschreibung



Mit dem Alarm-Objekt können Sie vorhandene Meldungen auslösen.

Verwendung

Sie können über die Auflistung "Alarms":

- Auf eine Meldung in der Auflistung zugreifen (Item-Methode)
- Ein neues Alarm-Objekt erzeugen (Create-Methode)
- Die AlarmID der Meldung auslesen (AlarmID-Eigenschaft)
- Den Status einer Meldung auslesen (State-Eigenschaft)
- Den Zeitstempel der Meldung auslesen (Timestamp-Eigenschaft)
- Eine Instanz des Alarm-Objekts erzeugen (Instance-Eigenschaft)
- Den Namen des Computers, auf dem die Meldung gekommen ist, auslesen (ComputerName-Eigenschaft)
- Den Namen des Benutzers, der die Meldung ausgelöst hat, auslesen oder setzen (UserName-Eigenschaft)
- Die Namen der Prozesswertblöcke lesen oder setzen (ProcessValues-Eigenschaft)
- Den Kommentar der Meldung lesen oder setzen (Comment-Eigenschaft)
- Das Serverpräfix der Meldung lesen oder setzen (Context-Eigenschaft)

Beispiel

Im folgenden Beispiel wird die im Editor "HMI-Meldungen" projektierte Meldung mit der Meldenummer "1" ausgelöst:

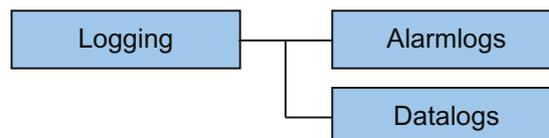
```
'VBS360
Dim MyAlarm
Set MyAlarm = HMIRuntime.Alarms(1)
MyAlarm.State = 5 'hmiAlarmStateCome + hmiAlarmStateComment
MyAlarm.Comment = "MyComment"
MyAlarm.UserName = "Hans-Peter"
MyAlarm.ProcessValues(1) = "Process Value 1"
MyAlarm.ProcessValues(4) = "Process Value 4"
MyAlarm.Create "MyApplication"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

Siehe auch

AlarmID (Seite 1556)

3.5.3.3 AlarmLogs (Auflistung) (RT Professional)

Beschreibung



Mit dem Objekt verbinden Sie ausgelagerte Archivsegmente des Meldearchivs wieder mit der Runtime oder löschen zuvor eingelagerte Archivsegmente des Meldearchivs wieder. Die einzulagernden Archivsegmente werden in den Ordner "Common Archiving" des WinCC-Projekts kopiert. Die zuvor eingelagerten Archivsegmente werden im Ordner "Common Archiving" gelöscht.

Über Parameter steuern Sie, von welchem Ort Archivsegmente eingelagert werden. Sie legen den Zeitraum fest, über den Archivsegmente eingelagert oder gelöscht werden.

Wenn bei der Operation mit den Archivsegmenten ein Fehler auftritt, gibt die eingesetzte Methode eine Fehlermeldung zurück.

Verwendung

- Methode "Restore"
Zuvor ausgelagerte Archivsegmente des Meldearchivs werden mit der Runtime verbunden.
- Methode "Remove"
Zuvor eingelagerte Archivsegmente des Meldearchivs werden aus dem Runtime-Projekt gelöscht.

Beispiel

Im folgenden Beispiel werden Archivsegmente des Meldearchivs eingelagert und der Rückgabewert als Trace ausgegeben.

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

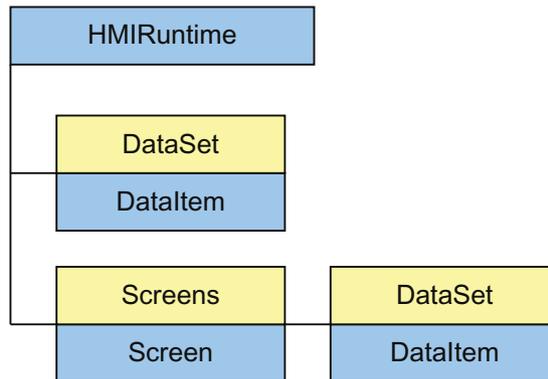
Siehe auch

[Restore \(Seite 2248\)](#)

[Remove \(Seite 2243\)](#)

3.5.3.4 Dataltem (RT Professional)

Beschreibung



Sie nutzen das **Dataltem**-Objekt, um auf die Inhalte der **DataSet**-Auflistung zuzugreifen. Werte oder Objektreferenzen werden in der Auflistung als **Dataltem** abgelegt.

Der Zugriff erfolgt über den Namen, unter dem der Wert in die Auflistung hinzugefügt wurde. Ein Einzelzugriff über **Index** ist nicht empfehlenswert, da sich der **Index** beim Hinzufügen und Löschen von Werten ändert. Sie verwenden den **Index**, um den kompletten Inhalt der Auflistung auszugeben. Die Ausgabe erfolgt in alphabetischer Reihenfolge.

Hinweis

Stellen Sie bei Objektreferenzen sicher, dass die Objekte multithreadfähig sind.

Beispiel

Das Beispiel zeigt, wie der Wert von 'Motor1' als Trace ausgegeben wird.

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

Das folgende Beispiel enumeriert alle Dataltem-Objekte der DataSet-Auflistung. Name und Value werden als Trace ausgegeben.

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

Hinweis

Bei Objekten kann Value evtl. nicht direkt ausgegeben werden.

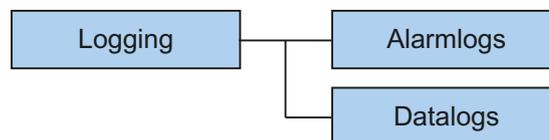
Siehe auch

[Value \(Seite 1968\)](#)

[Name \(Seite 1749\)](#)

3.5.3.5 DataLogs (Auflistung) (RT Professional)

Beschreibung



Mit dem Objekt verbinden Sie ausgelagerte Archivsegmente des Variablenarchivs wieder mit der Runtime oder löschen zuvor eingelagerte Archivsegmente des Variablenarchivs wieder. Die einzulagernden Archivsegmente werden in den Ordner "Common Archiving" des WinCC-

Projekts kopiert. Die zuvor eingelagerten Archivsegmente werden im Ordner "Common Archiving" gelöscht.

Über Parameter steuern Sie, von welchem Ort Archivsegmente eingelagert werden. Sie legen den Zeitraum fest, über den Archivsegmente eingelagert oder gelöscht werden. Zusätzlich stellen Sie den Typ des Archivs ein ("Variablenarchiv Fast", "Variablenarchiv Slow", "Variablenarchiv Fast und Variablenarchiv Slow").

Wenn bei der Operation mit den Archivsegmenten ein Fehler auftritt, gibt die eingesetzte Methode eine Fehlermeldung zurück.

Verwendung

- Methode "Restore"
Zuvor ausgelagerte Archivsegmente des Variablenarchivs werden mit der Runtime verbunden.
- Methode "Remove"
Zuvor eingelagerte Archivsegmente des Variablenarchivs werden aus dem Runtime-Projekt gelöscht.

Beispiel

In dem folgenden Beispiel werden Archivsegmente des Variablenarchiv Fast eingelagert und der Rückgabewert als Trace ausgegeben.

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

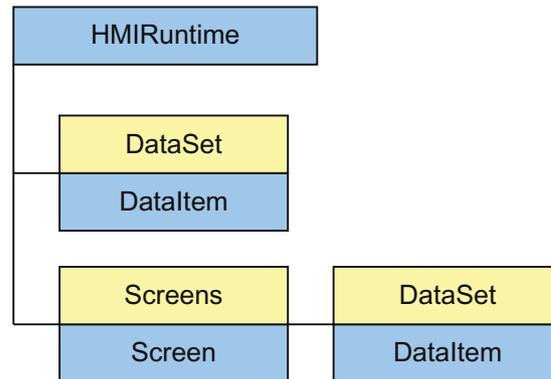
Siehe auch

[Restore \(Seite 2248\)](#)

[Remove \(Seite 2243\)](#)

3.5.3.6 DataSet (Auflistung) (RT Professional)

Beschreibung



Mit dem DataSet-Objekt tauschen Sie über mehrere Aktionen hinweg Daten aus.

Ein DataSet-Objekt ist global und am Screen-Objekt definiert. Auf die Daten können Sie aus jeder VBS-Aktion zugreifen.

Sie adressieren das am Screen-Objekt entsprechend der Bildhierarchie. Das DataSet-Objekt besteht, solange das Bild angezeigt wird. Das globale Objekt besteht über den gesamten Zeitraum der Runtime.

Der Zugriff erfolgt über das Dataltem-Objekt.

Hinweis

Sie können Objekte vom Typ Screen, Screens, ScreenItem, ScreenItems, Tag und TagSet nicht in die DataSet-Auflistung aufnehmen.

Das DataSet-Objekt unterstützt keine Klassen.

Verwendung

Sie können die Auflistung "DataSet" folgendermaßen verwenden:

- Enumerieren
Alle Objekte innerhalb der Auflistung ausgeben oder bearbeiten
- Eigenschaft "Count"
Die Anzahl der enthaltenen Elemente ausgeben

- Methode "Item"
Ein bestimmtes Objekt der Auflistung bearbeiten
- Methode "Add"
Ein Objekt zur Auflistung hinzufügen
- Methode "Remove"
Ein bestimmtes Objekt aus der Auflistung entfernen
- Methode "RemoveAll"
Alle Objekte aus der Auflistung entfernen

Der Zugriff auf die Elemente der Auflistung erfolgt folgendermaßen :

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

Bei einer bildspezifischen Auflistung erfolgt der Zugriff folgendermaßen :

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

In einem Bild greifen Sie folgendermaßen auf das DataSet-Objekt des Bildes zu:

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

Wenn beim Zugriff der angegebene Name nicht in der Auflistung existiert, wird VT_Empty zurückgeliefert und eine Exception ausgelöst.

Beispiel

Das Beispiel zeigt, wie Sie in verschiedenen Aktionen einen Wert in die Auflistung aufnehmen, den Wert auslesen und den Wert wieder entfernen.

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

Siehe auch

[RemoveAll \(Seite 2246\)](#)

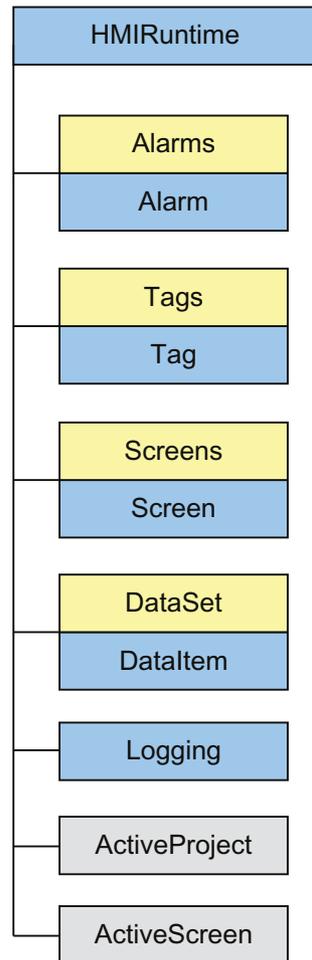
[Remove \(Seite 2243\)](#)

[Item \(Seite 2224\)](#)

[Add \(Seite 2162\)](#)

3.5.3.7 HMIRuntime (RT Professional)

Beschreibung



Das HMIRuntime-Objekt repräsentiert die grafische Runtime-Umgebung.

Verwendung

Sie verwenden das Objekt "HMIRuntime" z. B. folgendermaßen:

- Eigenschaft "Language"
Die aktuelle Runtime-Sprache lesen oder setzen
- Eigenschaft "BaseScreenName"
Den Namen des aktuellen Grundbildes lesen oder setzen
- Eigenschaft "ActiveProject"
Den Pfad des aktiven Runtime-Projekts lesen

- Eigenschaft "Tags"
Auf Variablen zugreifen
- Eigenschaft "DataSet"
Auf Variablen einer Auflistung zugreifen
- Methode "Stop"
Runtime beenden
- Methode "Trace"
Meldungen in einem Diagnosefenster ausgeben

Beispiel

Der folgende Befehl beendet WinCC Runtime:

```
'VBS3  
HMIRuntime.Stop
```

Siehe auch

[Trace \(Seite 2267\)](#)
[Stop \(Seite 2266\)](#)
[Language \(Seite 1703\)](#)
[Tags \(Seite 1857\)](#)
[Logging \(Seite 1725\)](#)
[DataSet \(Seite 1635\)](#)
[CurrentContext \(Seite 1631\)](#)
[MenuToolBarConfig \(Seite 1729\)](#)

3.5.3.8 Item (RT Professional)

Beschreibung

Das Item-Objekt liefert eine Referenz auf das aktuelle Objekt.

Verwendung

Sie verwenden das Item-Objekt, um z. B. die Eigenschaften des aktuell im Bild markierten Objekts anzusprechen.

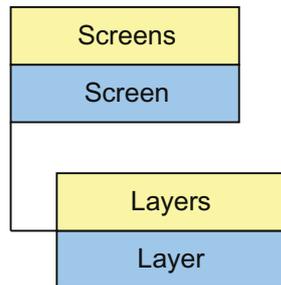
Beispiel

Im folgenden Beispiel setzen Sie die Hintergrundfarbe des Objekts auf Rot, das im Bild markiert ist:

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

3.5.3.9 Layer (RT Professional)

Beschreibung



Das Layer-Objekt wird als Ergebnis des Zugriffs auf die Layers-Auflistung zurückgegeben.

Parent-Objekt

Bild, in dem die Bildebene liegt

Verwendung

Über das Layer-Objekt greifen Sie abhängig von bestimmten Ereignissen auf die Eigenschaften einer kompletten Ebene zu, z. B. um abhängig von der Bedienberechtigung eine Ebene mit Bedienelementen aus- oder einzublenden.

Sie verwenden das Objekt "Layer" folgendermaßen:

- Eigenschaft "Visible"
Die Sichtbarkeit einer Ebene ein- oder ausschalten
- Eigenschaft "Name"
Den Namen einer Ebene auslesen

Hinweis

Die Layer-Eigenschaft gibt die Ebene aus, in der sich das Objekt befindet. Die Ebene "0" wird als Ebene "0" ausgegeben.

Beim Zugriff werden in VBS die Ebenen von 1 an gezählt. Deshalb sprechen Sie die Ebene "1" mit "Layers(2)" an.

Beispiel

Im folgenden Beispiel wird die Ebene 1 auf "unsichtbar" gesetzt:

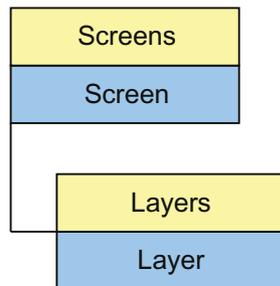
```
'VBS4  
Layers(2).Visible = vbFalse
```

Siehe auch

[Name \(Seite 1749\)](#)

3.5.3.10 Layers (Auflistung) (RT Professional)

Beschreibung



Mit der Layers-Auflistung greifen Sie auf alle 32 Ebenen des grafischen Runtime-Systems zu.

Parent-Objekt

Bild, in dem die Bildebene liegt

Verwendung

Sie verwenden die Auflistung "Layers" folgendermaßen:

- Eigenschaft "_NewEnum"
Alle Ebenen innerhalb der Auflistung bearbeiten
- Eigenschaft "Count"
Alle in der Auflistung enthaltenen Ebenen zählen
- Methode "Item"
Eine Ebene aus der Auflistung bearbeiten

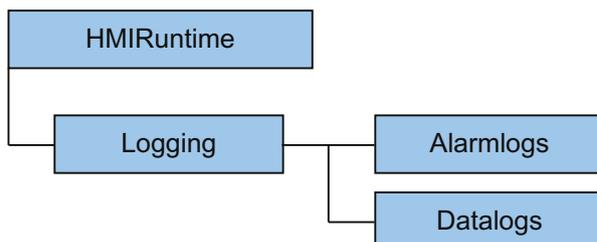
Die Eigenschaften sind Standard-Eigenschaften und Methoden einer Auflistung und werden in der WinCC-Dokumentation nicht detailliert beschrieben.

Siehe auch

Item (Seite 2224)

3.5.3.11 Logging (RT Professional)

Beschreibung



Mit dem Objekt verbinden Sie ausgelagerte Archivsegmente wieder mit der Runtime oder löschen zuvor eingelagerte Archivsegmente wieder. Die einzulagernden Archivsegmente werden in den Ordner "Common Archiving" des WinCC-Projekts kopiert. Die zuvor eingelagerten Archivsegmente werden im Ordner "Common Archiving" gelöscht.

Über Parameter steuern Sie, von welchem Ort Archivsegmente eingelagert werden. Sie legen den Zeitraum fest, über den Archivsegmente eingelagert oder gelöscht werden.

Wenn bei der Operation mit den Archivsegmenten ein Fehler auftritt, gibt die eingesetzte Methode eine Fehlermeldung zurück.

Verwendung

- Methode "Restore"
Zuvor ausgelagerte Archivsegmente des Meldearchivs und des Variablenarchivs werden mit der Runtime verbunden.
- Methode "Remove"
Zuvor eingelagerte Archivsegmente des Meldearchivs und des Variablenarchivs werden aus dem Runtime-Projekt gelöscht.

Beispiel

In dem folgenden Beispiel werden Archivsegmente aus dem Meldearchiv und dem Variablenarchiv eingelagert und der Rückgabewert als Trace ausgegeben.

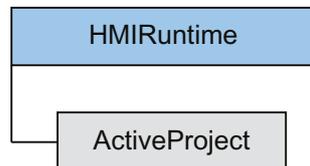
```
'VBS189  
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Restore("D:  
\Folder", "2004-09-14", "2004-09-20", -1) & vbNewLine
```

Siehe auch

Restore (Seite 2248)
Remove (Seite 2243)
DataLogs (Auflistung) (Seite 1309)
AlarmLogs (Auflistung) (Seite 1306)

3.5.3.12 Project (RT Professional)

Beschreibung



Mit dem Objekt fragen Sie Informationen des aktuellen Runtime-Projekts ab.
Das Project-Objekt wird als Ergebnis von ActiveProject zurückgegeben.

Verwendung

Sie können über das Objekt "Project" Folgendes lesen:

- den Pfad des aktuellen Runtime-Projekts (Eigenschaft "Path")
- den Namen des aktuellen Runtime-Projekts, ohne Pfadangabe und Datei-Erweiterung (Eigenschaft "Name")

Beispiel

Das folgende Beispiel gibt den Namen und den Pfad des aktuellen Runtime-Projekts als Trace aus:

```
'VBS159  
HMIRuntime.Trace "Name: " & HMIRuntime.ActiveProject.Name & vbNewLine  
HMIRuntime.Trace "Path: " & HMIRuntime.ActiveProject.Path & vbNewLine
```

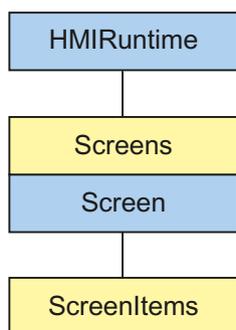
Siehe auch

Name (Seite 1749)

Path (Seite 1773)

3.5.3.13 Screen (RT Professional)

Beschreibung



Stellt das Bild dar, das momentan am Bediengerät angezeigt wird, oder das Permanentfenster in Runtime. Das Screen-Objekt wird als Ergebnis des Zugriffs auf die Screens-Auflistung zurückgegeben.

Das Screen-Objekt enthält zusätzlich folgende Auflistungen:

- Alle in dem angesprochenen Bild enthaltenen Grafikobjekte, die Sie über die Auflistung "ScreenItems" ansprechen können.
- Alle in dem angesprochenen Bild enthaltenen Ebenen, die Sie über das Objekt "Layers" ansprechen können.

Verwendung

Verwenden Sie das Screen-Objekt z. B. für folgende Aktionen:

- Eigenschaften "Width" und "Height"
Breite und Höhe eines Bildes lesen
- Eigenschaft "BackColor"
Hintergrundfarbe ändern

Als Index verwenden Sie den Objektnamen.

Beispiel

Im folgenden Beispiel wird die Hintergrundfarbe von Schwarz auf Grün geändert:

```
'VBS_Example_BackColor  
HMIRuntime.Screens("Rootscreen").BackColor = vbGreen
```

Parent-Objekt

Bildfenster, in dem das Screen-Objekt eingebettet ist.

Wenn das Screen-Objekt das Grundbild ist, ist das Parent-Objekt nicht definiert und auf Null gesetzt.

Hinweis

Wenn Sie einen Bildwechsel ausführen, werden alle geöffneten Referenzen auf nicht mehr vorhandene Bilder ungültig. Sie können mit diesen Referenzen dann nicht mehr arbeiten.

Beispiel

Im folgenden Beispiel wird die Breite des ersten Bildes in Runtime um 20 Pixel erhöht:

```
'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
```

Hinweise zu CrossReference

Alle Bilder, die Sie mit der Standardformulierung adressieren, werden durch die CrossReference von WinCC automatisch erfasst und sind in den Bildeigenschaften aufgeführt

```
HMIRuntime.BaseScreenName = "Screenname"
```

Wenn Sie Bilder mit anderen Formulierungen in Ihrem Code aufrufen, machen Sie diese über folgende Sektion der CrossReference bekannt:

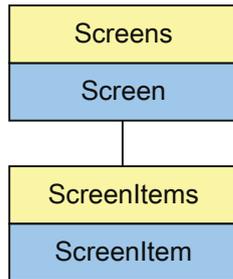
```
' WINCC:SCREENNAME_SECTION_START
Const ScreenNameInAction = "ScreenName"
' WINCC:SCREENNAME_SECTION_END
Sie können diese Sektion beliebig oft in VBS-Aktionen einfügen.
```

Siehe auch

- Refresh (Seite 2242)
- Activate (Seite 2157)
- ObjectSizeDeclutteringEnable (Seite 1755)
- ObjectSizeDeclutteringMax (Seite 1756)
- ObjectSizeDeclutteringMin (Seite 1757)
- LayerDeclutteringEnable (Seite 1710)
- Layers (Seite 1710)
- DataSet (Seite 1635)
- ExtendedZoomingEnable (Seite 1656)
- AccessPath (Seite 1547)

3.5.3.14 ScreenItem (RT Professional)

Beschreibung



Stellt ein Objekt im angegebenen Bild dar. Das ScreenItem-Objekt ist Element der ScreenItems-Auflistung.

Verwendung

Über das ScreenItem-Objekt können Sie abhängig von bestimmten Ereignissen auf die Eigenschaften von Grafikobjekten innerhalb eines Bildes zugreifen.

Sie verwenden das Objekt "ScreenItem" z. B. folgendermaßen:

- Eigenschaft "Visible"
Die Sichtbarkeit eines Objekts einschalten oder ausschalten
- Eigenschaften "Height" und "Width"
Die Breite und Höhe eines Objekts abfragen
- Eigenschaften "Top" und "Left"
Die Position eines Objekts verändern
- Eigenschaft "ObjectName"
Den Namen eines Grafikobjekts auslesen
- Eigenschaft "Parent"
Eine Referenz auf das übergeordnete Bild setzen

Verwenden Sie die ScreenItems-Eigenschaft, um ein Objekt im Bild zurückzugeben. Als Index verwenden Sie den Objektnamen.

Beispiel

Im folgenden Beispiel wird die Hintergrundfarbe des Kreises "myCircle" im Bild "RootScreen" auf grün gesetzt.

```
'VBS_Example_ScreenItems  
  
Dim objCircle  
Set objCircle = HMIRuntime.Screens("RootScreen").ScreenItems("myCircle")  
objCircle.BackColor = vbGreen
```

Hinweis

Um Speicherplatz auf dem Bediengerät zu sparen, werden beim Laden des Projekts keine Objektnamen übertragen. Wenn Sie dennoch Objektnamen übertragen wollen, rufen Sie in WinCC die Runtime-Einstellungen für das entsprechende Bediengerät auf. Unter "Allgemein" können Sie die Einstellung ändern. Die Objektnamen benötigen Sie, wenn Sie die Objekte über den Objektnamen ansprechen wollen, oder zum Debuggen eines Projekts.

Das Objekt "ScreenItem" verfügt je nach Ausprägung über unterschiedliche Eigenschaften. Folgende Eigenschaften finden Sie an jedem Objekt vom Typ "ScreenItem":

- Enabled
- Height
- Left
- ObjectName
- Parent
- Top
- Type
- Visible
- Width

Wenn Sie einen bestimmten Objekttyp ansprechen, kommen weitere Eigenschaften zu den Standardeigenschaften hinzu. Die zusätzlichen Eigenschaften finden Sie bei der Beschreibung der einzelnen Objekttypen.

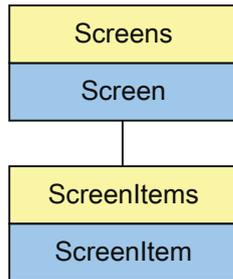
Siehe auch

[Activate \(Seite 2157\)](#)

[Layer \(Seite 1707\)](#)

3.5.3.15 ScreenItems (Auflistung) (RT Professional)

Beschreibung



Eine Auflistung von ScreenItem-Objekten mit allen Bildobjekten des angegebenen Bildes. Die Auflistung besitzt eine Eigenschaft "Parent". Über die Eigenschaft "Parent" erhalten Sie eine Referenz auf das Bild, in dem das Bildobjekt liegt.

Verwendung

Sie verwenden die Auflistung "ScreenItems" folgendermaßen:

- Alle Objekte innerhalb der Auflistung (also alle Objekte innerhalb eines Bildes) ausgeben oder bearbeiten
- Eigenschaft "Count"
Die Objekte eines Bildes aufzählen
- Methode "Item"
Ein bestimmtes Objekt der Auflistung bearbeiten

Verwenden Sie die ScreenItems-Eigenschaft, um ein Objekt aus dem Bild zurückzugeben. Als Index verwenden Sie den Objektnamen.

Beispiel

Im folgenden Beispiel wird die Hintergrundfarbe des Kreises "myCircle" im Bild "RootScreen" auf grün gesetzt.

```
Dim objCircle
Set objCircle = HMIRuntime.Screens("RootScreen").ScreenItems("myCircle")
objCircle.BackColor = vbGreen
```

Besonderheiten des ScreenItem-Objekts

Wenn Sie ein externes Control (ActiveX-Control oder OLE-Objekt) in WinCC einbetten, sind die Eigenschaften des eingebetteten Controls möglicherweise namensgleich mit den allgemeinen Eigenschaften des ScreenItem-Objekts. In diesem Fall haben die ScreenItem-Eigenschaften Vorrang.

Sie können die Eigenschaften des eingebetteten Controls aber über die Eigenschaft "object" ansprechen. Die Eigenschaft "object" gibt es nur bei ActiveX-Controls und OLE-Objekten.

Beispiel:

```
'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
```

Beispiel

Im folgenden Beispiel geben Sie die Namen der im aktuellen Bild enthaltenen Objekte in einer MessageBox aus:

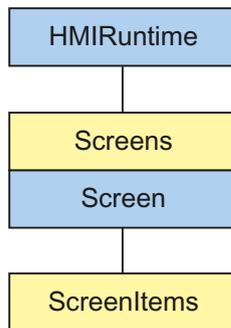
```
Sub OnClick(ByVal Item)
'VBS6
Dim lngAnswer
Dim lngIndex
lngIndex = 1
For lngIndex = 1 To ScreenItems.Count
lngAnswer = MsgBox(ScreenItems(lngIndex).Objectname, vbOKCancel)
If vbCancel = lngAnswer Then Exit For
Next
End Sub
```

Siehe auch

Item (Seite 2224)

3.5.3.16 Screens (Auflistung) (RT Professional)

Beschreibung



Mithilfe der Bildfenster-Technik können in WinCC Runtime mehrere Bilder gleichzeitig geöffnet sein, wobei aber nur ein Grundbild existiert. Die Auflistung "Screens" ermöglicht den Zugriff auf alle geöffneten Bilder in Runtime über den Bildnamen. Die Screens-Auflistung enthält auch alle unsichtbaren Bilder.

Der Zugriffsschlüssel, der im VBS-Umfeld in der Anweisung `HMIRuntime.Screens(<Zugriffsschlüssel>)` benötigt wird, muss folgender Syntaxbeschreibung entsprechen:

```
[<Grundbildname>.<Bildfenstername>[:<Bildname>] ...  
.<Bildfenstername>[:<Bildname>]
```

- Der Zugriffsschlüssel drückt die Bildhierarchie aus.
- Im Schlüssel können die Bildnamen an jeder Stelle weggelassen werden.
- Die "AccessPath"-Eigenschaft des "Screen"-Objekts entspricht dem vollständigen Zugriffsschlüssel.
- Das Grundbild ist über den Zugriffsschlüssel "" ansprechbar.

Beispiele

Die Bilder werden durch die Angabe der Hierarchie in der Auflistung adressiert. Sie können die Bilder mit oder ohne Verwendung des Bildnamens adressieren. In den folgenden Beispielen ist ein Grundbild "BaseScreenName" mit einem Bildfenster "ScreenWindow" projektiert. In dem Bildfenster ist ein Bild "ScreenName" enthalten.

Adressierung mit Verwendung des Bildnamens

```
'VBS8  
Set objScreen = HMIRuntime.Screens("BaseScreenName.ScreenWindow:ScreenName")
```

Adressierung ohne Verwendung des Bildnamens

```
'VBS9  
Set objScreen = HMIRuntime.Screens("ScreenWindow")
```

Referenzieren des Grundbildes auf verschiedene Arten

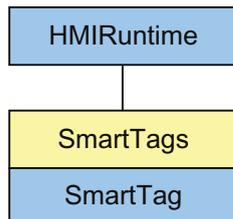
```
'VBS10  
Set objScreen = HMIRuntime.Screens(1)
```

```
'VBS11  
Set objScreen = HMIRuntime.Screens("")
```

```
'VBS12  
Set objScreen = HMIRuntime.Screens("BaseScreenName")
```

3.5.3.17 SmartTag (RT Professional)

Beschreibung



Stellt den Wert der angegebenen Prozessvariablen dar. Das SmartTag-Objekt ist Element der SmartTags-Auflistung.

Verwendung

Das SmartTag-Objekt ermöglicht es Ihnen, auf den Wert der angegebenen Prozessvariablen lesend und schreibend zuzugreifen. Das SmartTag-Objekt gibt keine Objektreferenz zurück. Verwenden Sie die SmartTags-Auflistung, um den Wert einer Prozessvariablen zurückzugeben. Als Index verwenden Sie den Variablennamen.

Hinweis

Mit der Einstellung "SmartTags liest Werte aus dem Cache" werden die Werte nicht direkt aus der Steuerung, sondern aus dem Prozessabbild (Cache) gelesen.

Das SmartTag-Objekt kann Werte auch direkt aus der Steuerung lesen. Dabei ist jedoch eine deutlich höhere Kommunikationslast zwischen HMI-Gerät und Steuerung zu erwarten.

Beispiel

```
'VBS_Example_SmartTags
'Writes tag value to local tag and returns a user-defined text through the
operating system channel for debug alarms.
Dim strAirPressure
strAirPressure = "Current air pressure: " + CStr(SmartTags("AirPressure"))
HMIRuntime.Trace strAirPressure
```

Hinweis

Wenn Sie mit der SmartTags-Auflistung auf eine Variable zugreifen wollen, die im Projekt noch nicht angelegt ist, wird kein Wert zurückgeliefert. Eine Zuweisung an eine nicht vorhandene Variable wird nicht ausgeführt:

```
Dim intVar
intVar = SmartTags("FillLevel")
```

"intVar" bleibt leer, wenn die Variable "FillLevel" nicht angelegt ist.

Hinweis

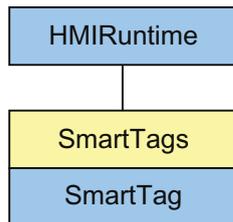
Wenn Sie mit der VBS-Funktion "TypeName" den Datentyp eines SmartTag-Objekts zurückgeben wollen, verwenden Sie folgende Syntax:

```
TypeName(SmartTags("FillLevel").value)
```

Um auf den Wert eines Arrayelements zuzugreifen, verwenden Sie "SmartTags("<Variable>") (Index)". Setzen Sie für "Index" die Nummer des gewünschten Arrayelements ein, z. B. "SmartTags("AirPressure")(2)".

3.5.3.18 SmartTags (RT Professional)

Beschreibung



Eine Auflistung von SmartTag-Objekten, die alle Variablen in WinCC Runtime darstellen.

Hinweis

Die SmartTags-Auflistung hat einen eingeschränkten Funktionsumfang: Sie können nur den Variablennamen verwenden, um auf ein SmartTag-Objekt zuzugreifen. Ein Zugriff über den Index oder die Verwendung der For-Each-Anweisung wird nicht unterstützt.

Hinweis

Wenn Sie mit der SmartTags-Auflistung auf eine Variable zugreifen wollen, die im Projekt noch nicht angelegt ist, wird kein Wert zurückgeliefert. Eine Zuweisung an eine nicht vorhandene Variable wird nicht ausgeführt:

```
Dim intVar  
intVar = SmartTags("FillLevel")
```

"intVar" bleibt leer, wenn die Variable "FillLevel" nicht angelegt ist.

Verwendung

Verwenden Sie die SmartTags-Auflistung, um ein SmartTag-Objekt zurückzugeben.
Verwenden Sie den Variablennamen, um das SmartTag-Objekt zu referenzieren:

```
'VBS_Example_SmartTags  
'Writes tag value to local tag and returns a user-defined text through the  
operating system channel for debug alarms.  
Dim strAirPressure  
strAirPressure = "Current air pressure: " + CStr(SmartTags("AirPressure"))  
HMIRuntime.Trace strAirPressure
```

In Runtime Advanced und Panels sprechen Sie die Variable direkt über ihren Namen an. Wenn der Variablenname den VBS-Namenskonventionen entspricht, brauchen Sie daher die SmartTags-Auflistung nicht zu verwenden. Beachten Sie folgendes Beispiel.

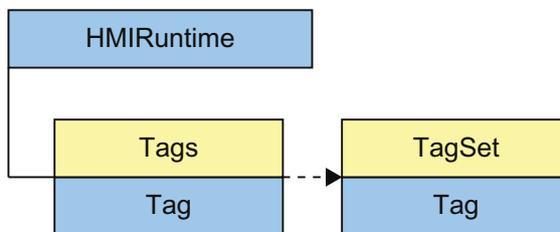
```
Dim strAirPressure  
strAirPressure = "Current air pressure: " + CStr(AirPressure)  
HMIRuntime.Trace strAirPressure
```

Siehe auch

SmartTag (Seite 1329)

3.5.3.19 Tag (RT Professional)

Beschreibung



Ein Tag-Objekt (Variable) wird über die Auflistung "Tags" zurückgegeben. Über das Tag-Objekt sind alle Eigenschaften und Methoden einer Variablen ansprechbar.

Beim Anlegen eines Tag-Objekts werden alle Eigenschaften mit folgenden Werten initialisiert:

- Value = VT_EMPTY
- Name = Variablenname
- QualityCode = BAD NON-SPECIFIC
- TimeStamp = 0
- LastError = 0
- ErrorDescription = " "

Hinweis

Eine Zusammenfassung der möglichen Quality Codes finden Sie im WinCC Information System unter dem Stichwort "Kommunikation" > "Diagnose" oder "Kommunikation" > "Quality Codes".

Verwendung

Sie verwenden das Objekt "Tag" folgendermaßen:

- Eigenschaften "Name", "QualityCode", "TimeStamp", "LastError" und "ErrorDescription"
Informationen zur Variable auslesen
- Methode "Write", Eigenschaft "Value"
Wert einer Variablen setzen
- Methode "Read", Eigenschaft "Value"
Wert einer Variablen lesen

Beispiel

Das folgende Beispiel zeigt das Lesen des Wertes einer Variablen "Tag1":

```
'VBS13
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read()
MsgBox objTag.Value
```

Deklaration von Variablen in WinCC

Definieren Sie interne Variablen in VB-Skript immer über die Anweisung "Dim", um fehlerhafte Variablenschreibweisen zu verhindern.

Beim Anlegen einer neuen Aktion wird die Anweisung "Option explicit" automatisch und nicht löscherbar in den Deklarationsbereich eingetragen.

Verwenden Sie die Anweisung "Option explicit" nicht in Ihrem Code, da es sonst zu Laufzeitfehlern kommen kann.

Beispiel

Das folgende Beispiel zeigt die Deklaration einer VBScript-Variablen "lngVar":

```
'VBS14  
Dim lngVar  
lngVar = 5  
MsgBox lngVar
```

Hinweis

Variablennamen dürfen keine Sonderzeichen enthalten.

Beachten Sie, dass eine Variable beim Anlegen keinen Wert enthält (Value = VT_EMPTY).
Initialisieren Sie Variablen nach der Deklaration mit einem entsprechenden Wert.

Hinweise zu CrossReference

Alle Variablen, die Sie mit der Standardformulierung

```
HMIRuntime.Tags ("TagName")
```

adressieren, werden durch die CrossReference von WinCC automatisch erfasst und sind in den Bildeigenschaften aufgeführt.

Wenn Sie Variablen mit anderen Formulierungen in Ihrem Code ansprechen, können Sie diese über folgende Sektion der CrossReference bekanntmachen:

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

Sie können diese Sektion beliebig oft in VBS-Aktionen einfügen.

Hinweis

Zusammengesetzte Variablennamen werden von der CrossReference möglicherweise nicht erfasst.

Siehe auch

Name (Seite 1749)

Value (Seite 1968)

ErrorDescription (Seite 1648)

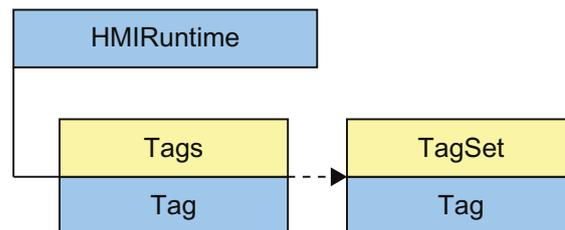
LastError (Seite 1705)

QualityCode (Seite 1780)

TimeStamp (Seite 1882)

3.5.3.20 Tags (Auflistung) (RT Professional)

Beschreibung



Die Auflistung "Tags" ermöglicht Ihnen den Zugriff auf Variablen in WinCC Runtime. Als Ergebnis des Zugriffs auf die "Tags"-Auflistung wird ein Objekt vom Typ "Tag" geliefert. Über das Tag-Objekt kann auf alle Eigenschaften und Methoden einer Variable zugegriffen werden.

Hinweis

"Tags" ist eine Auflistung mit eingeschränktem Funktionsumfang. Auf die Variablen innerhalb der Auflistung kann nicht über den Index, sondern nur über den Variablennamen zugegriffen werden. Die Standard-Methoden `get_Count` und `get_NewEnum` sind bei der Tags-Auflistung nicht anwendbar.

Verwendung

Der Zugriff auf Variablen der Auflistung erfolgt über:

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

Über die Auflistung Tags werden Variablen (Tag-Objekte) für lesenden und schreibenden Zugriff deklariert. Damit der lesende und schreibende Zugriff fehlerfrei ausgeführt werden kann, müssen die entsprechenden HMI-Variablen vorhanden sein.

Sie können in VBScript in HMI-Variablen direkt über Ihren Namen ansprechen und Werte setzen und lesen. Möchten Sie aber zusätzliche Eigenschaften der Variablen wie den Qualitycode oder den Zeitstempel erfragen, oder eine Fehlerbehandlung durchführen, so müssen Sie die Variable immer über die Tags-Auflistung ansprechen. Über das zurückgelieferte Tag-Objekt haben Sie Zugriff auf alle Eigenschaften und Methoden der Variablen.

Über die Methode "CreateTagSet" kann ein TagSet-Objekt erzeugt werden, das den Zugriff auf mehrere Variablen gleichzeitig ermöglicht.

Beispiel

Wenn Sie Variablen anlegen, verwenden Sie den -Variablennamen.

```
'VBS16
Dim objTag
Set objTag = HMIRuntime.Tags("Tagname")
Wenn Sie nur den Variablennamen verwenden, wird die Eigenschaft "TagPrefix" mit den Werten
aus dem aktuellen Kontext (dem aktuellen Bildfenster) belegt.
```

3.5.3.21 TagSet (Auflistung) (RT Professional)

Beschreibung

Das Objekt "TagSet" ermöglicht den gleichzeitigen Zugriff auf mehrere Variablen in einem Aufruf. Der gleichzeitige Zugriff erfolgt mit besserer Performance und geringerer Kommunikations-Last als beim Einzel-Zugriff auf die verschiedenen Variablen.

Verwendung

Sie verwenden das TagSet-Objekt folgendermaßen :

- Methode "Add"
Variablen zur Auflistung hinzufügen
- Methode "Item"
Auf die in der Auflistung enthaltenen Tag-Objekte und deren Eigenschaften zugreifen
- Methode "Write"
Alle Variablen der Auflistung schreiben
- Methode "Read"
Alle Variablen der Auflistung lesen
- Methode "Remove"
Einzelne Variablen aus der Auflistung entfernen
- Methode "RemoveAll"
Alle Variablen aus der Auflistung entfernen

Der Zugriff auf Variablen der Auflistung erfolgt über:

```
'VBS169
Dim myTags
myTags = HMIRuntime.Tags.CreateTagSet
myTags ("Tagname")
```

Damit der lesende und schreibende Zugriff auf Variablen (Tag-Objekte) der Auflistung fehlerfrei ausgeführt werden kann, müssen die entsprechenden Variablen in WinCC vorhanden sein.

Wenn beim lesenden oder schreibenden Zugriff ein Fehler aufgetreten ist, gibt die eingesetzte Methode über die Eigenschaften "LastError" und "ErrorDescription" eine Fehlermeldung zurück.

Beispiel

Das folgende Beispiel zeigt, wie man ein TagSet-Objekt erzeugt, Variablen hinzufügt und Werte schreibt.

```
'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
```

Siehe auch

[ErrorDescription \(Seite 1648\)](#)

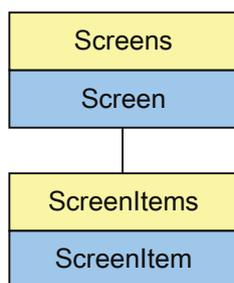
[LastError \(Seite 1705\)](#)

3.5.4 Objekttypen (Panels, Comfort Panels, RT Advanced, RT Professional)

3.5.4.1 Objekte A-I (Panels, Comfort Panels, RT Advanced, RT Professional)

AlarmControl (RT Professional)

Beschreibung



Stellt das Objekt "Meldeanzeige" dar. Das AlarmControl-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIAlarmControl

Beispiel

Im folgenden Beispiel wird das Objekt mit dem Namen "Control1" um 10 Pixel nach rechts verschoben:

```
'VBS54  
Dim objControl  
Set objControl = ScreenItems("Control1")  
objControl.Left = objControl.Left + 10
```

Abkürzung	Gültigkeit
Pa	Panels
A	RT Advanced
P	RT Professional

Tabelle 3-1 Eigenschaften

Eigenschaften	Read.	Write	Beschreibung
Activate (Seite 2069)	P	P	Legt fest, ob die darzustellenden Daten vom Meldeserver angefordert werden.
AlarmListType	P	P	Legt die Meldeklassen fest, die in Runtime gemeldet werden sollten.
AllServer	P	P	Legt fest, dass die Meldungen aller verfügbaren Server angezeigt werden.
ApplyProjectSettings (Seite 1564)	P	P	Legt fest, ob die Projekteinstellungen aus dem Editor "HMIMeldungen" übernommen werden.
ApplyProjectSettingsForDesign-Mode	P	P	Legt fest, dass für das Design die Projekteinstellungen verwendet werden.
AutoCompleteColumns (Seite 1568)	P	P	Legt fest, ob Leerspalten angezeigt werden, wenn das Control breiter ist als die projektierten Spalten.
AutoCompleteRows (Seite 1568)	P	P	Legt fest, ob Leerzeilen angezeigt werden, wenn das Control länger ist als die Anzahl der projektierten Zeilen.
AutoSelectionColors (Seite 1569)	P	P	Legt fest, ob die Markierungsfarben für Zellen und Zeilen mit den vom System festgelegten Farben dargestellt werden.
AutoSelectionRectColor (Seite 1570)	P	P	Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
Blocks	P	P	Legt die Meldeblöcke fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
Caption (Seite 1608)	P	P	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CellCut (Seite 1611)	P	P	Legt fest, ob die Inhalte der Zellen abgekürzt werden, wenn die Zellen zu schmal sind.
CellSpaceBottom (Seite 1612)	P	P	Legt den Abstand von unten fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceLeft (Seite 1612)	P	P	Legt den Einzug Links fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceRight (Seite 1613)	P	P	Legt den Einzug Rechts fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceTop (Seite 1613)	P	P	Legt den Abstand von oben fest, der in den Zellen der Tabelle verwendet wird.
Closeable (Seite 1617)	P	P	Legt fest, dass Control in Runtime geschlossen werden kann.
ColumnResize (Seite 1621)	P	P	Legt fest, ob Sie die Breite der Spalten ändern können.
ColumnScrollbar (Seite 1621)	P	P	Legt den Typ der Bildlaufleiste fest.

Eigenschaften	Read.	Write	Beschreibung
ColumnTitleAlignment (Seite 1624)	P	P	Legt fest, wie die Spaltenüberschriften ausgerichtet werden.
ColumnTitles (Seite 1625)	P	P	Legt fest, ob die Spaltenüberschrift angezeigt wird.
ControlDesignMode (Seite 1831)	P	P	Legt das Design fest.
DefaultMsgFilterSQL	P	P	Legt ein SQL-Statement als Standard für Meldefilter fest.
DefaultSort	P	P	Legt die Sortierung fest.
DefaultSort2	P	P	Legt die Sortierung fest.
DefaultSort2Column	P	P	Legt die Sortierung fest.
DisplayOptions (Seite 1639)	P	P	Legt fest, welche Meldungen angezeigt werden.
DoubleClickAction (Seite 1640)	P	P	Legt die Aktion fest, die bei Doppelklick auf eine Meldezeile in Runtime ausgeführt wird.
ExportDirectoryChangeable (Seite 1650)	P	P	Legt fest, ob das Verzeichnis für den Datenexport in Runtime geändert werden kann.
ExportDirectoryname (Seite 1651)	P	P	Legt das Verzeichnis fest, in das die Runtime-Daten exportiert werden.
ExportFileExtension (Seite 1651)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportFilename (Seite 1652)	P	P	Legt den Dateinamen der Datei fest, in welche die Runtime-Daten exportiert werden.
ExportFilenameChangeable (Seite 1652)	P	P	Legt fest, ob der Dateiname der Exportdatei in Runtime geändert werden kann.
ExportFormatGuid (Seite 1653)	P	P	Festgelegte Zuordnung von Ident-Nummer und Export-Provider.
ExportFormatName (Seite 1654)	P	P	Legt das Dateiformat für den Export fest. Zurzeit steht nur das "csv"-Dateiformat für den Export zur Verfügung.
ExportParameters (Seite 1654)	P	P	Legt die Parameter des ausgewählten Formats über den Dialog "Eigenschaften" fest.
ExportSelection (Seite 1655)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportShowDialog (Seite 1655)	P	P	Legt fest, ob der Dialog zum Datenexport in Runtime angezeigt wird.
Filters	P	P	Legt Kriterien in SQL-Syntax für die Datenbank fest.
Font (Seite 1669)	P	P	Legt die Schriftart fest oder gibt sie zurück.
GridLineColor (Seite 1678)	P	P	Legt die Farbe der Rasterlinien fest.
GridLineWidth (Seite 1679)	P	P	Legt die Stärke der Trennlinien in Pixel fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HitlistColumnAdd (Seite 1687) *	P	P	Übernimmt den ausgewählten Meldeblock aus der Liste der vorhandenen Meldeblöcke in die Liste der gewählten Meldeblöcke.
HitlistColumnCount (Seite 1687) *	P	P	Gibt die Anzahl der gewählten Meldeblöcke an, die in der Hitliste in Runtime angezeigt werden.
HitlistColumnIndex (Seite 1688)	P	P	Referenziert einen für die Hitliste gewählten Meldeblock.
HitlistColumnName (Seite 1688) *	P	P	Zeigt den Namen des Meldeblocks der Hitliste an, der mit dem Attribut "HitlistColumnIndex" referenziert wird.
HitlistColumnRemove (Seite 1689) *	P	P	Entfernt den markierten Meldeblock aus der Liste der gewählten Meldeblöcke und fügt ihn in die Liste der vorhandenen Meldeblöcke ein.

Eigenschaften	Read.	Write	Beschreibung
HitlistColumnRepos (Seite 1689) *	P	P	Ändert die Reihenfolge der Meldeblöcke. "Auf" und "Ab" bewegen den ausgewählten Meldeblock in der Liste nach oben oder unten.
HitlistColumnSort (Seite 1690) *	P	P	Legt fest, wie der im "HitlistColumnIndex" referenzierte Meldeblock für die Hitliste sortiert wird.
HitlistColumnSortIndex (Seite 1690) *	P	P	Gibt die Sortierreihenfolge des im "HitlistColumnIndex" referenzierten Meldeblocks der Hitliste an.
HitlistColumnVisible (Seite 1691) *	P	P	In der Liste werden die gewählten Meldeblöcke der Meldeliste oder Hitliste angezeigt, die in Runtime im Control verwendet werden.
HitListDefaultSort (Seite 1691)	P	P	Legt die Standardsortierung in den Tabellenspalten der Hitliste fest.
HitListMaxSourceItems (Seite 1692)	P	P	Legt die maximale Zahl der Datensätze fest, die aus dem Meldearchiv zum Erstellen der Hitliste verwendet werden.
HitListMaxSourceItemsWarn (Seite 1692)	P	P	Legt fest, ob ein Warnhinweis ausgegeben wird, wenn die maximale Zahl der Datensätze erreicht ist, die in "HitListMaxSourceItems" festgelegt wurde.
HitListRelTime (Seite 1694)	P	P	Legt fest, ob ein Zeitbereich für die Statistik verwendet wird.
HitListRelTimeFactor	P	P	Legt den Zeitfaktor fest, der mit dem Zeittyp "HitlistRelativeTimeFactorType" den Zeitraum bestimmt, für den die Statistik der Hitliste erstellt wird.
HitListRelTimeFactorType	P	P	Legt den Zeittyp fest, der mit dem Zeitfaktor "HitlistRelativeTimeFactor" den Zeitraum bestimmt, für den die Statistik der Hitliste erstellt wird.
HorizontalGridLines (Seite 1695)	P	P	Legt fest, ob horizontale Trennlinien angezeigt werden.
IconSpace (Seite 1697)	P	P	Legt den Abstand zwischen Symbol und Text in den Zellen der Tabelle fest.
Layer (Seite 1707)	P	P	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineColor (Seite 1721)	P	P	Gibt die Farbe für die Fenstertrennlinien an.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
LongTermArchiveConsistency (Seite 1726)	P	P	Wenn "LongTimeArchiveConsistency" auf "nein" gesetzt ist, werden 1000 Meldungen in der Langzeitarchivliste auf dem Einzelplatz, auf dem Server oder auf dem Client pro Server bzw. pro redundanter Serverpaar angezeigt.
MessageBlockAlign (Seite 1730) *	P	P	Legt fest, wie die Inhalte des ausgewählten Meldeblocks in der Tabelle ausgerichtet werden.
MessageBlockAutoPrecisions (Seite 1730) *	P	P	Legt fest, ob die Anzahl der Nachkommastellen automatisch bestimmt wird.
MessageBlockCaption (Seite 1731) *	P	P	Legt für den ausgewählten Meldeblock die Bezeichnung der Spaltenüberschrift in der Meldeanzeige fest.
MessageBlockCount (Seite 1731)	P	P	Gibt die Anzahl der vorhandenen Meldblöcke an, die für die Meldeliste und Hitliste zur Verfügung stehen.
MessageBlockDateFormat (Seite 1732) *	P	P	Legt fest, welches Datumsformat zur Anzeige der Meldungen verwendet wird.
MessageBlockExponentialFormat (Seite 1732) *	P	P	Legt fest, ob die Werte des ausgewählten Meldeblocks in Exponentialdarstellung angezeigt werden.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read.	Write	Beschreibung
MessageBlockFlashOn (Seite 1733) *	P	P	Legt fest, ob der Inhalt des gewählten Meldeblocks beim Erscheinen einer Meldung in Runtime blinkt.
MessageBlockHideText (Seite 1733) *	P	P	Legt fest, ob der Inhalt des ausgewählten Meldeblocks als Text angezeigt wird.
MessageBlockHideTitleText (Seite 1734) *	P	P	Legt fest, ob die Überschrift des ausgewählten Meldeblocks als Text angezeigt wird.
MessageBlockId (Seite 1734)	P	P	Festgelegte Zuordnung von Ident-Nummer und Meldeblock in der Meldeanzeige.
MessageBlockIndex (Seite 1734)	P	P	Referenziert einen vorhandenen Meldeblock.
MessageBlockLeadingZeros (Seite 1735) *	P	P	Legt fest, ob das Format des ausgewählten Meldeblocks mit führenden Nullen angezeigt wird.
MessageBlockLength (Seite 1735) *	P	P	Legt für den Inhalt des gewählten Meldeblocks die Länge in Zeichen fest.
MessageBlockName (Seite 1736)	P	P	Zeigt den Namen für den ausgewählten Meldeblock an.
MessageBlockPrecisions (Seite 1736) *	P	P	Legt die Anzahl der Nachkommastellen der Werte des ausgewählten Meldeblocks fest.
MessageBlockSelected (Seite 1737) *	P	P	Vorhandene Meldeblöcke sind Blöcke, die zur Verwendung im Control in Runtime für die Meldeliste oder Hitliste zur Verfügung stehen.
MessageBlockShowDate (Seite 1737) *	P	P	Legt fest, ob im Meldeblock "Uhrzeit" neben der Uhrzeit auch das Datum angezeigt wird.
MessageBlockShowIcon (Seite 1738) *	P	P	Legt fest, ob der Inhalt des ausgewählten Meldeblocks als Symbol angezeigt wird.
MessageBlockShowTitleIcon (Seite 1738) *	P	P	Legt fest, ob die Überschrift des ausgewählten Meldeblocks als Text angezeigt wird.
MessageBlockTextId (Seite 1739)	P	P	Legt die Bezeichnung für den gewählten Meldeblock mit Hilfe einer Text-Identnummer fest, die der Text-Library entnommen wurde.
MessageBlockTimeFormat (Seite 1739) *	P	P	Legt fest, welches Zeitformat bzw. Zeitdauerformat zur Anzeige der Meldungen verwendet wird.
MessageColumnAdd (Seite 1740) *	P	P	Übernimmt den markierten Meldeblock aus der Liste der vorhandenen Meldeblöcke in die Liste der gewählten Meldeblöcke.
MessageColumnCount (Seite 1741) *	P	P	Gibt die Anzahl der gewählten Meldeblöcke an, die in der Meldeliste in Runtime angezeigt werden.
MessageColumnIndex (Seite 1741)	P	P	Referenziert einen für die Meldeliste gewählten Meldeblock.
MessageColumnName (Seite 1742) *	P	P	Zeigt den Namen des Meldeblocks der Meldeliste an, der mit der Eigenschaft "MessageColumnIndex" referenziert wird.
MessageColumnRemove (Seite 1742) *	P	P	Entfernt den markierten Meldeblock aus der Liste der gewählten Meldeblöcke und fügt ihn in die Liste der vorhandenen Meldeblöcke ein.
MessageColumnRepos (Seite 1743) *	P	P	Ändert die Reihenfolge der Meldeblöcke. "Auf" und "Ab" bewegen den ausgewählten Meldeblock in der Liste nach oben oder unten.
MessageColumnSort (Seite 1743) *	P	P	Legt fest, wie der im "MessageColumnIndex" referenzierte Meldeblock sortiert wird.
MessageColumnSortIndex (Seite 1744) *	P	P	Gibt die Sortierreihenfolge des im "MessageColumnIndex" referenzierten Meldeblocks an.

Eigenschaften	Read.	Write	Beschreibung
MessageColumnVisible (Seite 1744) *	P	P	In der Liste werden die gewählten Meldeblöcke der Meldeliste oder Hitliste angezeigt, die in Runtime in der Meldeanzeige verwendet werden.
Moveable (Seite 1748)	P	P	Legt fest, ob das Fenster in Runtime bewegt werden kann.
MsgFilterSQL (Seite 1748)	P	P	Legt für die benutzerdefinierte Selektion der Meldungen ein oder mehrere SQL-Statements fest.
Name (Seite 1749)	P	P	Gibt den Objektnamen als STRING zurück.
OperatorAlarms	P	P	Legt die Einstellungen für Bedienmeldungen fest.
OperatorMessageId (Seite 1758) *	P	P	Festgelegte Zuordnung von Ident-Nummer und Auslöseereignis in der Meldeanzeige.
OperatorMessageIndex (Seite 1759) *	P	P	Referenziert das Meldeereignis für eine Bedienmeldung.
OperatorMessageName (Seite 1760) *	P	P	Zeigt bei Meldeereignissen für Bedienmeldungen den Namen an, der mit dem Ereignis "OperatorMessageIndex" referenziert wird.
OperatorMessageNumber (Seite 1760) *	P	P	Legen Sie eine Meldenummer für die Bedienmeldung des ausgewählten Meldeereignisses fest, wenn Sie nicht die Bedienmeldung von WinCC verwenden.
OperatorMessageSelected (Seite 1761) *	P	P	Aktivieren Sie die Meldeereignisse in der Liste, bei denen Bedienmeldungen ausgelöst werden.
OperatorMessageSource1 (Seite 1761) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 1" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource10 (Seite 1766) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 10" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource2 (Seite 1762) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 2" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource3 (Seite 1762) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 3" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource4 (Seite 1763) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 4" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource5 (Seite 1763) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 5" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource6 (Seite 1764) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 6" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource7 (Seite 1764) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 7" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource8 (Seite 1765) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 8" der hier projektierten Bedienmeldung hinzugefügt wird.
OperatorMessageSource9 (Seite 1765) *	P	P	Bestimmen Sie einen Meldeblock der bedienten Meldung, der zum "Prozesswertblock 9" der hier projektierten Bedienmeldung hinzugefügt wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read.	Write	Beschreibung
OperatorMessageSourceType1 (Seite 1767) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType10 (Seite 1771) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType2 (Seite 1767) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType3 (Seite 1768) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType4 (Seite 1768) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType5 (Seite 1769) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType6 (Seite 1769) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType7 (Seite 1770) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType8 (Seite 1770) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
OperatorMessageSourceType9 (Seite 1771) *	P	P	Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.
PageMode (Seite 1772)	P	P	Legt die Seitenausrichtung (Hochformat / Querformat) fest.
PageModeMessageNumber (Seite 1772)	P	P	Legt die Anzahl der Meldungen fest, die beim Blättern in der Langzeitarchivliste pro Seite angezeigt werden.
PrintJob	P	P	Legt einen Druckauftrag fest
RowScrollbar (Seite 1789)	P	P	Legt fest, ob Zeilen-Rollbalken angezeigt werden.
RowTitleAlignment	P	P	Legt fest, wie die Zeilenüberschriften ausgerichtet werden.
RowTitles	P	P	Legt fest, dass Zeilenköpfe angezeigt werden.
RTPersistence (Seite 1790)	P	P	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
RTPersistenceAuthorization (Seite 2046)	P	P	Legt die Berechtigung fest, die in Runtime zur Online-Projektierung erforderlich ist.
RTPersistenceType (Seite 1790)	-	-	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
SelectedCellColor (Seite 1802)	P	P	Gibt die Hintergrundfarbe der markierten Zelle an.
SelectedCellForeColor (Seite 1803)	P	P	Gibt die Schriftfarbe der markierten Zelle an.
SelectedRowColor (Seite 1804)	P	P	Gibt die Hintergrundfarbe der markierten Zeile an.
SelectedRowForeColor (Seite 1805)	P	P	Gibt die Schriftfarbe der markierten Zeile an.
SelectedTitleColor (Seite 1805)	P	P	Gibt die Hintergrundfarbe der markierten Tabellenüberschrift an.
SelectedTitleForeColor (Seite 1806)	P	P	Gibt die Schriftfarbe der markierten Tabellenüberschrift an.
SelectionColoring (Seite 1808)	P	P	Legt fest, ob Markierungsfarben für Zelle oder Zeile verwendet werden.
SelectionRect (Seite 1810)	P	P	Legt fest, ob ein Auswahlrahmen für die markierten Zellen bzw. Zeilen verwendet wird.

Eigenschaften	Read.	Write	Beschreibung
SelectionRectColor (Seite 1810)	P	P	Legt die Farbe des Auswahlrechtecks im Meldefenster fest, wenn der SelectionType "1" ist.
SelectionRectWidth (Seite 1811)	P	P	Legt die Linienstärke des Auswahlrechtecks im Meldefenster fest, wenn der SelectionType "1" ist.
SelectionType (Seite 1812)	P	P	Legt fest, wie viele Zeilen Sie markieren können.
ServerNames (Seite 1815)	P	P	Legt die Server eines verteilten Systems fest, von denen die Meldeanzeige Daten bezieht.
ShowSortButton (Seite 1825)	P	P	Legt fest, ob die Sortiertaste über dem vertikalen Rollbalken angezeigt wird.
ShowSortIcon (Seite 1825)	P	P	Legt fest, ob das Sortiersymbol angezeigt wird.
ShowSortIndex (Seite 1826)	P	P	Legt fest, ob das Sortiersymbol angezeigt wird.
ShowTitle (Seite 1829)	P	P	Legt die Darstellung der Fensterüberschrift des Controls fest.
Sizeable (Seite 1831)	P	P	Legt fest, ob in Runtime die Größe des Objekts geändert werden kann.
SortSequence (Seite 1833)	P	P	Legt fest, wie die Sortierreihenfolge durch Mausklick verändert werden kann.
StatusbarBackColor (Seite 1836)	P	P	Gibt die Hintergrundfarbe für die Statuszeile an.
StatusbarElementAdd (Seite 1836) *	P	P	Legt ein neues, benutzerdefiniertes Element der Statuszeile an.
StatusbarElementAutoSize (Seite 1837) *	P	P	Legt fest, ob die Breite des ausgewählten Elements der Statuszeile automatisch eingestellt wird.
StatusbarElementCount (Seite 1838) *	P	P	Gibt die Anzahl der projektierbaren Elemente der Statuszeile an.
StatusbarElementIconId (Seite 1838) *	P	P	Festgelegte Zuordnung von Ident-Nummer und Symbol eines Elementes der Statuszeile.
StatusbarElementId (Seite 1839) *	P	P	Eindeutige Ident-Nummer für das gewählte Element der Statuszeile.
StatusbarElementIndex (Seite 1840)	P	P	Referenziert ein Element der Statuszeile. Unter Verwendung dieser Eigenschaft können Sie einem bestimmten Element der Statuszeile die Werte anderer Eigenschaften zuweisen.
StatusbarElementName (Seite 1840) *	P	P	Zeigt den Objektnamen für das gewählte Element der Statuszeile an.
StatusbarElementRemove (Seite 1841) *	P	P	Entfernt das ausgewählte Element der Statuszeile.
StatusbarElementRename (Seite 1842) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Statuszeile, das über die Eigenschaft "StatusbarElementIndex" referenziert wird.
StatusbarElementRepos *	P	P	Legt den Index fest, der dem Element zugeordnet werden soll.
StatusbarElements (Seite 1842)	-	-	Legt die in der Statuszeile darzustellenden Elemente fest.
StatusbarElementText *	P	P	Legt den Text des momentan gewählten Elements der Statuszeile fest.
StatusbarElementTooltipText (Seite 1842) *	P	P	Legt den Text für den Tooltip des benutzerdefinierten Elements der Statuszeile fest.
StatusbarElementUserDefined (Seite 1843) *	P	P	Zeigt an, ob das Element der Statuszeile vom Projektteur als neues, benutzerdefiniertes Element hinzugefügt wurde.
StatusbarElementVisible (Seite 1844) *	P	P	Aktivieren Sie in der Liste die Elemente der Statuszeile, die sie in Runtime anzeigen wollen.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read.	Write	Beschreibung
StatusbarElementWidth (Seite 1844) *	P	P	Gibt die Breite des ausgewählten Elements der Statuszeile in Pixel an.
StatusbarFont	-	-	Statuszeile Schriftart
StatusbarFontColor (Seite 1845)	P	P	Gibt die Schriftfarbe der Texte in der Statuszeile an.
StatusbarShowTooltips (Seite 1846)	P	P	Legt fest, ob in Runtime die Tooltips zu den Elementen der Statuszeile angezeigt werden.
StatusbarText (Seite 1846)	P	P	Festgelegter Text in der Statuszeile.
StatusbarUseBackColor (Seite 1847)	P	P	Legt fest, ob die Hintergrundfarbe der Statuszeile angezeigt wird.
StatusbarVisible (Seite 1848)	P	P	Legt fest, ob die Statuszeile des Controls angezeigt wird.
TableColor	P	P	Gibt die Hintergrundfarbe für die Zeilen an. Über die öffnen Sie den Farbauswahl-Dialog.
TableColor2	P	P	Gibt die Hintergrundfarbe der "Zeilenfarbe 2" an.
TableForeColor	P	P	Gibt die Schriftfarbe der "Zeilenfarbe " an.
TableForeColor2	P	P	Gibt die Schriftfarbe der "Zeilenfarbe 2" an.
TimeBase (Seite 1866)	P	P	Legt die Zeitbasis fest.
TitleColor	P	P	Legt die Farbe der Titelzeile fest.
TitleCut	P	P	Legt fest, ob die Inhalte der Felder einer Titelleiste bei zu kleiner Spaltenbreite abgekürzt werden sollen.
TitleDarkShadowColor	P	P	Gibt die Farbe für die dunkle Seite der Schattierung an.
TitleForeColor	P	P	Gibt die Schriftfarbe der Tabellenüberschrift für den gewählten Zustand an.
TitleGridLineColor	P	P	Gibt die Farbe der Trennlinien in der Tabellenüberschrift an.
TitleLightShadowColor	P	P	Gibt die Farbe für die helle Seite der Schattierung an.
TitleSort	P	P	Legt fest, wie das Sortieren über Spaltenüberschrift ausgelöst wird.
TitleStyle	P	P	Legt fest, ob eine Schattierungsfarbe für die Tabellenüberschrift verwendet wird.
ToolBarAlignment	P	P	Legt die Position der Symbolleiste fest oder gibt sie zurück.
ToolBarBackColor	P	P	Gibt die Hintergrundfarbe der Symbolleiste an.
ToolBarButtonActive *	P	P	Legt fest, ob die mit der Taste verbundene Funktion in Runtime aktiv geschaltet ist.
ToolBarButtonAdd *	P	P	Legt eine neue, benutzerdefinierte Tastenfunktion an.
ToolBarButtonBeginGroup *	P	P	Legt fest, ob vor der ausgewählten Tastenfunktion ein Trennzeichen eingefügt wird.
ToolBarButtonClick	P	P	Klickt eine Schaltfläche der Symbolleiste.
ToolBarButtonCount *	P	P	Gibt die Anzahl der projektierbaren Tastenfunktionen an.
ToolBarButtonEnabled *	P	P	Legt fest, ob eine benutzerdefinierte Taste der Symbolleiste bedienbar ist.
ToolBarButtonHotKey *	P	P	Zeigt den Hotkey für die ausgewählte Tastenfunktion an.
ToolBarButtonId *	P	P	Eindeutige Ident-Nummer für die gewählte Tastenfunktion.
ToolBarButtonIndex	P	P	Referenziert eine Tastenfunktion.
ToolBarButtonLocked *	P	P	Legt für eine benutzerdefinierte Taste der Symbolleiste fest, ob der eingerastete, gedrückte Zustand der Taste dargestellt wird.
ToolBarButtonName *	P	P	Zeigt den Namen für die ausgewählte Tastenfunktion an.

Eigenschaften	Read.	Write	Beschreibung
ToolBarButtonPasswordLevel *	P	P	Zeigt die Berechtigung für die ausgewählte Tastenfunktion an.
ToolBarButtonRemove *	P	P	Entfernt die ausgewählte Tastenfunktion aus der Liste.
ToolBarButtonRename *	P	P	Ändert den Namen des benutzerdefinierten Elements der Symbolleiste, das über die Eigenschaft "ToolBarButtonIndex" referenziert wird.
ToolBarButtonRepos *	P	P	Ändert die Reihenfolge der Tastenfunktionen.
ToolBarButtons	-	-	Legt fest, welche Schaltflächen für die Symbolleiste verfügbar sind.
ToolBarButtonTooltipText *	P	P	Legt den Text für den Tooltip der Taste fest.
ToolBarButtonUserDefined *	P	P	Zeigt, ob die Taste der Symbolleiste vom Projektteur als neue, benutzerdefinierte Taste hinzugefügt wurde.
ToolBarButtonVisible *	P	P	Legt fest, dass die Schaltfläche sichtbar ist.
ToolBarShowTooltips	P	P	Legt fest, ob in Runtime die Tooltips zu den Tastenfunktionen angezeigt werden.
ToolBarUseBackColor	P	P	Legt fest, ob die Hintergrundfarbe der Symbolleiste angezeigt wird.
ToolBarUseHotKeys	P	P	Legt fest, ob die Hotkeys für die Tastenfunktionen in Runtime aktiviert sind.
ToolBarVisible	P	P	Legt fest, ob die Symbolleiste des Controls angezeigt wird.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
UseMessageColor (Seite 1958)	P	P	Legt fest, ob das Scrollen aktiviert ist.
UseSelectedTitleColor (Seite 1959)	P	P	Legt fest, ob eine Markierungsfarbe für die Tabellenüberschriften von markierten Tabellenzellen verwendet wird.
UseTableColor2 (Seite 1959)	P	P	Legt fest, ob eine zweite Zeilenfarbe bei der Darstellung der Tabelle verwendet wird.
VerticalGridLines (Seite 1972)	P	P	Legt fest, ob vertikale Trennlinien angezeigt werden.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

* im ES nicht sichtbar

Tabelle 3-2 Methoden

Methoden	Gültig.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
AttachDB (Seite 2163)	P, A	Führt die Funktion der Schaltfläche "Backup verbinden" des Controls aus.
CopyRows (Seite 2165)	P	Führt die Funktion der Schaltfläche "Zeilen kopieren" des Controls aus.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.
DetachDB (Seite 2169)	P, A	Führt die Funktion der Schaltfläche "Backup trennen" des Controls aus.
Export (Seite 2170)	P, A	Führt die Funktion der Schaltfläche "Archiv exportieren" oder "Daten exportieren" des Controls aus.

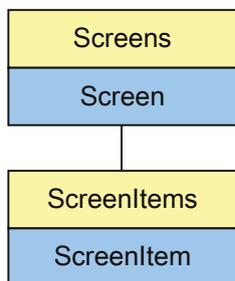
3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Methoden	Gültig.	Beschreibung
GetHistlistColumnCollection (Seite 2173)	P	Gibt die Auflistung aller Spalten-Objekte der Hitliste der Meldeanzeige als Typ "ICCAxCollection" zurück.
GetHitlistColumn (Seite 2174)	P	Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Hitliste der Meldeanzeige als Typ "ICCAxMessageColumn" zurück.
GetMessageBlock (Seite 2175)	P	Gibt das mit Namen oder Index bezeichnete Meldeblock-Objekt der Meldeanzeige als Typ "ICCAxMessageBlock" zurück.
GetMessageBlockCollection (Seite 2176)	P	Gibt die Auflistung aller Meldeblock-Objekte der Meldeanzeige als Typ "ICCAxCollection" zurück.
GetMessageColumn (Seite 2177)	P	Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Meldeanzeige als Typ "ICCAxMessageColumn" zurück.
GetOperatorMessage (Seite 2178)	P	Gibt das mit Namen oder Index bezeichnete Bedienmeldungs-Objekt der Meldeanzeige als Typ "ICCAxOperatorMessage" zurück.
GetMessageColumnCollection (Seite 2179)	P	Gibt die Auflistung aller Spalten-Objekte der Meldeanzeige als Typ "ICCAxCollection" zurück.
GetOperatorMessageCollection (Seite 2181)	P	Gibt die Auflistung aller Bedienmeldungs-Objekte der Meldeanzeige als Typ "ICCAxCollection" zurück.
GetRow (Seite 2182)	P	Gibt das mit der Zeilennummer bezeichnete Zeilen-Objekt der auf Tabellen basierenden Controls als Typ "ICCAxDataRow" zurück.
GetRowCollection (Seite 2183)	P	Gibt die Auflistung aller Zeilen-Objekte der auf Tabellen basierenden Controls als Typ "ICCAxDataRowCollection" zurück.
GetSelectedRow (Seite 2190)	P	Gibt das markierte Zeilen-Objekt eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.
GetSelectedRows (Seite 2191)	P	Gibt bei Mehrfachselektion die markierten Zeilen-Objekte eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.
GetStatusBarElement (Seite 2197)	P, A	Gibt das mit Namen oder Index bezeichnete Element der Statuszeile des Controls als Typ "ICCAxStatusBarElement" zurück.
GetStatusBarElementCollection (Seite 2198)	P, A	Gibt die Auflistung aller Elemente der Statuszeile des Controls als Typ "ICCAxCollection" zurück.
GetToolBarButton (Seite 2205)	P, A	Gibt die mit Namen oder Index bezeichnete Tastenfunktion der Symbolleiste des Controls als Typ "ICCAxToolBarButton" zurück.
GetToolBarButtonCollection (Seite 2207)	P, A	Gibt die Auflistung aller Tastenfunktionen der Symbolleiste des Controls als Typ "ICCAxCollection" zurück.
HideAlarm (Seite 2223)	P	Führt die Funktion der Schaltfläche "Meldung ausblenden" der Meldeanzeige aus.
LockAlarm (Seite 2225)	P	Führt die Funktion der Schaltfläche "Meldung sperren" der Meldeanzeige aus.
LoopInAlarm (Seite 2226)	P	Führt die Funktion der Schaltfläche "Loop in Alarm" der Meldeanzeige aus.
MoveToFirstLine (Seite 2227)	P	Führt die Funktion der Schaltfläche "Erste Meldung" der Meldeanzeige aus.
MoveToFirstPage (Seite 2228)	P	Führt die Funktion der Schaltfläche "Erste Seite" der Meldeanzeige aus.
MoveToLastLine (Seite 2229)	P	Führt die Funktion der Schaltfläche "Letzte Meldung" der Meldeanzeige aus.
MoveToLastPage (Seite 2229)	P	Führt die Funktion der Schaltfläche "Letzte Seite" der Meldeanzeige aus.
MoveToNextLine (Seite 2230)	P	Führt die Funktion der Schaltfläche "Nächste Meldung" der Meldeanzeige aus.
MoveToNextPage (Seite 2231)	P	Führt die Funktion der Schaltfläche "Nächste Seite" der Meldeanzeige aus.

Methoden	Gültig.	Beschreibung
MoveToPreviousLine (Seite 2232)	P	Führt die Funktion der Schaltfläche "Vorhergehende Meldung" der Meldeanzeige aus.
MoveToPreviousPage (Seite 2232)	P	Führt die Funktion der Schaltfläche "Vorhergehende Seite" der Meldeanzeige aus.
Print (Seite 2236)	P, A	Führt die Funktion der Schaltfläche "Drucken" des Controls aus.
QuitHorn (Seite 2236)	P	Führt die Funktion der Schaltfläche "Quittierung zentraler Melder" der Meldeanzeige aus.
QuitSelected (Seite 2237)	P	Führt die Funktion der Schaltfläche "Einzelquittierung" der Meldeanzeige aus.
QuitVisible (Seite 2237)	P	Führt die Funktion der Schaltfläche "Sammelquittierung" der Meldeanzeige aus.
SelectAll (Seite 2251)	P	Markiert alle Zeilen in einem tabellenbasierten Control.
SelectRow (Seite 2252)	P	Markiert eine bestimmte Zeile in einem tabellenbasierten Control.
ShowComment (Seite 2254)	P	Führt die Funktion der Schaltfläche "Kommentar-Dialog" der Meldeanzeige aus.
ShowDisplayOptionsDialog (Seite 2254)	P	Führt die Funktion der Schaltfläche "Anzeigeoptions-Dialog" der Meldeanzeige aus.
ShowEmergencyQuitDialog (Seite 2255)	P	Führt die Funktion der Schaltfläche "Not-Quittierung" der Meldeanzeige aus.
ShowHelp (Seite 2255)	P, A	Führt die Funktion der Schaltfläche "Hilfe" des Controls aus.
ShowHideList (Seite 2256)	P	Führt die Funktion der Schaltfläche "Liste auszublender Meldungen" der Meldeanzeige aus.
ShowHitList (Seite 2256)	P	Führt die Funktion der Schaltfläche "Hitliste" der Meldeanzeige aus.
ShowInfoText (Seite 2257)	P	Führt die Funktion der Schaltfläche "Infotext-Dialog" der Meldeanzeige aus.
ShowLockDialog (Seite 2257)	P	Führt die Funktion der Schaltfläche "Sperr-Dialog" der Meldeanzeige aus.
ShowLockList (Seite 2258)	P	Führt die Funktion der Schaltfläche "Sperrliste" der Meldeanzeige aus.
ShowLongTermArchiveList (Seite 2258)	P	Führt die Funktion der Schaltfläche "Langzeitarchivliste" der Meldeanzeige aus.
ShowMessageList (Seite 2259)	P	Führt die Funktion der Schaltfläche "Meldeliste" der Meldeanzeige aus.
ShowPropertyDialog (Seite 2260)	P, A	Führt die Funktion der Schaltfläche "Konfigurationsdialog" des Controls aus.
ShowSelectionDialog (Seite 2261)	P	Führt die Funktion der Schaltfläche "Selektions-Dialog" der Meldeanzeige aus.
ShowShortTermArchiveList (Seite 2262)	P	Führt die Funktion der Schaltfläche "Kurzzeitarchivliste" der Meldeanzeige aus.
ShowSortDialog (Seite 2263)	P	Führt die Funktion der Schaltfläche "Sortier-Dialog" der Meldeanzeige aus.
ShowTimebaseDialog (Seite 2264)	P	Führt die Funktion der Schaltfläche "Zeitbasis-Dialog" der Meldeanzeige aus.
UnhideAlarm (Seite 2267)	P	Führt die Funktion der Schaltfläche "Meldung einblenden" der Meldeanzeige aus.
UnlockAlarm (Seite 2268)	P	Führt die Funktion der Schaltfläche "Meldung freigeben" der Meldeanzeige aus.
UnselectAll (Seite 2268)	P	Entfernt alle Markierungen in den Zellen eines tabellenbasierten Controls.
UnselectRow (Seite 2269)	P	Entfernt die Markierungen in einer bestimmten Zelle eines tabellenbasierten Controls.

AlarmView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Meldeanzeige" dar. Das MessageView-Objekt ist Element der ScreenItems-Auflistung.

Wenn Sie die Einstellungen für dieses Objekt mit einer benutzerdefinierten Funktion ändern, bleiben die geänderten Einstellungen auch nach einem Neuaufruf des Bildes mit diesem Objekt erhalten.

Hinweis

Das Objekt "Einfache Meldeanzeige" kann nicht mit einer benutzerdefinierten Funktion dynamisiert werden.

Abkürzung	Gültigkeit
Pa	Panels
A	RT Advanced
P	RT Professional

Tabelle 3-3 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AlarmAreaHeight **			
AlarmAreaWidth **			
AlarmClasses	-	-	Legt die Meldeklassen fest, die in Runtime gemeldet werden sollten.
AlarmLog	-	-	Legt fest, welche die Art von Meldungen in der Meldeanzeige angezeigt werden.
AlarmSource	-	-	Legt fest, dass Meldungen angezeigt werden.
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BarButtonElements **			Legt fest, dass Schaltflächen angezeigt werden.
BarButtonStyle	-	-	
Columns	-	-	Legt fest, welche Spalten angezeigt werden.
ColumnsMoveable	-	-	Legt fest, dass die Spalten verschiebbar sind.
ColumnTextAckGroup	-	-	Legt die Beschriftung der Spalte "AckGroup" fest.
ColumnTextAlarmState	-	-	Legt die Beschriftung der Spalte "State" fest.
ColumnTextAlarmText	-	-	Legt die Beschriftung der Spalte "Text" fest.
ColumnTextClassName	-	-	Legt die Beschriftung der Spalte "Class Name" fest.
ColumnTextDate	-	-	Legt die Beschriftung der Spalte "Date" fest.
ColumnTextDevice	-	-	Legt die Beschriftung der Spalte "Device" fest.
ColumnTextDiagnosable	-	-	Legt die Beschriftung der Spalte "Diagnosable" fest.
ColumnTextNumber	-	-	Legt die Beschriftung der Spalte "Number" fest.
ColumnTextTime	-	-	Legt die Überschrift der Spalte "Uhrzeit" fest.
CountOfLinesPerAlarms	-	-	
CountOfVisibleAlarms	-	-	
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FilterTag **			Legt die Variable vom Typ String zum Filtern von Meldetexten fest.
FilterText **			Legt den Text zum Filtern des Meldetextes fest.
FitToSize	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FocusColor	A	A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth	A	A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
ForeColor	-	-	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
GridlineColor	A	A	Legt die Farbe der Rasterlinien fest.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalScrollingEnabled	-	-	
IsRunningUnderCE *	-	-	

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
LineAlarmView **			
Name (Seite 1749)	A	A	Gibt den Objektnamen als STRING zurück.
PreferredUseOnAck	-	-	
S7Device	-	-	
SecurityForSimpleViewEnabled **			
SelectionBackColor (Seite 1808)	A	A	Legt die Hintergrundfarbe der selektierten Zellen fest.
SelectionForeColor (Seite 1809)	A	A	Legt die Vordergrundfarbe der selektierten Zellen fest.
SeperateLineForAlarmText	-	-	
ShowAcknowledgeButton	-	-	
ShowAlarmsFromDate (Seite 1816)	A	A	Legt fest, dass nur diese Meldeereignisse angezeigt werden, die in dieser Variablen gespeichert sind.
ShowAlarmsToAcknowledge	-	-	Legt fest, dass unquittierte Meldungen angezeigt werden.
ShowColumnHeaders	-	-	
ShowHelpButton	-	-	
ShowHorizontalGridlines	-	-	
ShowLoopInAlarmButton	-	-	Legt fest, dass die Schaltfläche "Loop-In-Alarm" angezeigt wird.
ShowMilliseconds	-	-	Legt fest, dass die Zeit in Millisekunden angezeigt wird.
ShowPendingAlarms	-	-	
SortByTimeDirection (Seite 1832)	-	-	Legt fest, ob die zuletzt eingegangene Meldung im Objekt "AlarmControl" oben angezeigt wird.
SortByTimeEnabled (Seite 2052)	A	A	Legt fest, ob die Sortierung der Meldungen nach der Uhrzeit im Objekt "AlarmControl" geändert werden kann.
TableBackColor (Seite 1850)	A	A	Legt die Hintergrundfarbe der Tabellenzellen des angegebenen Objekts fest.
TableFont	-	-	Legt die Schriftart in der Tabelle fest.
TableForeColor	A	A	Gibt die Schriftfarbe der "Zeilenfarbe" an.
TableHeaderBackColor (Seite 1854)	A	A	Legt die Hintergrundfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
TableHeaderFont	-	-	Legt die Textfarbe der Überschrift fest.
TableHeaderForeColor (Seite 1855)	A	A	Legt die Textfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
VerticalScrollBarEnabled	-	-	
VerticalScrollingEnabled	-	-	
ViewType	-	-	Legt den Typ der Meldeanzeige fest.
ViewTypeForSaveStream **			Legt den Anzeigetyp für den Save Stream fest.
Visible (Seite 1974) *	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

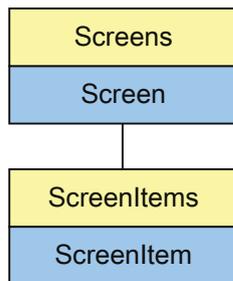
* im ES nicht sichtbar, ** keine Info dazu vorhanden

Tabelle 3-4 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	???	Aktiviert das Permanentfenster oder das Grundbild.
???		

Bar (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Balken" dar. Das Bar-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIBar

Abkürzung	Gültigkeit
Pa	Panels
A	RT Advanced
P	RT Professional

Tabelle 3-5 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AlarmLowerLimit (Seite 1557)	P	P	Legt den unteren Grenzwert fest, bei dem Alarm ausgelöst.
AlarmLowerLimitColor (Seite 1558)	P	P	Legt die Balkenfarbe für den Grenzwert "AlarmLowerLimit" fest.
AlarmLowerLimitEnabled (Seite 1558)	P	P	Legt fest, ob der Grenzwert "AlarmLowerLimit" überwacht wird.
AlarmLowerLimitRelative (Seite 1559)	P	P	Legt fest, ob der untere Grenzwert, bei dem Alarm ausgelöst wird, prozentual oder absolut ausgewertet wird.
AlarmUpperLimit (Seite 1560)	P	P	Legt den oberen Grenzwert fest, bei dem Alarm ausgelöst wird.
AlarmUpperLimitColor (Seite 1560)	P	P	Legt die Balkenfarbe für den Grenzwert "AlarmUpperLimit" fest.
AlarmUpperLimitEnabled (Seite 1561)	P	P	Legt fest, ob der Grenzwert "AlarmUpperLimit" überwacht wird.
AlarmUpperLimitRelative (Seite 1561)	P	P	Legt fest, ob der obere Grenzwert, bei dem Alarm ausgelöst wird, prozentual oder absolut ausgewertet wird.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
AverageLast15Values	P	P	Legt fest, dass der Durchschnitt der letzten 15 Werte angezeigt wird.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577) *	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578) *	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579) *	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580) *	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BarBackColor (Seite 1581)	P, A	P, A	Legt die Farbe des Balkenhintergrunds im angegebenen Objekt fest.
BarBackFillStyle (Seite 1581)	P	P	Legt das Füllmuster des Balkens fest.
BarBackFlashingColorOff	-	-	Legt die Rahmenhintergrundfarbe des Objekts für den Zustand "AUS" fest.
BarBackFlashingColorOn	-	-	Legt die Rahmenhintergrundfarbe des Objekts für den Zustand "EIN" fest.
BarBackFlashingEnabled	-	-	

Eigenschaften	Read	Write	Beschreibung
BarBackFlashingRate	-	-	Legt die Blinkfrequenz des Hintergrundes fest.
BarEdgeStyle	-	-	Legt den Stil der Rahmenlinie fest.
BarOrientation	P	P	Legt die Ausrichtung des Balkens fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderStyle3D	-	-	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
ColorChangeHysteresis (Seite 1619)	P	P	Legt die Hysterese prozentual zum Anzeigewert fest.
ColorChangeHysteresisEnabled (Seite 1619)	P	P	Legt fest, ob das Objekt mit Hysterese angezeigt wird.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
CountDivisions (Seite 1629)	P	P	Legt die Anzahl der Segmente fest, in die der Balken durch die großen Teilstriche der Skala unterteilt wird.
CountSubDivisions (Seite 1630)	A	A	Legt die Anzahl der Skalenteilstriche zwischen zwei Hauptteilstrichen des Objekts "Bar" fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Liniestärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	P	P	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662) *	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663) *	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665) *	-	-	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingOnLimitViolation	-	-	Legt fest, dass Blinken bei Grenzwertverletzung aktiviert ist.
FlashingRate (Seite 1666) *	-	-	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
ForeColor (Seite 1675)	P, A	P, A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
IntegerDigits (Seite 1700)	P	P, A	Legt die Anzahl der Vorkommastellen (0 bis 20) fest.
LargeTickLabelingStep	-	-	Legt fest, welche Skalenabschnitte beschriftet werden.
LargeTicksBold (Seite 1704)	P	P	Legt fest, ob die langen Teilstriche einer Skala fett angezeigt werden.
LargeTicksSize (Seite 1704)	P	P	Legt die Länge der langen Teilstriche einer Skala fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
Limit4LowerLimit (Seite 1712)	P	P	Legt den unteren Grenzwert für "Reserve4" fest.
Limit4LowerLimitColor (Seite 1712)	P	P	Legt die Farbe für den unteren Grenzwert "Reserve4" fest.
Limit4LowerLimitEnabled (Seite 1713)	P	P	Legt fest, ob der untere Grenzwert "Reserve4" überwacht wird.
Limit4LowerLimitRelative (Seite 1714)	P	P	Legt fest, ob der untere Grenzwert "Reserve4" prozentual oder absolut ausgewertet wird.
Limit4UpperLimit (Seite 1714)	P	P	Legt den oberen Grenzwert für "Reserve4" fest.
Limit4UpperLimitColor (Seite 1715)	P	P	Legt die Farbe für den oberen Grenzwert "Reserve4" fest.
Limit4UpperLimitEnabled (Seite 1716)	P	P	Legt fest, ob der obere Grenzwert "Reserve4" überwacht wird.
Limit4UpperLimitRelative (Seite 1716)	P	P	Legt fest, ob der obere Grenzwert "Reserve4" prozentual oder absolut ausgewertet wird.
Limit5LowerLimit (Seite 1717)	P	P	Legt den unteren Grenzwert für "Reserve5" fest.
Limit5LowerLimitColor (Seite 1717)	P	P	Legt die Farbe für den unteren Grenzwert "Reserve5" fest.
Limit5LowerLimitEnabled (Seite 1718)	P	P	Legt fest, ob der untere Grenzwert "Reserve5" überwacht wird.
Limit5LowerLimitRelative (Seite 1719)	P	P	Legt fest, ob der untere Grenzwert "Reserve5" prozentual oder absolut ausgewertet wird.
Limit5UpperLimit (Seite 1719)	P	P	Legt den oberen Grenzwert für "Reserve5" fest.
Limit5UpperLimitColor (Seite 1720)	P	P	Legt die Farbe für den oberen Grenzwert "Reserve5" fest.
Limit5UpperLimitEnabled (Seite 1720)	P	P	Legt fest, ob der obere Grenzwert "Reserve5" überwacht wird.
Limit5UpperLimitRelative (Seite 1721)	P	P	Legt fest, ob der obere Grenzwert "Reserve5" prozentual oder absolut ausgewertet wird.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienenden fest.
MaximumValue (Seite 1729)	P, A	P, A	Legt den Maximalwert der Skala im angegebenen Objekt fest.
MinimumValue (Seite 1745)	P, A	P, A	Legt den Minimalwert der Skala im angegebenen Objekt fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
Precision (Seite 1778)	P	P	Legt die Anzahl der Nachkommastellen (0 bis 20) fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
ProcessValue (Seite 1779)	P, A	P, A	Legt die Voreinstellung für den anzuzeigenden Wert fest.
ScaleColor (Seite 1792)	P, A	P, A	Legt die Farbe der Skala des angegebenen Objekts fest.
ScaleGradation (Seite 1792)	P, A	P, A	Legt den Abstand zwischen zwei großen Teilstrichen der Skala fest.
ScaleLabelFieldLength *	-	-	Legt den Abstand zur Position der Achsenbeschriftung fest.
ScaleLabelingDoubleLined	-	-	Legt fest, dass die die Skala bei horizontaler Balkenrichtung zweizeilig ist.
ScalePosition (Seite 1794)	P	P	Legt die Position der Skala des angegebenen Objekts fest.
ScaleStart	-	-	Legt das Minimum der Skala fest.
ScalingType (Seite 1797)	P	P	Legt die Art der Balkenskalierung fest.
SegmentColoring (Seite 1801)	P, A	P, A	Legt die Art des Farbumschlags fest, mit der im Objekt "Bar" das Überschreiten von Grenzwerten angezeigt wird.
ShowBadTagState (Seite 1817)	P	P	Legt fest, ob das Objekt grau dargestellt wird, wenn die verbundene Variable einen schlechten QualityCode oder Variablenstatus besitzt.
ShowLargeTicksOnly (Seite 1820)	P	P	Legt die Länge der Achsabschnitte in Pixel fest oder gibt sie zurück.
ShowLimitLines	-	-	Legt fest, dass bei den Grenzwerten eine Linie angezeigt wird.
ShowLimitMarkers (Seite 1821)	P	P	Legt fest, ob die Grenzwerte als Skalenwert angezeigt werden.
ShowScale (Seite 1824)	P	P	Legt fest, ob die Werte zusätzlich in einer Skala angezeigt werden.
ShowSignForPositiveLabel	-	-	Legt fest, dass bei positiven Beschriftungen ein "+" angezeigt wird.
ShowTickLabels (Seite 1828)	-	-	Legt fest, ob in der Skala die Beschriftung angezeigt wird.
ShowTrendIndicator (Seite 1830)	P	P	Legt fest, ob die Tendenz (steigend oder fallend) des zu überwachenden Messwertes mit einem kleinen Pfeil angezeigt wird.
StartValue (Seite 1835)	P	P	Legt den Wert des Nullpunktes der Skalanzeige fest.
ToleranceLowerLimit (Seite 1890)	P	P	Legt fest, ob der Grenzwert "ToleranceLowerLimit" überwacht wird.
ToleranceLowerLimitColor (Seite 1891)	P	P	Legt die Farbe für den unteren Grenzwert "ToleranceLowerLimit" fest.
ToleranceLowerLimitEnabled (Seite 1891)	P	P	Legt fest, ob der Grenzwert "ToleranceLowerLimit" überwacht wird.
ToleranceUpperLimit (Seite 1892)	P	P	Legt fest, ob der Grenzwert "ToleranceUpperLimit" überwacht wird.
ToleranceUpperLimitColor (Seite 1893)	P	P	Legt die Farbe für den oberen Grenzwert "ToleranceUpperLimit" fest.
ToleranceUpperLimitEnabled (Seite 1893)	P	P	Legt fest, ob der Grenzwert "ToleranceUpperLimit" überwacht wird.
ToleranceUpperLimitRelative (Seite 1894)	P	P	Legt fest, ob der untere Grenzwert "ToleranzHigh" prozentual oder absolut ausgewertet wird.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
TrendIndicatorColor (Seite 1922)	P	P	Legt die Farbe der Trendanzeige fest.
Unit (Seite 1951)	P, A	P, A	Legt die Maßeinheit im Objekt "IOField" fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
UseAutoScaling	-	-	Legt fest, dass die Intervalle der Skala automatisch angepasst werden.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
UseExponentialFormat (Seite 1957)	P	P	Legt fest, ob die Zahlen mit Exponenten (z. B. "1,00e+000") dargestellt werden.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
WarningLowerLimit (Seite 1977)	P	P	Legt den unteren Grenzwert "WarningLowerLimit" fest.
WarningLowerLimitColor (Seite 1978)	P	P	Legt die Farbe für den unteren Grenzwert "WarningLowerLimit" fest.
WarningLowerLimitEnabled (Seite 1978)	P	P	Legt fest, ob der Grenzwert "WarningLowerLimit" überwacht wird.
WarningLowerLimitRelative (Seite 1979)	P	P	Legt fest, ob der untere Grenzwert "WarningLowerLimit" prozentual oder absolut ausgewertet wird.
WarningUpperLimit (Seite 1981)	P	P	Legt den oberen Grenzwert "WarningUpperLimit" fest.
WarningUpperLimitColor (Seite 1982)	P	P	Legt die Farbe für den oberen Grenzwert "WarningUpperLimit" fest.
WarningUpperLimitEnabled (Seite 1982)	P	P	Legt fest, ob der Grenzwert "WarningUpperLimit" überwacht wird.
WarningUpperLimitRelative (Seite 1983)	P	P	Legt fest, ob der obere Grenzwert "WarningUpperLimit" prozentual oder absolut ausgewertet wird.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.
ZeroPoint (Seite 2007)	P	P	Legt die Lage des Nullpunkts prozentual zur Balkenhöhe fest.

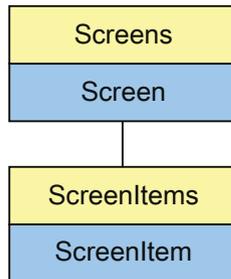
* im ES nicht sichtbar

Tabelle 3-6 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

BatteryView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Ladezustand" dar. Das BatteryView-Objekt ist Element der ScreenItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	RT Advanced
P	RT Professional

Tabelle 3-7 Eigenschaften

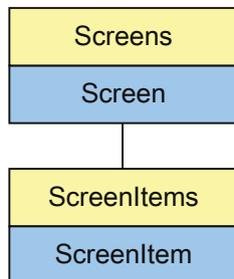
Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-8 Methoden

Methoden	Gült.	Beschreibung
		Nicht gefunden

Button (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Schaltfläche" dar. Das Button-Objekt ist Element der ScreenItems-Auflistung.

Die Verfügbarkeit der folgenden Objekteigenschaften hängen vom gewählten Modus der "Schaltfläche" ab:

Eigenschaft	"Text"	"Textliste"	Grafik"
TextOff	x	--	--
TextOn	x	--	--

Typkennzeichnung in VBS

HMIButton

Abkürzung	Gültigkeit
Pa	Panels
A	RT Advanced
P	RT Professional

Tabelle 3-9 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AdaptBorder (Seite 1553)	P	P	Legt fest, ob sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BitNumber (Seite 1585)	-	-	Legt das Bit fest, dessen Zustand sich ändern muss, um eine Wertänderung auszulösen.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderBrightColor3D (Seite 1589)	P	P	Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderShadeColor3D (Seite 1599)	P	P	Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderStyle3D	A	A	Legt fest, ob das Objekt eine 3D-Rahmenshattierung besitzt.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
BorderWidth3D (Seite 1603)	P	P	Legt die Breite des Rahmens für die 3D-Darstellung des angegebenen Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	P	P	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P, A	P, A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
FocusColor (Seite 1668)	A	A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth (Seite 1669)	A	A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P, A	P, A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HelpText	A	-	Gibt den Tooltip zurück, der in Runtime als Bedienungshilfe für das angegebene Objekt angezeigt wird.
HorizontalAlignment (Seite 1694)	P, A	P, A	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
HotKey (Seite 1696)	-	-	Legt den Hotkey fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienenden fest.
Mode (Seite 1747)	P	P	Legt den Feldtyp des angegebenen Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
PictureAlignment (Seite 1774)	P	P	Legt die Darstellungsart des Hintergrundbildes im Prozessbild fest oder gibt sie zurück.

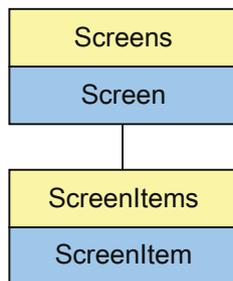
Eigenschaften	Read	Write	Beschreibung
PictureList	-	-	Legt die Grafikliste fest, die das Objekt mit Werten versorgt.
PictureOff (Seite 1775)	P	P	Legt das Bild fest, das im Zustand "Aus" angezeigt wird.
PictureOn (Seite 1776)	P	P	Legt das Bild fest, das im Zustand "Ein" angezeigt wird.
Pressed (Seite 1778)	P	P	Legt fest, ob das angegebene Objekt gedrückt dargestellt ist.
ProcessValue	-	-	Legt die Voreinstellung für den anzuzeigenden Wert fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
StyleSettings (Seite 1849)	P	P	Legt fest, in welchem Stil das Objekt dargestellt wird.
TextList (Seite 1859)	-	-	Eine Auflistung, welche die Zuordnungen zwischen dem Ausgabe- wert und dem tatsächlich auszugebenden Ausgabertext enthält.
TextOff (Seite 1860)	P, A	P, A	Legt den Text fest, der im Zustand "Aus" des angegebenen Objekts angezeigt wird.
TextOn	A	A	Legt den Text fest, der im Zustand "Ein" des angegebenen Objekts angezeigt wird.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
Toggle (Seite 1888)	P	P	Legt fest, ob das angegebene Objekt in Runtime nach dem Betätigen einrastet.
ToolTipText (Seite 1908)	P	P, A	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
TransparentColorPictureOff (Seite 1915)	P	P	Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Aus" auf "transparent" gesetzt wird.
TransparentColorPictureOn (Seite 1916)	P	P	Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Ein" auf "transparent" gesetzt wird.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
UseTransparentColorPictureOff (Seite 1961)	P	P	Legt fest, ob die mit der Eigenschaft "TransparentColorPictureOff" festgelegte Transparentfarbe für den Zustand "Aus" verwendet wird.
UseTransparentColorPictureOn (Seite 1962)	P	P	Legt fest, ob die mit der Eigenschaft "TransparentColorPictureOn" festgelegte Transparentfarbe für den Zustand "Ein" verwendet wird.
VerticalAlignment (Seite 1971)	P, A	P, A	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.
WindowsStyle (Seite 1988)	P	P	Legt fest, ob das Objekt im allgemeinen Windows-Stil dargestellt wird.

Tabelle 3-10 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

ChannelDiagnose (RT Professional)

Beschreibung



Stellt das Objekt "Kanaldiagnose" dar. Das ChannelDiagnose-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIChannelDiagnose

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-11 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Flashing *	-	-	Legt fest, dass der Textblock blinkt.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

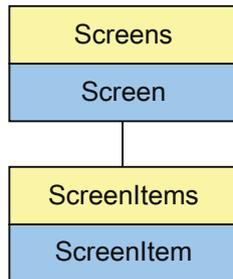
* im ES nicht sichtbar

Tabelle 3-12 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

CheckBox (RT Professional)

Beschreibung



Stellt das Objekt "Kontrollkästchen" dar. Das CheckBox-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMICheckBox

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-13 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AdaptBorder (Seite 1553)	P	P	Legt fest, ob sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CheckmarkAlignment (Seite 1616)	P	P	Legt fest, ob die Felder rechtsbündig angeordnet sind.
CheckmarkCount (Seite 1616)	P	P	Legt die Anzahl der Felder fest.
CornerStyle (Seite 1628)	-	-	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	-	-	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

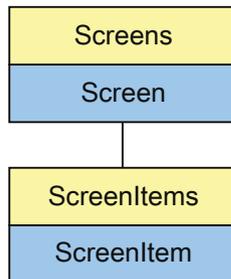
Eigenschaften	Read	Write	Beschreibung
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P	P	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalAlignment (Seite 1694)	P	P	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Index (Seite 1698)	P	P	Legt den Hintergrund der Raster-Steuerelemente fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	-	-	Legt die Form der Linienenden fest.
LogOperation (Seite 1725)	P	P	Legt fest, ob nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.
Name (Seite 1749)	P	P	Gibt den Objektnamen als STRING zurück.
ProcessValue (Seite 1779)	P	P	Legt die Voreinstellung für den anzuzeigenden Wert fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowBadTagState (Seite 1817)	P	P	Legt fest, ob das Objekt grau dargestellt wird, wenn die verbundene Variable einen schlechten QualityCode oder Variablenstatus besitzt.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
Text (Seite 1859)	P	P	Legt die Beschriftung für das Textfeld fest.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
Texts	-	-	Legt die Texte für die Kontrollkästchen fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
VerticalAlignment (Seite 1971)	P	P	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-14 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Circle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Kreis" dar. Das Circle-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMICircle

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-15 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P, A	P, A	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P, A	P, A	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P, A	P, A	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	A	A	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form des Linienendes fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
Radius	P	P, A	Legt den Radius des angegebenen Objekts "Circle" fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.

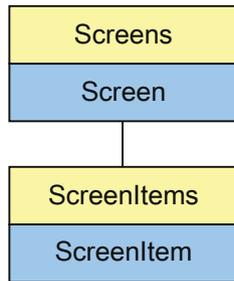
Eigenschaften	Read	Write	Beschreibung
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-16 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

CircleSegment (RT Professional)

Beschreibung



Stellt das Objekt "Kreissegment" dar. Das CircleSegment-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMICircleSegment

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-17 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.

Eigenschaften	Read	Write	Beschreibung
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle	P	P	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndAngle (Seite 1647)	P	P	Legt den Winkel fest, um den der Endpunkt des angegebenen Objekts von der Nulllage (0°) abweicht.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
Radius	P	P	Legt den Radius des angegebenen Objekts "Circle" fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
StartAngle (Seite 1833)	P	P	Legt das SQL-Statement fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.

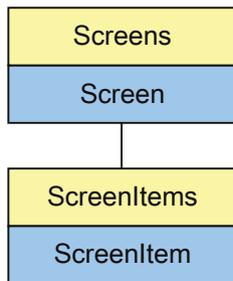
Eigenschaften	Read	Write	Beschreibung
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-18 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

CircularArc (RT Professional)

Beschreibung



Stellt das Objekt "Kreisbogen" dar. Das CircularArc-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMICircularArc

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-19 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Color (Seite 1618)	P	P	Legt die Linienfarbe des angegebenen Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndAngle (Seite 1647)	P	P	Legt den Winkel fest, um den der Endpunkt des angegebenen Objekts von der Nulllage (0°) abweicht.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form des Linienendes fest.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
Radius	P	P	Legt den Radius des angegebenen Objekts "Circle" fest.
StartAngle (Seite 1833)	P	P	Legt das SQL-Statement fest.
Style (Seite 1848)	P	P	Legt den Linienstil des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

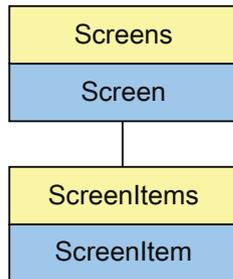
Eigenschaften	Read	Write	Beschreibung
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-20 Methoden

Methoden	Güt.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Clock (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Uhr" dar. Das Clock-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

Clock

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-21 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Analog (Seite 1562)	P	P	Legt fest, ob die Uhr als Analoguhr dargestellt wird.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
DialColor (Seite 1636)	P, A	P, A	Legt die Farbe des Zifferblattes im angegebenen Objekt fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.

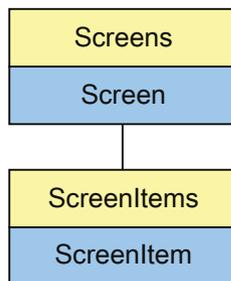
Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HourNeedleHeight (Seite 1696)	P, A	P, A	Legt die Länge des Stundenzeigers im Objekt "Clock" fest.
HourNeedleWidth (Seite 1697)	P, A	P, A	Legt die Breite des Stundenzeigers im Objekt "Clock" fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LockSquaredExtent (Seite 1725)	P	P	Legt fest, ob bei einem Bildaufschlag die Variablenwerte für den darzustellenden Zeitbereich aus den Archiven geladen werden.
MinuteNeedleHeight (Seite 1746)	P, A	P, A	Legt die Länge des Minutenzeigers im Objekt "Clock" fest.
MinuteNeedleWidth (Seite 1746)	P, A	P, A	Legt die Breite des Minutenzeigers im Objekt "Clock" fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
NeedleBorderColor (Seite 1751)	P, A	P, A	Legt den Namen der Datenquelle fest.
NeedleColor (Seite 1752)	P, A	P, A	Legt die Zeigerfarbe im Objekt "Clock" fest.
NeedleFillStyle (Seite 1753)	P	P	Legt die Füllart der Zeiger fest.
Picture (Seite 1774)	-	-	Legt das Bild fest, das im Grafik-Objekt in Runtime angezeigt wird.
SecondNeedleHeight (Seite 1800)	P, A	P, A	Legt die Länge des Sekundenzeigers im Objekt "Clock" fest.
SecondNeedleWidth (Seite 1800)	P, A	P, A	Legt die Breite des Sekundenzeigers im Objekt "Clock" fest.
ShowFocusRectangle (Seite 1820)	P	P	Legt fest, ob in Runtime die Schaltfläche einem Selektionsrahmen erhält, wenn sie aktiviert wird.
ShowTicks (Seite 1828)	P, A	P, A	Legt fest, ob die Teilstriche in der Skala des Objekts angezeigt werden.
TicksColor (Seite 1862)	P, A	P, A	Legt die Darstellung der Skala fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-22 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Connector (RT Professional)

Beschreibung



Stellt das Objekt "Verbinder" dar. Das Connector-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIConnector

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-23 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderEndStyle (Seite 1592)	P	P	Legt die Art der Linienenden des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Color (Seite 1618)	P	P	Legt die Linienfarbe des angegebenen Objekts fest.
ConnectionType (Seite 1626)	P	P	Legt die Art des Verbinders fest. Sie können zwischen zwei Verbindungstypen wählen.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndStyle (Seite 1647)	-	-	Legt fest, wie das Linienende des angegebenen Objekts dargestellt wird.
FirstConnectedObjectIndex (Seite 1660)	P	P	Legt die Indexnummer des oberen Verbinderpunktes fest.
FirstConnectedObjectName (Seite 1661)	P	P	Legt den Objektnamen des Objekts fest, das am oberen Verbinderpunkt angedockt ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
LastConnectedObjectIndex (Seite 1704)	P	P	Legt die Indexnummer des Verbindungspunkts zum letzten, verbundene Objekts fest.
LastConnectedObjectName (Seite 1705)	P	P	Legt den Objektnamen des Objekts fest, das am unteren Verbinderpunkt angedockt ist.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P	P	Gibt den Objektnamen als STRING zurück.
Points *	-	-	Legt die Eckpunkte fest.
StartStyle	-	-	Legt fest, wie der Linienanfang des angegebenen Objekts dargestellt wird.
Style (Seite 1848)	P	P	Legt den Linienstil des angegebenen Objekts fest.

Eigenschaften	Read	Write	Beschreibung
SwapFirstWithLastConnection (Seite 1849)	P	P	Legt fest, ob der Text im Objekt horizontal dargestellt wird.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, dass die Schriftgröße aus der Datenquelle benutzt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

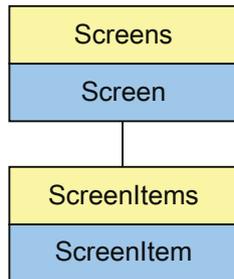
* im ES nicht sichtbar

Tabelle 3-24 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

DateTimeField (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Datum/Uhrzeit-Feld" dar. Das DateTimeField-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIDateTimeField

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-25 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle	-	-	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587) *	-	-	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor	A	A	Legt die Linienfarbe des Objekts fest.

Eigenschaften	Read	Write	Beschreibung
BorderStyle3D (Seite 1600)	A	A	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
DisplaySystemTime	-	-	Legt fest, dass die Systemzeit angezeigt wird.
EdgeStyle	-	-	Legt den Linienstil des angegebenen Objekts fest.
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
Font (Seite 1669) *	-	-	Legt die Schriftart fest oder gibt sie zurück.
ForeColor	A	A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Format	-	-	Legt fest, dass das Format des Textblocks angezeigt wird.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HelpText (Seite 1685)	A	-	Gibt den Tooltip zurück, der in Runtime als Bedienhilfe für das angegebene Objekt angezeigt wird.
HorizontalAlignment	A	A	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
LineWrap *	-	-	Legt den Zeilenumbruch innerhalb des Objekts fest.
LongDateTimeFormat	-	-	Legt fest, dass Datum und Uhrzeit im Langformat angezeigt werden.
Mode (Seite 1747)	-	-	Legt den Feldtyp des angegebenen Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
ProcessValue (Seite 1779)	A	A	Legt die Voreinstellung für den anzuzeigenden Wert fest.
ShowDate	-	-	Legt fest, dass das Datum angezeigt wird.
ShowTime	-	-	Legt fest, dass nur die Uhrzeit angezeigt wird.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
VerticalAlignment (Seite 1971)	A	A	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974) *	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

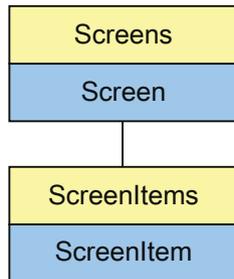
* im ES nicht sichtbar

Tabelle 3-26 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.

DiskSpaceView (RT Professional)

Beschreibung



Stellt das Objekt "Speicherplatzanzeige" dar. Das DiskSpaceView-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

IXDiskSpaceView

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-27 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Alarm (Seite 1555)	P	P	Legt den Grenzwert für die Speicherplatzanzeige fest, ab dem eine Meldung ausgegeben wird.
AlarmColor (Seite 1556)	P	P	Legt fest, welche die Art von Meldungen in der Meldeanzeige angezeigt werden.
Drive (Seite 1642)	P	P	Legt den Buchstaben des Laufwerks fest, das überwacht werden soll.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.

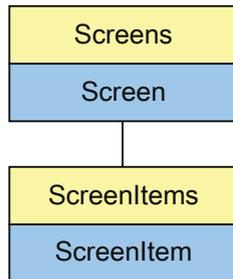
Eigenschaften	Read	Write	Beschreibung
Free (Seite 1676)	P	P	Gibt die Größe des freien Speicherplatzes zurück.
FreePercent (Seite 1677)	P	P	Gibt die Messwerte für den freien Speicherplatz in Prozent zurück.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Interval (Seite 1701)	P	P	Legt die Zeitintervalle fest, zum Aktualisieren der angezeigten Messwerte.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
NormalColor	P	P	Legt die Farbe des Objekts im Normalbereich fest.
Tolerance (Seite 1889)	P	P	Legt den Grenzwert für die Speicherplatzanzeige fest, ab dem eine Abweichung gemeldet wird.
ToleranceColor (Seite 1889)	P	P	Legt die Farben fest, in denen der Balken der Speicherplatzanzeige dargestellt wird, sobald der Toleranzbereich überschritten ist.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Total (Seite 1912)	P	P	Gibt die Speicherkapazität zurück.
Used (Seite 1953)	P	P	Legt fest, ob das Scrollen aktiviert ist.
UsedPercent (Seite 1957)	P	P	Gibt die Messwerte für den belegten Speicherplatz in Prozent zurück.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Warning (Seite 1976)	P	P	Legt den Grenzwert für die Speicherplatzanzeige fest, ab dem eine Warnung ausgegeben wird.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-28 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Ellipse (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Ellipse" dar. Das Ellipse-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIEllipse

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-29 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P, A	P, A	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P, A	P, A	Legt die Linienfarbe des Objekts fest.

Eigenschaften	Read	Write	Beschreibung
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P, A	P, A	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle	A	A	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
RadiusHeight	P	P	Legt die Nebenachse des Objekts "Ellipse" fest.
RadiusWidth	P	P	Legt die Hauptachse des Objekts "Ellipse" fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.

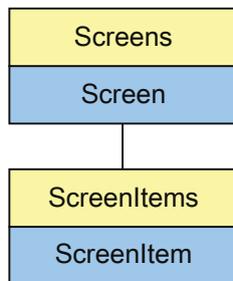
Eigenschaften	Read	Write	Beschreibung
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-30 Methode

Methode	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

EllipseSegment (RT Professional)

Beschreibung



Stellt das Objekt "Ellipsensegment" dar. Das EllipseSegment-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIEllipseSegment

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-31 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	P	P	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndAngle (Seite 1647)	P	P	Legt den Winkel fest, um den der Endpunkt des angegebenen Objekts von der Nulllage (0°) abweicht.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

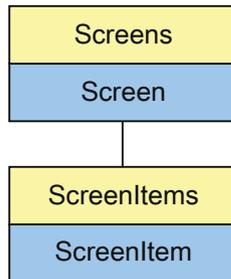
Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
RadiusHeight	P	P	Legt die Nebenachse des Objekts "Ellipse" fest.
RadiusWidth	P	P	Legt die Hauptachse des Objekts "Ellipse" fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
StartAngle (Seite 1833)	P	P	Legt das SQL-Statement fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-32 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

EllipticalArc (RT Professional)

Beschreibung



Stellt das Objekt "Ellipsenbogen" dar. Das EllipticalArc-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIEllipticalArc

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-33 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Color (Seite 1618)	P	P	Legt die Linienfarbe des angegebenen Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

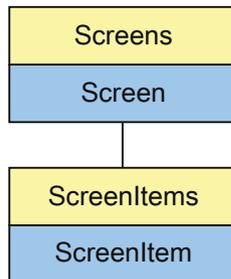
Eigenschaften	Read	Write	Beschreibung
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Liniestärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndAngle (Seite 1647)	P	P	Legt den Winkel fest, um den der Endpunkt des angegebenen Objekts von der Nulllage (0°) abweicht.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	-	-	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	-	-	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Liniende fest.
LineWidth (Seite 1723)	P	P	Legt die Liniestärke des angegebenen Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
RadiusHeight	P	P	Legt die Nebenachse des Objekts "Ellipse" fest.
RadiusWidth	P	P	Legt die Hauptachse des Objekts "Ellipse" fest.
StartAngle (Seite 1833)	P	P	Legt das SQL-Statement fest.
Style (Seite 1848)	P	P	Legt den Linienstil des angegebenen Objekts fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-34 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

ForeignControl (RT Professional)

Beschreibung



Objekt-Typ des ScreenItem-Objektes. Repräsentiert das Grafikobjekt "Control".

Der Objekt-Typ Control nimmt immer die Eigenschaften des von Ihnen ausgewählten Control-Typs an. Bei von WinCC bereitgestellten Controls finden Sie die Eigenschaften unter der Beschreibung der entsprechenden Controls.

Bei Controls von Fremdanbietern werden die Eigenschaften von den Controls mitgebracht und sind damit nicht Bestandteil dieser Beschreibung. Sie können die Control-Eigenschaften jedoch über die Eigenschaft "Item" abfragen.

Typkennzeichnung in VBS

Spezielle WinCC-Typbezeichnung oder versionsunabhängige ProgID

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "Control1" um 10 Pixel nach rechts verschoben:

```
'VBS36  
Dim objControl  
Set objControl = ScreenItems("Control1")  
objControl.Left = objControl.Left + 10
```

Besonderheit

Bei von WinCC bereitgestellten Controls wird eine spezielle Kennung als Typ zurückgeliefert. Sie ist unter der Rubrik "Typkennzeichnung in VBS" bei den Einzelbeschreibungen der WinCC-Controls zu finden.

Zugriff auf spezifische Eigenschaften anstelle von Ereignis-Parametern des Fremdanbieter-Controls

Die Parameterliste eines Ereignisses bei Fremdanbieter-Controls wird in WinCC nicht ausgelesen. Um auf eigene Parameter von Control-Ereignissen zugreifen zu können, implementieren Sie im Control Eigenschaften anstelle von Parametern.

Das folgende Code-Beispiel zeigt den Quellcode des Controls "UserControlTestModify", in dem die Eigenschaften "Title" und "Description" anstelle von Parametern definiert sind.

```
public partial class UserControlTestModify : UserControl
{
    public event MyEventHandler MyEvent = null;

    public string Title
    {
        get;
        private set;
    }

    public string Description
    {
        get;
        set;
    }

    public UserControlTestModify()
    {
        InitializeComponent();
    }
    private void OnMyEvent(Exception e)
    {
        if (this.MyEvent!= null)
        {
            // Place place data in Properties before invoking event.
            Title = "title";
            Description = "description";
            this.MyEvent.Invoke();
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.OnMyEvent (null);
    }
}
```

Die spezifischen Eigenschaften eines Fremdanbieter-Controls werden in WinCC mit IntelliSense nicht angezeigt.

Das folgende Beispiel zeigt, wie Sie in WinCC auf die spezifische Eigenschaft "Title" des Fremdanbieter-Controls "FControl" zugreifen:

```
HMIRuntime.Screens("StartScreen").ScreenItems("FControl").Title = "MyForeignControl"
```

ProgID und UserfriendlyName von Fremdanbieter-Controls

Bei WinCC-fremden Controls wird die versionsunabhängige ProgID als Typ zurückgeliefert.

Aus der ProgID können Sie die versionsabhängige ProgID oder den "User friendlyName" ermitteln: In folgendem Beispiel ist "Control1" ein im Bild eingebettetes Control, das über die Type-Eigenschaft bereits die versionsunabhängige ProgID zurückgibt.

Hinweis

Da nicht jedes Control eine versionsabhängige ProgID besitzt, sollte zum Abfragen der versionsabhängigen ProgID oder des UserFriendlyName eine Fehlerbehandlung eingebaut werden. Wird keine Fehlerbehandlung verwendet, wird der Code sofort ohne Ergebnis beendet, wenn keine ProgID gefunden wird.

Ermitteln Sie die versionsabhängige ProgID wie folgt:

```
'VBS37
Dim objControl
Dim strCurrentVersion
Set objControl = ScreenItems("Control1")
strCurrentVersion = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type &
"\CurVer\")
MsgBox strCurrentVersion
```

Hinweis

Damit obiges Beispiel funktioniert, sollte ein Multimediacontrol ins Bild eingefügt werden.

Ermitteln Sie den UserFriendlyName wie folgt:

```
'VBS38
Dim objControl
Dim strFriendlyName
Set objControl = ScreenItems("Control1")
strFriendlyName = CreateObject("WScript.Shell").RegRead("HKCR\" & objControl.Type & "\")
MsgBox strFriendlyName
```

Hinweis

Damit obiges Beispiel funktioniert, sollte ein Multimediacontrol ins Bild eingefügt werden.

Wenn Sie ein nicht von WinCC bereitgestelltes Control verwenden, kann es vorkommen, dass die vom Control mitgebrachten Eigenschaften namensgleich sind mit den allgemeinen ScreenItem-Eigenschaften. In diesem Fall haben die ScreenItem-Eigenschaften Vorrang. Auf die "verdeckten" Eigenschaften eines Fremdanbieter-Controls kann über die zusätzliche Eigenschaft "object" zugegriffen werden. Sprechen Sie die Eigenschaften eines Fremdanbieter-Controls also z.B. in folgender Form an:

```
Control.object.type
```

Bei Namensgleichheit werden die Eigenschaften des ScreenItem-Objektes verwendet., wenn Sie die folgende Form verwenden:

`Control.type`

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-35 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
OCXState (Seite 1757) *	-	-	Legt den Status des OCX fest.
ProgID (Seite 1780) *	-	-	Bei WinCC-fremden Controls wird die versionsunabhängige ProgID als Typ zurückgeliefert.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

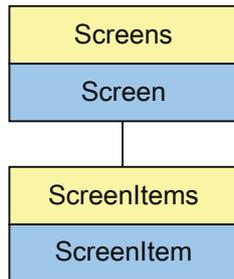
* im ES nicht sichtbar

Tabelle 3-36 Methoden

Methoden	Gült.	Beschreibung

FunctionTrendControl (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "f(x)-Kurvenanzeige" dar. Das FunctionTrendControl-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIFunctionTrendControl

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-37 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Height (Seite 1680)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
ObjectName (Seite 2036)	P	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.

Eigenschaften	Read	Write	Beschreibung
Type (Seite 2140)	P	-	Gibt den Typ des angegebenen Objekts als STRING zurück.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-38 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
AttachDB (Seite 2163)	P, A	Führt die Funktion der Schaltfläche "Backup verbinden" des Controls aus.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.
DetachDB (Seite 2169)	P, A	Führt die Funktion der Schaltfläche "Backup trennen" des Controls aus.
Export (Seite 2170)	P, A	Führt die Funktion der Schaltfläche "Archiv exportieren" oder "Daten exportieren" des Controls aus.
GetStatusBarElement (Seite 2197)	P, A	Gibt das mit Namen oder Index bezeichnete Element der Statuszeile des Controls als Typ "ICCAxStatusBarElement" zurück.
GetStatusBarElementCollection (Seite 2198)	P, A	Gibt die Auflistung aller Elemente der Statuszeile des Controls als Typ "ICCAxCollection" zurück.
GetToolBarButton (Seite 2205)	P, A	Gibt die mit Namen oder Index bezeichnete Tastenfunktion der Symbolleiste des Controls als Typ "ICCAxToolBarButton" zurück.
GetToolBarButtonCollection (Seite 2207)	P, A	Gibt die Auflistung aller Tastenfunktionen der Symbolleiste des Controls als Typ "ICCAxCollection" zurück.
GetTrend (Seite 2208)	P, A	Gibt die mit Namen oder Index bezeichnete Kurve der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxTrend" bzw. "ICCAxFunctionTrend" zurück.
GetTrendCollection (Seite 2209)	P, A	Gibt die Auflistung aller Kurven der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
GetTrendWindow (Seite 2210)	P, A	Gibt das mit Namen oder Index bezeichnete Kurvenfenster-Objekt der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxTrendWindow" zurück.
GetTrendWindowCollection (Seite 2211)	P, A	Gibt die Auflistung aller Kurvenfenster-Objekte der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
GetXAxis (Seite 2218)	P, A	Gibt das mit Namen oder Index bezeichnete X-Achsen-Objekt der f(x)-Kurvenanzeige als Typ "ICCAxValueAxis" zurück.
GetXAxisCollection (Seite 2219)	P, A	Gibt die Auflistung aller X-Achsen-Objekte der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
GetYAxis (Seite 2220)	P, A	Gibt das mit Namen oder Index bezeichnete Y-Achsen-Objekt der f(x)-Kurvenanzeige als Typ "ICCAxValueAxis" zurück.
GetYAxisCollection (Seite 2221)	P, A	Gibt die Auflistung aller Y-Achsen-Objekte der f(x)-Kurvenanzeige vom Typ "ICCAxCollection" zurück.
MoveAxis (Seite 2226)	P, A	Führt die Funktion der Schaltfläche "Achsenbereich verschieben" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Methoden	Gült.	Beschreibung
NextTrend (Seite 2233)	P, A	Führt die Funktion der Schaltfläche "Nächste Kurve" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
OneToOneView (Seite 2234)	P, A	Führt die Funktion der Schaltfläche "Originalansicht" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
PreviousTrend (Seite 2235)	P, A	Führt die Funktion der Schaltfläche "Vorhergehende Kurve" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
Print (Seite 2236)	P, A	Führt die Funktion der Schaltfläche "Drucken" des Controls aus.
ShowHelp (Seite 2255)	P, A	Führt die Funktion der Schaltfläche "Hilfe" des Controls aus.
ShowPropertyDialog (Seite 2260)	P, A	Führt die Funktion der Schaltfläche "Konfigurationsdialog" des Controls aus.
ShowTagSelection (Seite 2264)	P, A	Führt die Funktion der Schaltfläche "Datenanbindung wählen" des Controls aus.
ShowTimeSelection (Seite 2265)	P, A	Führt die Funktion der Schaltfläche "Zeitbereich wählen" des Controls aus.
ShowTrendSelection (Seite 2265)	P, A	Führt die Funktion der Schaltfläche "Kurven wählen" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
StartStopUpdate (Seite 2266)	P, A	Führt die Funktion der Schaltfläche "Start" oder "Stopp" des Controls aus.
ZoomArea (Seite 2274)	P, A	Führt die Funktion der Schaltfläche "Zoomen Ausschnitt" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
ZoomInOut (Seite 2274)	P, A	Führt die Funktion der Schaltfläche "Zoomen +/-" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
ZoomInOutX (Seite 2276)	P, A	Führt die Funktion der Schaltfläche "X-Achse zoomen +/-" der f(x)-Kurvenanzeige aus.
ZoomInOutY (Seite 2276)	P, A	Führt die Funktion der Schaltfläche "Y-Achse zoomen +/-" der f(x)-Kurvenanzeige aus.
ZoomMove (Seite 2277)	P, A	Führt die Funktion der Schaltfläche "Kurvenbereich verschieben" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Siehe auch

[BackColor \(Seite 1573\)](#)
[Caption \(Seite 1608\)](#)
[LoadDataImmediately \(Seite 1724\)](#)
[Online \(Seite 1757\)](#)
[TimeBase \(Seite 1866\)](#)
[ShowRuler \(Seite 1823\)](#)
[ApplyProjectSettings \(Seite 1564\)](#)
[BorderColor \(Seite 1590\)](#)
[BorderWidth \(Seite 1601\)](#)
[Closeable \(Seite 1617\)](#)
[ConnectTrendWindows \(Seite 1626\)](#)
[ExportDirectoryChangeable \(Seite 1650\)](#)
[ExportDirectoryname \(Seite 1651\)](#)

ExportFileExtension (Seite 1651)
ExportFilename (Seite 1652)
ExportFilenameChangeable (Seite 1652)
ExportFormatGuid (Seite 1653)
ExportFormatName (Seite 1654)
ExportParameters (Seite 1654)
ExportSelection (Seite 1655)
ExportShowDialog (Seite 1655)
Font (Seite 1669)
GraphDirection (Seite 1678)
Layer (Seite 1707)
LineColor (Seite 1721)
LineWidth (Seite 1723)
Moveable (Seite 1748)
Name (Seite 1749)
RTPersistence (Seite 1790)
RTPersistenceType (Seite 1790)
UseTrendNameAsLabel (Seite 1962)
ToolbarAlignment (Seite 1895)
ToolbarBackColor (Seite 1895)
ToolbarButtonActive (Seite 1896)
ToolbarButtonAdd (Seite 1897)
ToolbarButtonBeginGroup (Seite 1897)
ToolbarButtonCount (Seite 1898)
ToolbarButtonEnabled (Seite 1899)
ToolbarButtonHotKey (Seite 1899)
ToolbarButtonID (Seite 1900)
ToolbarButtonIndex (Seite 1900)
ToolbarButtonLocked (Seite 1901)
ToolbarButtonName (Seite 1902)
ToolbarButtonAuthorization (Seite 1902)
ToolbarButtonRemove (Seite 1903)
ToolbarButtonRename (Seite 1903)
ToolbarButtonRepos (Seite 1904)
ToolbarButtonTooltipText (Seite 1905)

ToolbarButtonUserDefined (Seite 1905)
ToolbarShowTooltips (Seite 1906)
ToolbarUseBackColor (Seite 1906)
ToolbarUseHotKeys (Seite 1907)
ToolbarVisible (Seite 1908)
TrendActualize (Seite 1917)
TrendAdd (Seite 1917)
TrendBeginTime (Seite 1918)
TrendColor (Seite 1918)
TrendCount (Seite 1919)
TrendEndTime (Seite 1919)
TrendExtendedColorSet (Seite 1920)
TrendFill (Seite 1920)
TrendFillColor (Seite 1921)
TrendIndex (Seite 1922)
TrendLabel (Seite 1923)
TrendLineStyle (Seite 1923)
TrendLineType (Seite 1924)
TrendLineWidth (Seite 1925)
TrendLowerLimit (Seite 1925)
TrendLowerLimitColor (Seite 1926)
TrendLowerLimitColoring (Seite 1926)
TrendMeasurePoints (Seite 1927)
TrendName (Seite 1927)
TrendPointColor (Seite 1928)
TrendPointSize (Seite 1928)
TrendPointWidth (Seite 1929)
TrendProvider (Seite 1929)
TrendRangeType (Seite 1930)
TrendRemove (Seite 1931)
TrendRename (Seite 1931)
TrendRepos (Seite 1932)
TrendSelectTagNameX (Seite 1932)
TrendSelectTagNameY (Seite 1933)
TrendTagNameX (Seite 1933)

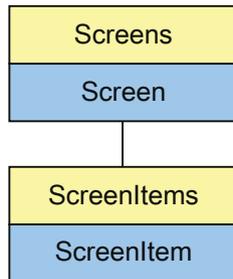
TrendTagNameY (Seite 1934)
TrendTimeRangeBase (Seite 1934)
TrendTimeRangeFactor (Seite 1935)
TrendTrendWindow (Seite 1935)
TrendUncertainColor (Seite 1936)
TrendUncertainColoring (Seite 1936)
TrendUpperLimit (Seite 1937)
TrendUpperLimitColor (Seite 1937)
TrendUpperLimitColoring (Seite 1938)
TrendVisible (Seite 1939)
TrendWindowAdd (Seite 1939)
TrendWindowCoarseGrid (Seite 1940)
TrendWindowCoarseGridColor (Seite 1940)
TrendWindowFineGrid (Seite 1941)
TrendWindowFineGridColor (Seite 1941)
TrendWindowForegroundTrendGrid (Seite 1942)
TrendWindowGridInTrendColor (Seite 1942)
TrendWindowHorizontalGrid (Seite 1943)
TrendWindowIndex (Seite 1943)
TrendWindowName (Seite 1944)
TrendWindowRemove (Seite 1944)
TrendWindowRename (Seite 1945)
TrendWindowRepos (Seite 1945)
TrendWindowRulerColor (Seite 1946)
TrendWindowRulerLayer (Seite 1947)
TrendWindowRulerStyle (Seite 1947)
TrendWindowRulerWidth (Seite 1948)
TrendWindowSpacePortion (Seite 1948)
TrendWindowVerticalGrid (Seite 1949)
TrendWindowVisible (Seite 1949)
TrendXAxis (Seite 1950)
TrendYAxis (Seite 1950)
ShowRulerInAxis (Seite 1823)
ShowScrollbars (Seite 1824)
ShowTitle (Seite 1829)

ShowTrendIcon (Seite 1830)
StatusbarBackColor (Seite 1836)
StatusbarElementAdd (Seite 1836)
StatusbarElementAutoSize (Seite 1837)
StatusbarElementCount (Seite 1838)
StatusbarElementIconId (Seite 1838)
StatusbarElementID (Seite 1839)
StatusbarElementIndex (Seite 1840)
StatusbarElementName (Seite 1840)
StatusbarElementRemove (Seite 1841)
StatusbarElementRename (Seite 1842)
StatusbarElements (Seite 1842)
StatusbarElementTooltipText (Seite 1842)
StatusbarElementUserDefined (Seite 1843)
StatusbarElementVisible (Seite 1844)
StatusbarElementWidth (Seite 1844)
StatusbarFontColor (Seite 1845)
StatusbarShowTooltips (Seite 1846)
StatusbarText (Seite 1846)
StatusbarUseBackColor (Seite 1847)
StatusbarVisible (Seite 1848)
XAxisAdd (Seite 1989)
XAxisAutoPrecisions (Seite 1990)
XAxisAutoRange(i) (Seite 1990)
XAxisBeginValue (Seite 1991)
XAxisColor (Seite 1991)
XAxisCount (Seite 1992)
XAxisEndValue (Seite 1992)
XAxisExponentialFormat (Seite 1993)
XAxisIndex (Seite 1993)
XAxisInTrendColor (Seite 1994)
XAxisName (Seite 1995)
XAxisPrecisions (Seite 1995)
XAxisRemove (Seite 1995)
XAxisRepos (Seite 1996)

XAxisTrendWindow (Seite 1997)
XAxisVisible (Seite 1997)
YAxisAdd (Seite 1998)
YAxisAutoPrecisions (Seite 1999)
YAxisAutoRange (Seite 1999)
YAxisBeginValue (Seite 2000)
YAxisColor (Seite 2000)
YAxisCount (Seite 2001)
YAxisEndValue (Seite 2001)
YAxisExponentialFormat (Seite 2002)
YAxisIndex (Seite 2002)
YAxisInTrendColor (Seite 2003)
YAxisLabel (Seite 2003)
YAxisName (Seite 2004)
YAxisPrecisions (Seite 2004)
YAxisRemove (Seite 2004)
YAxisRepos (Seite 2005)
YAxisScalingType (Seite 2006)
YAxisTrendWindow (Seite 2006)
YAxisVisible (Seite 2007)
ControlDesignMode (Seite 1831)
RTPersistencePasswordLevel (Seite 2046)

Gauge (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Zeigerinstrument" dar. Das Gauge-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMI Gauge

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-39 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AngleMax (Seite 1563)	P, A	P, A	Legt den Winkel für das Skalende des Objekts "Gauge" fest.
AngleMin (Seite 1563)	P, A	P, A	Legt den Winkel für den Skalenanfang des Objekts "Gauge" fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackPicture	-	-	Legt eine Grafik für den Hintergrund fest.

Eigenschaften	Read	Write	Beschreibung
BorderInnerStyle3D (Seite 1597)	P	P	Legt die Darstellung des inneren Teils des Objektrahmens fest.
BorderOuterStyle3D (Seite 1598)	P	P	Legt die Darstellung des äußeren Teils des Objektrahmens fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
BorderWidth3D (Seite 1603)	P	P	Legt die Breite des Rahmens für die 3D-Darstellung des angegebenen Objekts fest.
CaptionColor (Seite 1609)	P, A	P, A	Legt die Farbe des Texts fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CaptionFont	-	-	Legt den Zeichensatz für die Beschriftung fest.
CaptionText (Seite 1610)	P, A	P, A	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CaptionTop (Seite 1611)	P	P	Legt den Abstand der Instrumentenbeschriftung zur Oberkante des angegebenen Objekts fest.
CenterColor (Seite 1614)	P, A	P, A	Legt die Farbe des Mittelpunktes für das Objekt "Gauge" fest.
CenterSize (Seite 1615)	P	P	Legt den Durchmesser des runden Skalenmittelpunkts fest.
DangerRangeColor (Seite 1632)	P, A	P, A	Legt die Farbe des Gefahrenbereichs auf der Skala des Objekts "Gauge" fest.
DangerRangeStart (Seite 1633)	P, A	P, A	Legt fest, ab welchem Skalenwert der Gefahrenbereich des Objekts "Gauge" beginnt.
DangerRangeVisible (Seite 1634)	P, A	P, A	Legt fest, ob der Gefahrenbereich in der Skala des Objekts "Gauge" angezeigt wird. {
DialColor (Seite 1636)	P, A	P, A	Legt die Farbe des Zifferblattes im angegebenen Objekt fest.
DialFillStyle (Seite 1636)	P	P	Legt die Art des Hintergrundes des angegebenen Objekts fest.
DialPicture	-	-	Legt eine Grafik für das Zifferblatt fest.
DialSize (Seite 1637)	P	P	Legt den Durchmesser der Skalenscheibe in Bezug auf den kleineren Wert der Geometrieattribute "Width" und "Height" fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Flashing *	-	-	Legt fest, dass der Textblock blinkt.
Gradation (Seite 1677)	P, A	P, A	Legt die Wertdifferenz zwischen zwei Hauptteilstrichen des Objekts "Gauge" fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LockSquaredExtent (Seite 1725)	P	P	Legt fest, ob bei einem Bildaufschlag die Variablenwerte für den darzustellenden Zeitbereich aus den Archiven geladen werden.
MaximumValue (Seite 1729)	P, A	P, A	Legt den Maximalwert der Skala im angegebenen Objekt fest.
MinimumValue (Seite 1745)	P, A	P, A	Legt den Minimalwert der Skala im angegebenen Objekt fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
NormalRangeColor (Seite 1753)	P, A	P, A	Legt die Farbe des Normalbereichs auf der Skala des Objekts "Gauge" fest. "
NormalRangeVisible (Seite 1754)	P, A	P, A	Legt fest, ob der Normalbereich in der Skala des Objekts "Gauge" angezeigt wird.
PointerColor (Seite 1777)	P, A	P, A	Legt die Zeigerfarbe des Objekts "Gauge" fest.
ProcessValue (Seite 1779)	P, A	P, A	Legt die Voreinstellung für den anzuzeigenden Wert fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
ScaleLabelColor (Seite 1793)	P, A	P, A	Legt die Farbe der Beschriftung der Skalenteilung des Objekts "Gauge" fest.
ScaleLabelFont	-	-	Legt die Schriftart für die Skalenbeschriftung fest.
ScaleTickColor (Seite 1794)	P, A	P, A	Legt die Farbe der Skalenteilung des Objekts "Gauge" fest.
ScaleTickLabelPosition (Seite 1795)	P	P	Legt den Durchmesser des gedachten Kreises fest, auf dem sich die Beschriftung der Skalenteilung befindet.
ScaleTickLength (Seite 1795)	P	P	Legt die Länge der Hauptstriche der Skalenteilung fest.
ScaleTickPosition (Seite 1796)	P	P	Legt den Durchmesser des gedachten Kreises fest, auf dem sich die Skalenteilung befindet.
ShowDecimalPoint (Seite 1819)	P	P	Legt fest, ob die Skala mit Dezimalzahlen (Dezimalkomma und eine Dezimalstelle) oder mit ganzen Zahlen beschriftet wird.
ShowPeakValuePointer (Seite 1821)	P, A	P, A	Legt fest, ob für das angegebene Objekt ein Schleppzeiger verwendet wird.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UnitColor (Seite 1951)	P, A	P, A	Legt die Textfarbe für die Bezeichnung der Maßeinheit im Objekt "Gauge" fest.
UnitFont	-	-	Legt die Schriftart für die Einheit fest.
UnitText (Seite 1952)	P, A	P, A	Legt den Text für die Maßeinheit des angegebenen Objekts fest.
UnitTop (Seite 1952)	P	P	Legt den Abstand der Maßeinheit zur Oberkante des angegebenen Objekts fest.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
WarningRangeColor (Seite 1979)	P, A	P, A	Legt die Farbe des Warnbereichs auf der Skala des Objekts "Gauge" fest.
WarningRangeStart (Seite 1980)	P, A	P, A	Legt fest, ab welchem Skalenwert der Warnbereich des Objekts "Gauge" beginnt.
WarningRangeVisible (Seite 1981)	P, A	P, A	Legt fest, ob der Warnbereich in der Skala des Objekts "Gauge" angezeigt wird.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

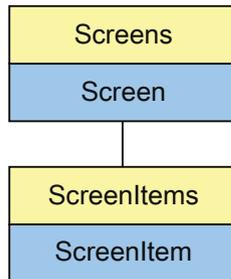
* im ES nicht sichtbar

Tabelle 3-40 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

GraphicIOField (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Grafisches E/A-Feld" dar. Das GraphicIOField-Objekt ist Element der ScreenItems-Auflistung.

Die Verfügbarkeit der folgenden Objekteigenschaften hängen vom gewählten Modus ab:

Eigenschaft	"Eingabe"	"Ausgabe"	"Eingabe/Ausgabe"	"Zwei Zustände"
BorderColor	--	--	--	x
BorderStyle3D	x	x	x	--
Enabled	x	--	x	--
FocusColor	x	--	x	--
FocusWidth	x	--	x	--
HelpText	x	--	x	--
TransparentColor	x	x	x	--
UseTransparent-Color	x	x	x	--

Typkennzeichnung in VBS

HMIGraphicIOField

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-41 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AboveUpperLimitColor (Seite 1545)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Oberer Grenzwert überschritten" fest.
AdaptPicture	-	-	Legt fest, dass die Größe des Objekts in Runtime automatisch angepasst wird.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
AutoSizing	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.
BackColor (Seite 1573)	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BelowLowerLimitColor (Seite 1585)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Unterer Grenzwert unterschritten" fest.
BitNumber (Seite 1585)	-	-	Legt das Bit fest, dessen Zustand sich ändern muss, um eine Wertänderung auszulösen.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderStyle3D	A	A	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	P	P	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P, A	P, A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.

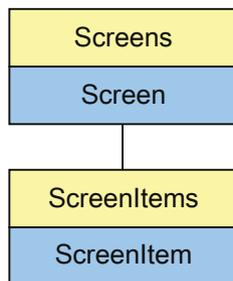
Eigenschaften	Read	Write	Beschreibung
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingOnLimitViolation	-	-	Legt fest, dass Blinken bei Grenzwertverletzung aktiviert ist.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
FlashTransparentColor (Seite 1667)	P	P	Legt die Farbe des Bitmap-Objektes eines Blinkbildes fest, die auf "transparent" gesetzt wird.
FocusColor (Seite 1668)	A	A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth (Seite 1669)	A	A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HelpText	A	-	Gibt den Tooltip zurück, der in Runtime als Bedienungshilfe für das angegebene Objekt angezeigt wird.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienenden fest.
Mode (Seite 1747)	-	-	Legt den Feldtyp des angegebenen Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
OnValue	-	-	Legt den Wert für den Zustand "EIN" im "Zwei-Zustände"-Modus fest.
PictureList	-	-	Legt die Grafikliste fest, die das Objekt mit Werten versorgt.
PictureOff (Seite 1775)	-	-	Legt das Bild fest, das im Zustand "Aus" angezeigt wird.
PictureOn (Seite 1776)	-	-	Legt das Bild fest, das im Zustand "Ein" angezeigt wird.
ProcessValue (Seite 1779)	P	P	Legt die Voreinstellung für den anzuzeigenden Wert fest.
ScrollBarOrientation	-	-	Legt die Orientierung der Bildlaufleiste fest.
ShowScrollBar	-	-	Legt den Typ der Bildlaufleiste fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
TransparentColor (Seite 1914)	P, A	P, A	Legt fest, welche Farbe der zugeordneten Grafik (*.bmp, *.dib) des angegebenen Objekts auf "transparent" gesetzt wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
UseFlashTransparentColor (Seite 1958)	P	P	Legt fest, ob die Farbe des Bitmap-Objektes eines Blinkbildes auf "transparent" gesetzt wird.
UseTransparentColor (Seite 1960)	P, A	P, A	Legt fest, ob die mit der Eigenschaft "TransparentColor" festgelegte Farbe des angegebenen Objekts transparent dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-42 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

GraphicView (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Grafikanzeige" dar. Das GraphicView-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIGraphicView

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-43 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AdaptPicture	A	-	Legt fest, dass die Größe des Objekts in Runtime automatisch angepasst wird.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
AutoSizing	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderStyle3D	A	A	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	P	P	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

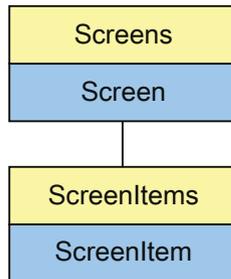
Eigenschaften	Read	Write	Beschreibung
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienenden fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
Picture (Seite 1774)	P	P	Legt das Bild fest, das im Grafik-Objekt in Runtime angezeigt wird.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
TransparentColor (Seite 1914)	P, A	P, A	Legt fest, welche Farbe der zugeordneten Grafik (*.bmp, *.dib) des angegebenen Objekts auf "transparent" gesetzt wird.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
UseTransparentColor (Seite 1960)	P, A	P, A	Legt fest, ob die mit der Eigenschaft "TransparentColor" festgelegte Farbe des angegebenen Objekts transparent dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-44 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Group (RT Professional)

Beschreibung



Stellt das Objekt "Gruppe" dar. Das Group-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIGroup

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-45 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
ForeColor (Seite 1675)	P	P	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.

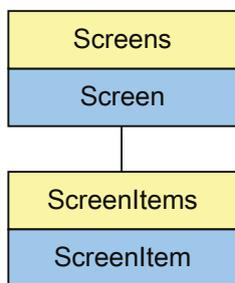
Eigenschaften	Read	Write	Beschreibung
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	-	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-46 Methoden

Methoden	Gült.	Beschreibung

HTMLBrowser (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "HTML-Browser" dar. Das HTMLBrowser-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIHTMLBrowser

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-47 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Address (Seite 1554)	P, A	P, A	Legt die Web-Adresse fest, die im HTML-Browser geöffnet wird.
Enabled (Seite 1644)	P, A	P, A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Flashing *	-	-	Legt fest, dass der Textblock blinkt.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektname als STRING zurück.
ShowStatusBar (Seite 1826)	P	P	Legt fest, ob die Statusleiste angezeigt wird.
ShowToolBar (Seite 1829) *	-	-	Legt fest, ob eine Symbolleiste angezeigt wird.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

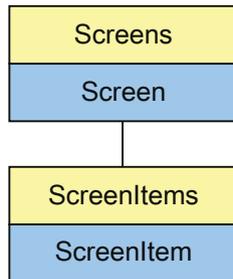
* in ES nicht sichtbar

Tabelle 3-48 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.
SetHtml	??	

IOField (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "E/A-Feld" dar. Das IOField-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIIOField

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-49 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AboveUpperLimitColor (Seite 1545)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Oberer Grenzwert überschritten" fest.
AcceptOnExit (Seite 1546)	P	P	Legt fest, ob das Eingabefeld automatisch bestätigt wird, wenn es verlassen wird.
AcceptOnFull (Seite 1546)	P	P	Legt fest, ob das Eingabefeld automatisch verlassen und bestätigt wird, wenn die festgelegte Anzahl an Werten eingegeben wurde.

Eigenschaften	Read	Write	Beschreibung
AdaptBorder (Seite 1553)	P	P	Legt fest, ob sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.
AskOperationMotive (Seite 1565)	P	P	Legt fest, ob der Grund für die Bedienung dieses Objekts mitprotokolliert wird.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BelowLowerLimitColor (Seite 1585)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Unterer Grenzwert unterschritten" fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P, A	P, A	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderStyle3D	A	A	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
BottomMargin	-	-	Legt den Abstand des Texts zum unteren Objektrand fest.
ClearOnError (Seite 1617)	P	P	Legt fest, ob eine Fehleingabe in diesem Objekt automatisch gelöscht wird.
ClearOnFocus (Seite 1617)	P	P	Legt fest, ob der Feldeintrag gelöscht wird, sobald das E/A-Feld aktiviert wird.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
CursorControl (Seite 1632)	P	P	Legt fest, ob der Mauszeiger nach dem Verlassen des Feldes auf das nächste Feld der TAB-Reihenfolge springt.
DataFormat (Seite 1634)	P	P	Gibt den Datentyp des Objektes IOField zurück.
EdgeStyle (Seite 1642)	A	A	Legt den Linienstil des angegebenen Objekts fest.
EditOnFocus (Seite 1644)	P	P	Legt fest, ob die Dateneingabe sofort möglich ist, wenn das Eingabefeld mit der Taste "Tab" ausgewählt ist.
Enabled (Seite 1644)	P, A	P, A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
FieldLength (Seite 1657)	-	-	Legt fest, dass das Feld "Feldlänge Zeichenkette" nur gelesen werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingOnLimitViolation	-	-	Legt fest, dass Blinken bei Grenzwertverletzung aktiviert ist.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P, A	P, A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
FormatPattern (Seite 1676)	P	P	Legt das Format des Ausgabewerts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HelpText	A	-	Gibt den Tooltip zurück, der in Runtime als Bedienungshilfe für das angegebene Objekt angezeigt wird.
HiddenInput (Seite 1686)	P	P	Legt fest, ob der Eingabewert während der Eingabe angezeigt wird oder ein * für jedes Zeichen.
HorizontalAlignment (Seite 1694)	P, A	P, A	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
InputValue (Seite 1700)	P	P	Definiert den im EA-Feld vom Benutzer eingegebenen Wert.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LeftMargin	-	A	Legt den Abstand des Texts zum linken Objektrand fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
LogOperation (Seite 1725)	P	P	Legt fest, ob nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.
LowerLimit (Seite 1727)	P	P	Legt den unteren Grenzwert für Eingabewerte fest.
Mode (Seite 1747)	P	P	Legt den Feldtyp des angegebenen Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
ProcessValue (Seite 1779)	P, A	P, A	Legt die Voreinstellung für den anzuzeigenden Wert fest.

Eigenschaften	Read	Write	Beschreibung
RightMargin	-	-	Legt den Abstand des Texts zum rechten Objektrand fest.
ShiftDecimalPoint (Seite 1816)	-	-	Legt fest, dass das Feld "Dezimalkomma verschieben" nur gelesen werden kann.
ShowBadTagState (Seite 1817)	P	P	Legt fest, ob das Objekt grau dargestellt wird, wenn die verbundene Variable einen schlechten QualityCode oder Variablenstatus besitzt.
ShowLeadingZeros	-	-	Legt fest, dass führende Nullstellen angezeigt werden.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
TopMargin	-	-	Legt den Abstand des Texts zum oberen Objektrand fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
Unit (Seite 1951)	P	P	Legt die Maßeinheit im Objekt "IOField" fest.
UpperLimit (Seite 1953)	P	P	Legt den oberen Grenzwert für Eingabewerte fest.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
UseTagLimitColors	P	P	Legt fest, ob bei einer Grenzwertverletzung die projizierten Farben verwendet werden.
VerticalAlignment (Seite 1971)	P, A	P, A	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

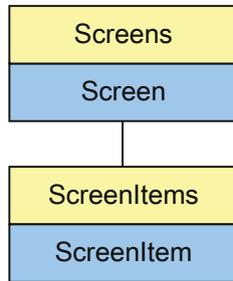
Tabelle 3-50 Methoden

Methoden	Güt.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

3.5.4.2 Objekte K-Z (Panels, Comfort Panels, RT Advanced, RT Professional)

Line (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Linie" dar. Das Line-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMILine

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-51 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderEndStyle (Seite 1592)	P, A	P, A	Legt die Art der Linienenden des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P, A	P, A	Legt die Art der Linienenden des angegebenen Objekts fest.

Eigenschaften	Read	Write	Beschreibung
Color (Seite 1618)	P, A	P, A	Legt die Linienfarbe des angegebenen Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndLeft	-	-	Legt die X-Position des Endpunkts fest.
EndStyle (Seite 1647)	P, A	P, A	Legt fest, wie das Linienende des angegebenen Objekts dargestellt wird.
EndTop	-	-	Legt die Y-Position des Endpunkts fest.
FillStyle	A	A	Legt das Füllmuster des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
LineWidth (Seite 1723)	P, A	P, A	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
RotationAngle (Seite 1786)	P	P	Legt den Rotationswinkel in Grad fest.
RotationCenterLeft (Seite 1786)	P	P	Legt die X-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
RotationCenterTop (Seite 1787)	P	P	Legt die Y-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
StartLeft	-	-	Legt die X-Position des Startpunkts fest.
StartStyle	A	A	Legt fest, wie der Linienanfang des angegebenen Objekts dargestellt wird.
StartTop (Seite 1835)	-	-	Legt den vertikalen Abstand des Startpunkts vom oberen Bildrand in Pixel fest.
Style (Seite 1848)	P, A	P, A	Legt den Linienstil des angegebenen Objekts fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.

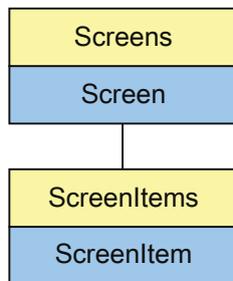
Eigenschaften	Read	Write	Beschreibung
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-52 Methoden

Methoden	Güt.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Listbox (RT Professional)

Beschreibung



Stellt das Objekt "Listenfeld" dar. Das Listbox-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMLListBox

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "ListBox1" um 10 Pixel nach rechts verschoben:

```
'VBS21
Dim objListBox
Set objListBox = ScreenItems("ListBox1")
objListBox.Left = objListBox.Left + 10
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-53 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AskOperationMotive (Seite 1565)	P	P	Legt fest, ob der Grund für die Bedienung dieses Objekts mitprotokolliert wird.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
CountVisibleItems (Seite 1630)	P	P	Legt fest, wie viele Zeilen die Auswahlliste enthält.
EdgeStyle (Seite 1642)	-	-	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.

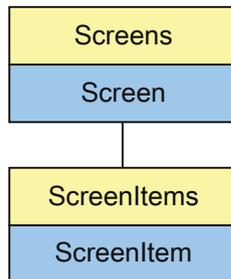
Eigenschaften	Read	Write	Beschreibung
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P	P	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalAlignment (Seite 1694)	P	P	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Index (Seite 1698)	P	P	Legt den Hintergrund der Raster-Steuer-elemente fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
LogOperation (Seite 1725)	P	P	Legt fest, ob nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.
Name (Seite 1749)	P	-	Gibt den Objekt-namen als STRING zurück.
SelectedIndex (Seite 1803)	P	P	Legt den Index fest und gibt ihn zurück, dessen zugehöriger Text im Kombinationsfeld oder Listefeld hervorgehoben angezeigt wird.
SelectedText (Seite 1801)	P	P	Zeigt den mit dem Attribut "Selektiertes Feld" (SelIndex) festgelegten Text, der im Kombinationsfeld oder Listefeld hervorgehoben dargestellt wird.
Text (Seite 1859)	P	P	Legt die Beschriftung für das Textfeld fest.
Texts	-	-	Legt die Texte für die Kontrollkästchen fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-54 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

MediaPlayer (RT Professional)

Beschreibung



Stellt das Objekt "Media Player" dar. Das MediaControl-Objekt ist Element der ScreenItems-Auflistung

Typkennzeichnung in VBS

HMIMediaControl

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "Control1" um 16 Pixel nach rechts verschoben:

```
'VBS60
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left + 16
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-55 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AspectRatio	P	P	Legt fest, ob bei einer Größenänderung das Seitenverhältnis erhalten bleiben soll.
AutoStart	P	P	Legt fest, dass der Media Player automatisch gestartet wird.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FileName	P, A	P, A	Legt den Namen der zu ladenden Datei fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
PictureSizeMode	P	P	Legt fest, dass die Statusleiste angezeigt wird.
PlayCount	A	A	Spielt die Datei mehrere Male ab.
PlayEndless	P	P	Legt fest, dass die Mediendatei endlos wiedergegeben wird.
PopupMenuEnabled	P	P	Legt fest, dass das Kontextmenü verwendbar ist.
ShowControls	P, A	P, A	Legt fest, dass eine Symbolleiste angezeigt wird.
ShowFeatureBackward	P	P	Legt fest, dass in Runtime die Taste "Rückwärts" angezeigt wird.
ShowFeatureForward	P	P	Legt fest, dass in Runtime die Taste "Vorwärts" angezeigt wird.
ShowFeatureFullScreen	P	P	Legt fest, dass der Media Player in den Vollbildmodus umgeschaltet werden kann.
ShowFeaturePause	P	P	Legt fest, dass der Media Player in den Vollbildmodus umgeschaltet werden kann.
ShowFeaturePlay	P	P	Legt fest, dass in Runtime die Taste "Wiedergabe" angezeigt wird.
ShowFeatureStop	P	P	Legt fest, dass in Runtime die Taste "Stopp" angezeigt wird.
ShowFeatureVolume	P	P	Legt die Lautstärke fest.

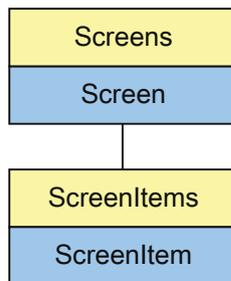
Eigenschaften	Read	Write	Beschreibung
ShowStatusBar (Seite 1826)	A	A	Legt fest, ob die Statusleiste angezeigt wird.
ShowTracker	A	A	Legt fest, dass die Zeitleiste angezeigt wird.
StepSeconds	P	P	Legt die Zeit in Sekunden fest, um die gesprungen, wenn die Taste "Vorwärts" oder "Rückwärts" betätigt wird.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-56 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

MultiLineEdit (RT Professional)

Beschreibung



Stellt das Objekt "Mehrzeiliger Text" dar. Das MultiLineEdit-Objekt ist Element der ScreenItems-Auflistung

Typkennzeichnung in VBS

HMIMultiLineEdit

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "MultiLineEdit1" um 10 Pixel nach rechts verschoben:

```
'VBS21
Dim objMultiLineEdit
Set objMultiLineEdit = ScreenItems("MultiLineEdit1")
objMultiLineEdit.Left = objMultiLineEdit.Left + 10
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-57 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
EdgeStyle (Seite 1642)	-	-	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P	P	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.

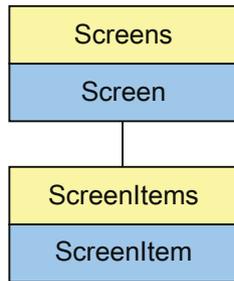
Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalAlignment (Seite 1694)	P	P	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
Text (Seite 1859)	P	P	Legt die Beschriftung für das Textfeld fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-58 Methoden

Methoden	Gültigkeit	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.
GetSelectionText	??	
SetSelection	??	
SetSelectionText	??	

OLEView (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "OLE-Objekt" dar. Das OLEView-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

Versionsunabhängige ProgID

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-59 Eigenschaften

Eigenschaften	Read	Write	Beschreibung

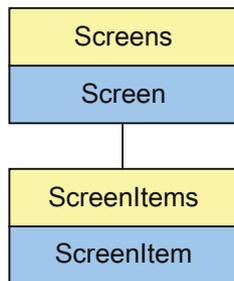
Eigenschaften	Read	Write	Beschreibung

Tabelle 3-60 Methoden

Methoden	Dynamisierbar	Gültigkeit	Beschreibung

OnlineTableControl (RT Professional)

Beschreibung



Stellt das Objekt "Tabellenanzeige" dar. Das OnlineTableControl-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIOnlineTableControl

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-61 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
ApplyProjectSettingsForDesignMode (Seite 1564)	-	-	Legt fest, dass für das Design die Projekteinstellungen verwendet werden.
AutoCompleteColumns (Seite 1568)	P	P	Legt fest, ob Leerspalten angezeigt werden, wenn das Control breiter ist als die projektierten Spalten.
AutoCompleteRows (Seite 1568)	P	P	Legt fest, ob Leerzeilen angezeigt werden, wenn das Control länger ist als die Anzahl der projektierten Zeilen.

Eigenschaften	Read	Write	Beschreibung
AutoSelectionColors (Seite 1569)	P	P	Legt fest, ob die Markierungsfarben für Zellen und Zeilen mit den vom System festgelegten Farben dargestellt werden.
AutoSelectionRectColor (Seite 1570)	P	P	Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
Caption (Seite 1608)	P	P	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CellCut (Seite 1611)	P	P	Legt fest, ob die Inhalte der Zellen abgekürzt werden, wenn die Zellen zu schmal sind.
CellSpaceBottom (Seite 1612)	P	P	Legt den Abstand von unten fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceLeft (Seite 1612)	P	P	Legt den Einzug Links fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceRight (Seite 1613)	P	P	Legt den Einzug Rechts fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceTop (Seite 1613)	P	P	Legt den Abstand von oben fest, der in den Zellen der Tabelle verwendet wird.
Closeable (Seite 1617)	P	P	Legt fest, dass Control in Runtime geschlossen werden kann.
ColumnResize (Seite 1621)	P	P	Legt fest, ob Sie die Breite der Spalten ändern können.
ColumnScrollbar (Seite 1621)	P	P	Legt den Typ der Bildlaufleiste fest.
ColumnTitleAlignment	P	P	Legt fest, wie die Spaltenüberschriften ausgerichtet werden.
ColumnTitles (Seite 1625)	P	P	Legt fest, ob die Spaltenüberschrift angezeigt wird.
ControlDesignMode (Seite 1831)	P	P	Legt das Design fest.
EnableEdit (Seite 1646)	P	P	Legt fest, ob Sie die angezeigten Daten in Runtime ändern können.
ExportDirectoryChangeable (Seite 1650)	P	P	Legt fest, ob das Verzeichnis für den Datenexport in Runtime geändert werden kann.
ExportDirectoryname (Seite 1651)	P	P	Legt das Verzeichnis fest, in das die Runtime-Daten exportiert werden.
ExportFileExtension (Seite 1651)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportFilename (Seite 1652)	P	P	Legt den Dateinamen der Datei fest, in welche die Runtime-Daten exportiert werden.
ExportFilenameChangeable (Seite 1652)	P	P	Legt fest, ob der Dateiname der Exportdatei in Runtime geändert werden kann.
ExportFormatGuid (Seite 1653)	P	P	Festgelegte Zuordnung von Ident-Nummer und Export-Provider.
ExportFormatName (Seite 1654)	P	P	Legt das Dateiformat für den Export fest. Zurzeit steht nur das "csv"-Dateiformat für den Export zur Verfügung.
ExportParameters (Seite 1654)	P	P	Legt die Parameter des ausgewählten Formats über den Dialog "Eigenschaften" fest.
ExportSelection (Seite 1655)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.

Eigenschaften	Read	Write	Beschreibung
ExportShowDialog (Seite 1655)	P	P	Legt fest, ob der Dialog zum Datenexport in Runtime angezeigt wird.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
GridLineColor (Seite 1678)	P	P	Legt die Farbe der Rasterlinien fest.
GridLineWidth (Seite 1679)	P	P	Legt die Stärke der Trennlinien in Pixel fest.
Height (Seite 1680)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalGridLines (Seite 1695)	P	P	Legt fest, ob horizontale Trennlinien angezeigt werden.
IconSpace (Seite 1697)	P	P	Legt den Abstand zwischen Symbol und Text in den Zellen der Tabelle fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineColor (Seite 1721)	P	P	Gibt die Farbe für die Fenstertrennlinien an.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
LoadDataImmediately (Seite 1724)	P	P	Legt fest, ob bei einem Bildaufschlag die Variablenwerte für den darzustellenden Zeitbereich aus den Archiven geladen werden.
Moveable (Seite 1748)	P	P	Legt fest, ob das Fenster in Runtime bewegt werden kann.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
Online (Seite 1757)	P	P	Legt den Namen der Datenquelle fest.
PrintJob	P	P	Legt einen Druckauftrag fest
RowScrollbar (Seite 1789)	P	P	Legt fest, ob Zeilen-Rollbalken angezeigt werden.
RowTitleAlignment	P	P	Legt fest, wie die Zeilenüberschriften ausgerichtet werden.
RowTitles	P	P	Legt fest, dass Zeilenköpfe angezeigt werden.
RTPersistence (Seite 1790)	P	P	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
RTPersistencePasswordLevel (Seite 2046)	P	P	Legt die Berechtigung fest, die in Runtime zur Online-Projektierung erforderlich ist.
RTPersistenceType (Seite 1790)	-	-	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
SelectedCellColor (Seite 1802)	P	P	Gibt die Hintergrundfarbe der markierten Zelle an.
SelectedCellForeColor (Seite 1803)	P	P	Gibt die Schriftfarbe der markierten Zelle an.
SelectedRowColor (Seite 1804)	P	P	Gibt die Hintergrundfarbe der markierten Zeile an.
SelectedRowForeColor (Seite 1805)	P	P	Gibt die Schriftfarbe der markierten Zeile an.
SelectedTitleColor (Seite 1805)	P	P	Gibt die Hintergrundfarbe der markierten Tabellenüberschrift an.
SelectedTitleForeColor (Seite 1806)	P	P	Gibt die Schriftfarbe der markierten Tabellenüberschrift an.
SelectionColoring (Seite 1808)	P	P	Legt fest, ob Markierungsfarben für Zelle oder Zeile verwendet werden.
SelectionRect (Seite 1810)	P	P	Legt fest, ob ein Auswahlrahmen für die markierten Zellen bzw. Zeilen verwendet wird.
SelectionRectColor (Seite 1810)	P	P	Legt die Farbe des Auswahlrechtecks im Meldefenster fest, wenn der SelectionType "1" ist.
SelectionRectWidth (Seite 1811)	P	P	Legt die Linienstärke des Auswahlrechtecks im Meldefenster fest, wenn der SelectionType "1" ist.

Eigenschaften	Read	Write	Beschreibung
SelectionType (Seite 1812)	P	P	Legt fest, wie viele Zeilen Sie markieren können.
ShowSortButton (Seite 1825)	P	P	Legt fest, ob die Sortiertaste über dem vertikalen Rollbalken angezeigt wird.
ShowSortIcon (Seite 1825)	P	P	Legt fest, ob das Sortiersymbol angezeigt wird.
ShowSortIndex (Seite 1826)	P	P	Legt fest, ob das Sortiersymbol angezeigt wird.
ShowTitle (Seite 1829)	P	P	Legt die Darstellung der Fensterüberschrift des Controls fest.
Sizeable (Seite 1831)	P	P	Legt fest, ob in Runtime die Größe des Objekts geändert werden kann.
SortSequence (Seite 1833)	P	P	Legt fest, wie die Sortierreihenfolge durch Mausclick verändert werden kann.
StatusbarBackColor (Seite 1836)	P	P	Gibt die Hintergrundfarbe für die Statuszeile an.
StatusbarElementAdd (Seite 1836) *	P	P	Legt ein neues, benutzerdefiniertes Element der Statuszeile an.
StatusbarElementAutoSize (Seite 1837) *	P	P	Legt fest, ob die Breite des ausgewählten Elements der Statuszeile automatisch eingestellt wird.
StatusbarElementCount (Seite 1838) *	P	P	Gibt die Anzahl der projektierbaren Elemente der Statuszeile an.
StatusbarElementIconId (Seite 1838) *	P	P	Festgelegte Zuordnung von Ident-Nummer und Symbol eines Elementes der Statuszeile.
StatusbarElementId (Seite 1839) *	P	P	Eindeutige Ident-Nummer für das gewählte Element der Statuszeile.
StatusbarElementIndex (Seite 1840)	P	P	Referenziert ein Element der Statuszeile. Unter Verwendung dieser Eigenschaft können Sie einem bestimmten Element der Statuszeile die Werte anderer Eigenschaften zuweisen.
StatusbarElementName (Seite 1840) *	P	P	Zeigt den Objektnamen für das gewählte Element der Statuszeile an.
StatusbarElementRemove (Seite 1841) *	P	P	Entfernt das ausgewählte Element der Statuszeile.
StatusbarElementRename (Seite 1842) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Statuszeile, das über die Eigenschaft "StatusbarElementIndex" referenziert wird.
StatusbarElementRepos *	P	P	Legt den Index fest, der dem Element zugeordnet werden soll.
StatusbarElements (Seite 1842)	-	-	Legt die in der Statuszeile darzustellenden Elemente fest.
StatusbarElementText *	P	P	Legt den Text des momentan gewählten Elements der Statuszeile fest.
StatusbarElementTooltipText (Seite 1842) *	P	P	Legt den Text für den Tooltip des benutzerdefinierten Elements der Statuszeile fest.
StatusbarElementUserDefined (Seite 1843) *	P	P	Zeigt an, ob das Element der Statuszeile vom Projektneur als neues, benutzerdefiniertes Element hinzugefügt wurde.
StatusbarElementVisible (Seite 1844) *	P	P	Aktivieren Sie in der Liste die Elemente der Statuszeile, die sie in Runtime anzeigen wollen.
StatusbarElementWidth (Seite 1844) *	P	P	Gibt die Breite des ausgewählten Elements der Statuszeile in Pixel an.
StatusbarFont	-	-	Statuszeile Schriftart
StatusbarFontColor (Seite 1845)	P	P	Gibt die Schriftfarbe der Texte in der Statuszeile an.

Eigenschaften	Read	Write	Beschreibung
StatusbarShowTooltips (Seite 1846)	P	P	Legt fest, ob in Runtime die Tooltips zu den Elementen der Statuszeile angezeigt werden.
StatusbarText (Seite 1846)	P	P	Festgelegter Text in der Statuszeile.
StatusbarUseBackColor (Seite 1847)	P	P	Legt fest, ob die Hintergrundfarbe der Statuszeile angezeigt wird.
StatusbarVisible (Seite 1848)	P	P	Legt fest, ob die Statuszeile des Controls angezeigt wird.
TableColor (Seite 1851)	P	P	Gibt die Hintergrundfarbe für die Zeilen an. Über die öffnen Sie den Farbauswahl-Dialog.
TableColor2 (Seite 1851)	P	P	Gibt die Hintergrundfarbe der "Zeilenfarbe 2" an.
TableForeColor (Seite 1852)	P	P	
TableForeColor2 (Seite 1853)	P	P	Gibt die Schrifffarbe der "Zeilenfarbe 2" an.
TimeBase (Seite 1866)	P	P	Legt die Zeitzone fest, auf deren Basis die Zeitwerte angezeigt werden.
TimeColumnActualize (Seite 1867) *	P	P	Legt fest, ob die Werte der ausgewählten Spalte aktualisiert werden.
TimeColumnAdd (Seite 1867) *	P	P	Legt eine neue Zeitspalte an.
TimeColumnAlign (Seite 1868) *	P	P	Legt fest, wie die ausgewählte Zeitspalte ausgerichtet wird.
TimeColumnBackColor (Seite 1869) *	P	P	Gibt die Hintergrundfarbe der ausgewählten Zeitspalte an.
TimeColumnBeginTime (Seite 1869) *	P	P	Legt den Anfangszeitpunkt des Zeitbereichs für die ausgewählte Zeitspalte fest.
TimeColumnCaption (Seite 1870) *	P	P	Legt die Bezeichnung der Zeitspalte fest.
TimeColumnCount (Seite 1870) *	P	P	Gibt die Anzahl der projizierten Zeitspalten an.
TimeColumnDateFormat (Seite 1871) *	P	P	Legt fest, welches Datumsformat zur Darstellung der ausgewählten Zeitspalte verwendet wird.
TimeColumnEndTime (Seite 1871) *	P	P	Legt den Endzeitpunkt des Zeitbereichs für die ausgewählte Zeitspalte fest.
TimeColumnForeColor (Seite 1872) *	P	P	Gibt die Schrifffarbe der ausgewählten Zeitspalte an.
TimeColumnHideText (Seite 1872) *	P	P	Legt fest, ob der Inhalt der Zeitspalte als Text angezeigt wird.
TimeColumnHideTitleText (Seite 1873) *	P	P	Legt fest, ob die Überschrift der Zeitspalte als Text angezeigt wird.
TimeColumnIndex (Seite 1873)	P	P	Referenziert eine projizierte Zeitspalte.
TimeColumnLength (Seite 1874) *	P	P	Legt die Breite für die ausgewählte Zeitspalte fest.
TimeColumnMeasurePoints (Seite 1874) *	P	P	Legt die Anzahl der Messpunkte fest, die in der ausgewählten Zeitspalte angezeigt werden.
TimeColumnName (Seite 1875) *	P	P	Legt den Namen der ausgewählten Zeitspalte fest.
TimeColumnRangeType (Seite 1875) *	P	P	Legt den Zeitbereich fest, der für die ausgewählte Zeitspalte verwendet wird.
TimeColumnRemove (Seite 1876) *	P	P	Entfernt die ausgewählte Zeitspalte aus der Liste.
TimeColumnRename (Seite 1876) *	P	P	Ändert den Namen der Zeitspalte, die über die Eigenschaft "TimeColumnIndex" referenziert wird.
TimeColumnRepos (Seite 1877) *	P	P	Ändert die Reihenfolge der Zeitspalten mit den zugehörigen Wertspalten.
TimeColumnShowDate (Seite 1877) *	P	P	Legt fest, ob die ausgewählte Zeitspalte mit Datum und Uhrzeit angezeigt wird.
TimeColumnShowIcon (Seite 1878) *	P	P	Legt fest, ob der Inhalt der Zeitspalte als Symbol angezeigt wird.

Eigenschaften	Read	Write	Beschreibung
TimeColumnShowTitleIcon (Seite 1878) *	P	P	Legt fest, ob die Werte der ausgewählten Spalte aktualisiert werden.
TimeColumnSort (Seite 1879) *	P	P	Legt fest, wie die im "TimeColumnIndex" referenzierte Zeitspalte sortiert wird.
TimeColumnSortIndex (Seite 1879) *	P	P	Gibt die Sortierreihenfolge der im "TimeColumnIndex" referenzierten Zeitspalte an.
TimeColumnTimeFormat (Seite 1880) *	P	P	Legt fest, welches Zeitformat zur Darstellung der ausgewählten Zeitspalte verwendet wird.
TimeColumnTimeRangeBase (Seite 1880) *	P	P	Legt die Zeiteinheit zur Bestimmung des Zeitbereichs fest.
TimeColumnTimeRangeFactor (Seite 1881) *	P	P	Legt den Faktor zur Bestimmung des Zeitbereichs fest.
TimeColumnUseValueColumnColors (Seite 1881) *	P	P	Legt fest, ob die ausgewählte Zeitspalte in den Farben der Wertspalte angezeigt wird.
TimeColumnVisible (Seite 1882) *	P	P	In der Liste werden die Zeitspalten aufgelistet, die Sie angelegt haben.
TimeStepBase	P	P	Ein Zeitbereich wird über eine Basis und einen Faktor festgelegt.
TimeStepFactor	P	P	Ein Zeitbereich wird über eine Basis und einen Faktor festgelegt.
TitleColor	P	P	Legt die Farbe der Titelzeile fest.
TitleCut (Seite 1884)	P	P	Legt fest, ob die Inhalte der Felder einer Titelleiste bei zu kleiner Spaltenbreite abgekürzt werden sollen.
TitleDarkShadowColor (Seite 1884)	P	P	Gibt die Farbe für die dunkle Seite der Schattierung an.
TitleForeColor (Seite 1885)	P	P	Gibt die Schriftfarbe der Tabellenüberschrift für den gewählten Zustand an.
TitleGridLineColor (Seite 1886)	P	P	Gibt die Farbe der Trennlinien in der Tabellenüberschrift an.
TitleLightShadowColor (Seite 1886)	P	P	Gibt die Farbe für die helle Seite der Schattierung an.
TitleSort (Seite 1887)	P	P	Legt fest, wie das Sortieren über Spaltenüberschrift ausgelöst wird.
TitleStyle (Seite 1888)	P	P	Legt fest, ob eine Schattierungsfarbe für die Tabellenüberschrift verwendet wird.
ToolBarAlignment (Seite 1895)	P	P	Legt die Position der Symbolleiste fest oder gibt sie zurück.
ToolBarBackColor (Seite 1895)	P	P	Gibt die Hintergrundfarbe der Symbolleiste an.
ToolBarButtonActive (Seite 1896) *	P	P	Legt fest, ob die mit der Taste verbundene Funktion in Runtime aktiv geschaltet ist.
ToolBarButtonAdd (Seite 1897) *	P	P	Legt eine neue, benutzerdefinierte Tastenfunktion an.
ToolBarButtonBeginGroup (Seite 1897) *	P	P	Legt fest, ob vor der ausgewählten Tastenfunktion ein Trennzeichen eingefügt wird.
ToolBarButtonClick	P	P	Klickt eine Schaltfläche der Symbolleiste.
ToolBarButtonCount (Seite 1898) *	P	P	Gibt die Anzahl der projektierbaren Tastenfunktionen an.
ToolBarButtonEnabled (Seite 1899) *	P	P	Legt fest, ob eine benutzerdefinierte Taste der Symbolleiste bedienbar ist.
ToolBarButtonHotKey (Seite 1899) *	P	P	Zeigt den Hotkey für die ausgewählte Tastenfunktion an.
ToolBarButtonId (Seite 1900) *	P	P	Eindeutige Ident-Nummer für die gewählte Tastenfunktion.
ToolBarButtonIndex (Seite 1900)	P	P	Referenziert eine Tastenfunktion.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
ToolBarButtonLocked (Seite 1901) *	P	P	Legt für eine benutzerdefinierte Taste der Symbolleiste fest, ob der eingerastete, gedrückte Zustand der Taste dargestellt wird.
ToolBarButtonName (Seite 1902) *	P	P	Zeigt den Namen für die ausgewählte Tastenfunktion an.
ToolBarButtonPasswordLevel (Seite 1902) *	P	P	Zeigt die Berechtigung für die ausgewählte Tastenfunktion an.
ToolBarButtonRemove (Seite 1903) *	P	P	Entfernt die ausgewählte Tastenfunktion aus der Liste.
ToolBarButtonRename (Seite 1903) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Symbolleiste, das über die Eigenschaft "ToolBarButtonIndex" referenziert wird.
ToolBarButtonRepos (Seite 1904) *	P	P	Ändert die Reihenfolge der Tastenfunktionen.
ToolBarButtons	-	-	Legt fest, welche Schaltflächen für die Symbolleiste verfügbar sind.
ToolBarButtonTooltipText (Seite 1905) *	P	P	Legt den Text für den Tooltip der Taste fest.
ToolBarButtonUserDefined (Seite 1905) *	P	P	Zeigt, ob die Taste der Symbolleiste vom Projektteur als neue, benutzerdefinierte Taste hinzugefügt wurde.
ToolBarButtonVisible *	P	P	Legt fest, dass die Schaltfläche sichtbar ist.
ToolBarShowTooltips (Seite 1906)	P	P	Legt fest, ob in Runtime die Tooltips zu den Tastenfunktionen angezeigt werden.
ToolBarUseBackColor (Seite 1906)	P	P	Legt fest, ob die Hintergrundfarbe der Symbolleiste angezeigt wird.
ToolBarUseHotKeys (Seite 1907)	P	P	Legt fest, ob die Hotkeys für die Tastenfunktionen in Runtime aktiviert sind.
ToolBarVisible (Seite 1908)	P	P	Legt fest, ob die Symbolleiste des Controls angezeigt wird.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
UseColumnBackColor	P	P	Legt fest, ob das Scrollen aktiviert ist.
UseColumnForeColor	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseSelectedTitleColor (Seite 1959)	P	P	Legt fest, ob eine Markierungsfarbe für die Tabellenüberschriften von markierten Tabellenzellen verwendet wird.
UseTableColor2 (Seite 1959)	P	P	Legt fest, ob eine zweite Zeilenfarbe bei der Darstellung der Tabelle verwendet wird.
ValueColumnAdd *	P	P	Fügt eine Wertespalte hinzu.
ValueColumnAlign *	P	P	Legt die Ausrichtung einer Spalte fest.
ValueColumnAutoPrecisions *	P	P	Legt fest, ob die Anzahl der anzuzeigenden Nachkommastellen automatisch ermittelt wird.
ValueColumnBackColor *	P	P	Legt die Hintergrundfarbe fest.
ValueColumnCaption *	P	P	Legt eine Überschrift fest.
ValueColumnCount *	P	P	Zeigt die Anzahl der Wertespalten an.
ValueColumnExponentialFormat *	P	P	Legt fest, dass in einer Spalte Exponentialschreibweise verwendet wird.
ValueColumnForeColor *	P	P	Legt die Vordergrundfarbe für eine Wertespalte fest.
ValueColumnHideText *	P	P	Legt fest, ob der Text angezeigt wird.
ValueColumnHideTitleText *	P	P	Legt fest, ob der Text angezeigt wird.
ValueColumnIndex	P	P	Der Index legt fest, auf welche Spalte sich andere Eigenschaften, wie z.B. Anfangswert, beziehen.
ValueColumnLength *	P	P	Legt die Anzahl der angezeigten Zeichen fest.

Eigenschaften	Read	Write	Beschreibung
ValueColumnName *	P	P	Legt den Namen einer Spalte fest.
ValueColumnPrecisions *	P	P	Legt die Anzahl an Nachkommastellen fest.
ValueColumnProvider *	P	P	
ValueColumnProviderCLSID *	P	P	Legt die Provider CLSID für die Daten einer Spalte fest.
ValueColumnRemove *	P	P	Löscht eine Wertespalte.
ValueColumnRename *	P	P	Benennt eine Wertespalte um.
ValueColumnRepos *	P	P	Legt fest, dass eine Wertespalte neu positioniert wird.
ValueColumnSelectTagName *	P	P	Legt eine Variable fest.
ValueColumnShowIcon *	P	P	Legt fest, ob ein Symbol angezeigt wird.
ValueColumnShowTitleIcon *	P	P	Legt fest, ob im Titel ein Symbol angezeigt wird.
ValueColumnSort *	P	P	Legt die Art der Sortierung in einer Spalte fest.
ValueColumnSortIndex *	P	P	Legt die Reihenfolge bei der Sortierung fest.
ValueColumnTagName *	P	P	Legt den Namen der Variablen fest, deren Werte in einer Spalte angezeigt werden.
ValueColumnTimeColumn *	P	P	Legt die zugehörige Zeitspalte fest.
ValueColumnVisible *	P	P	Legt die Sichtbarkeit einer Wertespalte fest.
VerticalGridLines (Seite 1972)	P	P	Legt fest, ob vertikale Trennlinien angezeigt werden.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

* in ES nicht sichtbar

Tabelle 3-62 Methoden

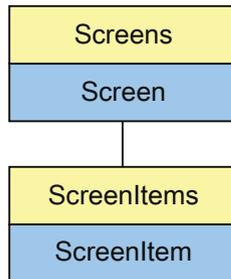
Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
AttachDB (Seite 2163)	P	Führt die Funktion der Schaltfläche "Backup verbinden" des Controls aus.
CalculateStatistic (Seite 2164)	P	Führt die Funktion der Schaltfläche "Statistik berechnen" der f(t)-Kurvenanzeige und der Tabellenanzeige aus.
CopyRows (Seite 2165)	P	Führt die Funktion der Schaltfläche "Zeilen kopieren" des Controls aus.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.
DetachDB (Seite 2169)	P, A	Führt die Funktion der Schaltfläche "Backup trennen" des Controls aus.
Edit (Seite 2170)	P	Führt die Funktion der Schaltfläche "Bearbeiten" der Tabellenanzeige aus.
Export (Seite 2170)	P, A	Führt die Funktion der Schaltfläche "Archiv exportieren" oder "Daten exportieren" des Controls aus.
GetRow (Seite 2182)	P	Gibt das mit der Zeilennummer bezeichnete Zeilen-Objekt der auf Tabellen basierenden Controls als Typ "ICCAxDataRow" zurück.
GetRowCollection (Seite 2183)	P	Gibt die Auflistung aller Zeilen-Objekte der auf Tabellen basierenden Controls als Typ "ICCAxDataRowCollection" zurück.
GetSelectedRow (Seite 2190)	P	Gibt das markierte Zeilen-Objekt eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Methoden	Gült.	Beschreibung
GetSelectedRows (Seite 2191)	P	Gibt bei Mehrfachselektion die markierten Zeilen-Objekte eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.
GetStatusBarElement (Seite 2197)	P, A	Gibt das mit Namen oder Index bezeichnete Element der Statuszeile des Controls als Typ "ICCAxStatusBarElement" zurück.
GetStatusBarElementCollection (Seite 2198)	P, A	Gibt die Auflistung aller Elemente der Statuszeile des Controls als Typ "ICCAxCollection" zurück.
GetTimeColumn (Seite 2203)	P	Gibt das mit Namen oder Index bezeichnete Zeitspalten-Objekt der Tabellenanzeige als Typ "ICCAxTimeColumn" zurück.
GetTimeColumnCollection (Seite 2204)	P	Gibt die Auflistung aller Zeitspalten-Objekte der Tabellenanzeige als Typ "ICCAxCollection" zurück.
GetToolBarButton (Seite 2205)	P, A	Gibt die mit Namen oder Index bezeichnete Tastenfunktion der Symbolleiste des Controls als Typ "ICCAxToolBarButton" zurück.
GetToolBarButtonCollection (Seite 2207)	P, A	Gibt die Auflistung aller Tastenfunktionen der Symbolleiste des Controls als Typ "ICCAxCollection" zurück.
GetValueColumn (Seite 2215)	P	Gibt das mit Namen oder Index bezeichnete Wertspalten-Objekt der f(t)-Kurvenanzeige als Typ "ICCAxValueColumn" zurück.
GetValueColumnCollection (Seite 2216)	P	Gibt die Auflistung aller Wertspalten-Objekte der f(t)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
MoveToFirst (Seite 2227)	P	Führt die Funktion der Schaltfläche "Erste Zeile" des Controls aus.
MoveToLast (Seite 2228)	P	Führt die Funktion der Schaltfläche "Letzter Datensatz" des Controls aus.
MoveToNext (Seite 2230)	P	Führt die Funktion der Schaltfläche "Nächster Datensatz" des Controls aus.
MoveToPrevious (Seite 2231)	P	Führt die Funktion der Schaltfläche "Vorhergehender Datensatz" des Controls aus.
NextColumn (Seite 2233)	P	Führt die Funktion der Schaltfläche "Nächste Spalte" der Tabellenanzeige aus.
PreviousColumn (Seite 2235)	P	Führt die Funktion der Schaltfläche "Vorhergehende Spalte" der Tabellenanzeige aus.
Print (Seite 2236)	P, A	Führt die Funktion der Schaltfläche "Drucken" des Controls aus.
SelectAll (Seite 2251)	P	Markiert alle Zeilen in einem tabellenbasierten Control.
SelectRow (Seite 2252)	P	Markiert eine bestimmte Zeile in einem tabellenbasierten Control.
SelectStatisticArea (Seite 2251)	P	Führt die Funktion der Schaltfläche "Statistikbereich festlegen" der Tabellenanzeige aus.
ShowColumnSelection (Seite 2253)	P	Führt die Funktion der Schaltfläche "Spalten wählen" der Tabellenanzeige aus.
ShowHelp (Seite 2255)	P, A	Führt die Funktion der Schaltfläche "Hilfe" des Controls aus.
ShowPropertyDialog (Seite 2260)	P, A	Führt die Funktion der Schaltfläche "Konfigurationsdialog" des Controls aus.
ShowTagSelection (Seite 2264)	P, A	Führt die Funktion der Schaltfläche "Datenanbindung wählen" des Controls aus.
ShowTimeSelection (Seite 2265)	P, A	Führt die Funktion der Schaltfläche "Zeitbereich wählen" des Controls aus.
StartStopUpdate (Seite 2266)	P, A	Führt die Funktion der Schaltfläche "Start" oder "Stopp" des Controls aus.
UnselectAll (Seite 2268)	P	Entfernt alle Markierungen in den Zellen eines tabellenbasierten Controls.
UnselectRow (Seite 2269)	P	Entfernt die Markierungen in einer bestimmten Zelle eines tabellenbasierten Controls.

OnlineTrendControl (RT Professional)

Beschreibung



Stellt das Objekt "f(t)-Kurvenanzeige" dar. Das OnlineTrendControl-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIOnlineTrendControl

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-63 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
ApplyProjectSettingsForDesignMode (Seite 1564)	-	-	Legt fest, dass für das Design die Projekteinstellungen verwendet werden.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
Caption (Seite 1608)	P	P	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.

Eigenschaften	Read	Write	Beschreibung
Closeable (Seite 1617)	P	P	Legt fest, dass Control in Runtime geschlossen werden kann.
ConnectTrendWindows (Seite 1626)	P	P	Legt fest, ob die projizierten Kurvenanzeigen verbunden werden.
ControlDesignMode (Seite 1831)	P	P	Legt das Design fest.
ExportDirectoryChangeable (Seite 1650)	P	P	Legt fest, ob das Verzeichnis für den Datenexport in Runtime geändert werden kann.
ExportDirectoryname (Seite 1651)	P	P	Legt das Verzeichnis fest, in das die Runtime-Daten exportiert werden.
ExportFileExtension (Seite 1651)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportFilename (Seite 1652)	P	P	Legt den Dateinamen der Datei fest, in welche die Runtime-Daten exportiert werden.
ExportFilenameChangeable (Seite 1652)	P	P	Legt fest, ob der Dateiname der Exportdatei in Runtime geändert werden kann.
ExportFormatGuid (Seite 1653)	P	P	Festgelegte Zuordnung von Ident-Nummer und Export-Provider.
ExportFormatName (Seite 1654)	P	P	Legt das Dateiformat für den Export fest. Zurzeit steht nur das "csv"-Dateiformat für den Export zur Verfügung.
ExportParameters (Seite 1654)	P	P	Legt die Parameter des ausgewählten Formats über den Dialog "Eigenschaften" fest.
ExportSelection (Seite 1655)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportShowDialog (Seite 1655)	P	P	Legt fest, ob der Dialog zum Datenexport in Runtime angezeigt wird.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
GraphDirection (Seite 1678)	P	P	Legt fest, an welchem Rand des Kurvenfensters die aktuellen Werte angezeigt werden.
Height (Seite 1680)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineColor (Seite 1721)	P	P	Gibt die Farbe für die Fenstertrennlinien an.
LineWidth	P	P	Legt die Linienstärke des angegebenen Objekts fest.
LoadDataImmediately (Seite 1724)	P	P	Legt fest, ob bei einem Bildaufschlag die Variablenwerte für den darzustellenden Zeitbereich aus den Archiven geladen werden.
Moveable (Seite 1748)	P	P	Legt fest, ob das Fenster in Runtime bewegt werden kann.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
Online (Seite 1757)	P	P	Legt den Namen der Datenquelle fest.
PercentageAxis	P	P	Legt fest, dass eine Achse mit prozentualer Skalierung angezeigt wird.
PercentageAxisAlign	P	P	
PercentageAxisColor	P	P	Legt die Farbe der prozentualen Achse fest.
PrintJob	P	P	Legt einen Druckauftrag fest
RTPersistence (Seite 1790)	P	P	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.

Eigenschaften	Read	Write	Beschreibung
RTPersistencePasswordLevel (Seite 2046)	P	P	Legt die Berechtigung fest, die in Runtime zur Online-Projektierung erforderlich ist.
RTPersistenceType (Seite 1790)	-	-	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
ShowRuler (Seite 1823)	P	P	Legt fest, ob für die Achsbeschriftung des Objekts "Online-TrendControl" eine Skaleneinteilung (Hilfslinie) eingeblendet wird.
ShowRulerInAxis (Seite 1823)	P	P	Legt fest, ob die Lineale auch in den Zeitachsen angezeigt werden.
ShowScrollbars (Seite 1824)	P	P	Legt den Typ der Bildlaufleiste fest.
ShowStatisticRuler	P	P	Legt das SQL-Statement fest.
ShowTitle (Seite 1829)	P	P	Legt die Darstellung der Fensterüberschrift des Controls fest.
ShowTrendIcon (Seite 1830)	P	P	Legt fest, ob unterhalb der Wertachsen ein Symbol eingeblendet wird.
Sizeable (Seite 1831)	P	P	Legt fest, ob in Runtime die Größe des Objekts geändert werden kann.
StatusbarBackColor (Seite 1836)	P	P	Gibt die Hintergrundfarbe für die Statuszeile an.
StatusbarElementAdd (Seite 1836) *	P	P	Legt ein neues, benutzerdefiniertes Element der Statuszeile an.
StatusbarElementAutoSize (Seite 1837) *	P	P	Legt fest, ob die Breite des ausgewählten Elements der Statuszeile automatisch eingestellt wird.
StatusbarElementCount (Seite 1838) *	P	P	Gibt die Anzahl der projektierbaren Elemente der Statuszeile an.
StatusbarElementIconId (Seite 1838) *	P	P	Festgelegte Zuordnung von Ident-Nummer und Symbol eines Elementes der Statuszeile.
StatusbarElementId (Seite 1839) *	P	P	Eindeutige Ident-Nummer für das gewählte Element der Statuszeile.
StatusbarElementIndex (Seite 1840)	P	P	Referenziert ein Element der Statuszeile. Unter Verwendung dieser Eigenschaft können Sie einem bestimmten Element der Statuszeile die Werte anderer Eigenschaften zuweisen.
StatusbarElementName (Seite 1840) *	P	P	Zeigt den Objektnamen für das gewählte Element der Statuszeile an.
StatusbarElementRemove (Seite 1841) *	P	P	Entfernt das ausgewählte Element der Statuszeile.
StatusbarElementRename (Seite 1842) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Statuszeile, das über die Eigenschaft "StatusbarElementIndex" referenziert wird.
StatusbarElementRepos *	P	P	Legt den Index fest, der dem Element zugeordnet werden soll.
StatusbarElements (Seite 1842)	-	-	Legt die in der Statuszeile darzustellenden Elemente fest.
StatusbarElementText *	P	P	Legt den Text des momentan gewählten Elements der Statuszeile fest.
StatusbarElementTooltipText (Seite 1842) *	P	P	Legt den Text für den Tooltip des benutzerdefinierten Elements der Statuszeile fest.
StatusbarElementUserDefined (Seite 1843) *	P	P	Zeigt an, ob das Element der Statuszeile vom Projektteur als neues, benutzerdefiniertes Element hinzugefügt wurde.
StatusbarElementVisible (Seite 1844) *	P	P	Aktivieren Sie in der Liste die Elemente der Statuszeile, die sie in Runtime anzeigen wollen.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
StatusbarElementWidth (Seite 1844) *	P	P	Gibt die Breite des ausgewählten Elements der Statuszeile in Pixel an.
StatusbarFont	-	-	Statuszeile Schriftart
StatusbarFontColor (Seite 1845)	P	P	Gibt die Schriftfarbe der Texte in der Statuszeile an.
StatusbarShowTooltips (Seite 1846)	P	P	Legt fest, ob in Runtime die Tooltips zu den Elementen der Statuszeile angezeigt werden.
StatusbarText (Seite 1846)	P	P	Festgelegter Text in der Statuszeile.
StatusbarUseBackColor (Seite 1847)	P	P	Legt fest, ob die Hintergrundfarbe der Statuszeile angezeigt wird.
StatusbarVisible (Seite 1848)	P	P	Legt fest, ob die Statuszeile des Controls angezeigt wird.
TimeAxes	-	-	Legt Einstellungen für Zeitachsen fest.
TimeAxisActualize *	P	P	
TimeAxisAdd *	P	P	Fügt eine Achse hinzu.
TimeAxisAlignment *	P	P	Legt die Ausrichtung einer Achse fest.
TimeAxisBeginTime (Seite 1864) *	P	P	Legt den Startzeitpunkt der Darstellung der angegebenen Kurve fest.
TimeAxisColor *	P	P	Legt die Farbe einer Achse fest.
TimeAxisCount *	P	P	Zeigt die Anzahl an Zeitachsen an.
TimeAxisDateFormat *	P	P	Legt das Format für Datumsangaben fest.
TimeAxisEndTime (Seite 1864) *	P	P	Legt den Endzeitpunkt der Darstellung der angegebenen Kurve fest.
TimeAxisIndex	P	P	Der Index legt fest, auf welche Achse sich andere Eigenschaften, wie z.B. Anfangswert, beziehen.
TimeAxisInTrendColor *	P	P	Legt fest, dass für eine Achse die Farbe der Kurve verwendet wird.
TimeAxisLabel (Seite 1865) *	P	P	Legt die Bezeichnung der Zeitachse fest. Ob die Angabe ausgewertet wird, ist abhängig von der Eigenschaft "ConfigureTimeAxis(i)".
TimeAxisMeasurePoints *	P	P	Legt die Messpunkte fest.
TimeAxisName *	P	P	Legt den Namen einer Achse fest.
TimeAxisRangeType *	P	P	Legt den Typ des Zeitbereichs fest.
TimeAxisRemove *	P	P	Löscht eine Zeitachse
TimeAxisRename *	P	P	Benennt eine Zeitachse um.
TimeAxisRepos *	P	P	Legt fest, dass die Zeitachse neu positioniert wird
TimeAxisShowDate *	P	P	Legt fest, dass das Datum angezeigt wird.
TimeAxisTimeFormat (Seite 1865) *	P	P	Legt das Format für Zeitangaben fest.
TimeAxisTimeRangeBase *	P	P	Ein Zeitbereich wird über eine Basis und einen Faktor festgelegt.
TimeAxisTimeRangeFactor *	P	P	Ein Zeitbereich wird über eine Basis und einen Faktor festgelegt.
TimeAxisTrendWindow *	P	P	Legt das Diagramm fest.
TimeAxisVisible *	P	P	Legt fest, dass die Zeitachse sichtbar ist.
TimeBase (Seite 1866)	P	P	Legt die Zeitzone fest, auf deren Basis die Zeitwerte angezeigt werden.
ToolbarAlignment (Seite 1895)	P	P	Legt die Position der Symbolleiste fest oder gibt sie zurück.

Eigenschaften	Read	Write	Beschreibung
ToolBarBackColor (Seite 1895)	P	P	Gibt die Hintergrundfarbe der Symbolleiste an.
ToolBarButtonActive (Seite 1896) *	P	P	Legt fest, ob die mit der Taste verbundene Funktion in Runtime aktiv geschaltet ist.
ToolBarButtonAdd (Seite 1897) *	P	P	Legt eine neue, benutzerdefinierte Tastenfunktion an.
ToolBarButtonBeginGroup (Seite 1897) *	P	P	Legt fest, ob vor der ausgewählten Tastenfunktion ein Trennzeichen eingefügt wird.
ToolBarButtonClick	P	P	Klickt eine Schaltfläche der Symbolleiste.
ToolBarButtonCount (Seite 1898) *	P	P	Gibt die Anzahl der projektierbaren Tastenfunktionen an.
ToolBarButtonEnabled (Seite 1899) *	P	P	Legt fest, ob eine benutzerdefinierte Taste der Symbolleiste bedienbar ist.
ToolBarButtonHotKey (Seite 1899) *	P	P	Zeigt den Hotkey für die ausgewählte Tastenfunktion an.
ToolBarButtonId (Seite 1900) *	P	P	Eindeutige Ident-Nummer für die gewählte Tastenfunktion.
ToolBarButtonIndex (Seite 1900)	P	P	Referenziert eine Tastenfunktion.
ToolBarButtonLocked (Seite 1901) *	P	P	Legt für eine benutzerdefinierte Taste der Symbolleiste fest, ob der eingerastete, gedrückte Zustand der Taste dargestellt wird.
ToolBarButtonName (Seite 1902) *	P	P	Zeigt den Namen für die ausgewählte Tastenfunktion an.
ToolBarButtonPasswordLevel (Seite 1902) *	P	P	Zeigt die Berechtigung für die ausgewählte Tastenfunktion an.
ToolBarButtonRemove (Seite 1903) *	P	P	Entfernt die ausgewählte Tastenfunktion aus der Liste.
ToolBarButtonRename (Seite 1903) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Symbolleiste, das über die Eigenschaft "ToolBarButtonIndex" referenziert wird.
ToolBarButtonRepos (Seite 1904) *	P	P	Ändert die Reihenfolge der Tastenfunktionen.
ToolBarButtons	-	-	Legt fest, welche Schaltflächen für die Symbolleiste verfügbar sind.
ToolBarButtonTooltipText (Seite 1905) *	P	P	Legt den Text für den Tooltip der Taste fest.
ToolBarButtonUserDefined (Seite 1905) *	P	P	Zeigt, ob die Taste der Symbolleiste vom Projekteur als neue, benutzerdefinierte Taste hinzugefügt wurde.
ToolBarButtonVisible *	P	P	Legt fest, dass die Schaltfläche sichtbar ist.
ToolBarShowTooltips (Seite 1906)	P	P	Legt fest, ob in Runtime die Tooltips zu den Tastenfunktionen angezeigt werden.
ToolBarUseBackColor (Seite 1906)	P	P	Legt fest, ob die Hintergrundfarbe der Symbolleiste angezeigt wird.
ToolBarUseHotKeys (Seite 1907)	P	P	Legt fest, ob die Hotkeys für die Tastenfunktionen in Runtime aktiviert sind.
ToolBarVisible (Seite 1908)	P	P	Legt fest, ob die Symbolleiste des Controls angezeigt wird.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
TrendAdd (Seite 1917) *	P	P	Legt eine neue Kurve an.
TrendAutoRangeBeginTagName *	P	P	Legt eine Variable fest, die den Anfangswert definiert.
TrendAutoRangeBeginValue *	P	P	Legt den Anfangswert für den automatisch angepassten Darstellungsbereich fest.
TrendAutoRangeEndTagName *	P	P	Legt eine Variable fest, die den Endwert definiert.
TrendAutoRangeEndValue *	P	P	Legt eine Variable fest, die den Endwert definiert.
TrendAutoRangeSource *	P	P	Legt die Quelle für den Darstellungsbereich fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
TrendColor *	P	P	Legt die Farbe der Trendanzeige fest oder gibt sie zurück.
TrendCount (Seite 1919) *	P	P	Gibt die Anzahl der projizierten Kurven an.
TrendExtendedColorSet (Seite 1920) *	P	P	Legt fest, ob Sie die Punktfarbe und die Füllfarbe projizieren können und ob die Farben in Runtime dargestellt werden.
TrendFill (Seite 1920) *	P	P	Legt fest, ob die Fläche unterhalb der Kurve gefüllt dargestellt wird.
TrendFillColor (Seite 1921) *	P	P	Gibt die Füllfarbe der Kurve an.
TrendIndex (Seite 1922)	P	P	Referenziert eine projizierte Kurve.
TrendLabel (Seite 1923) *	P	P	Legt die Bezeichnung der ausgewählten Kurve fest.
TrendLineStyle (Seite 1923) *	P	P	Legt fest, welche Linienart zur Darstellung der Kurve verwendet wird.
TrendLineType (Seite 1924) *	P	P	Legt fest, wie die Kurve dargestellt wird.
TrendLineWidth (Seite 1925) *	P	P	Legt die Linienstärke der dargestellten Linie fest.
TrendLowerLimit (Seite 1925) *	P	P	Gibt den unteren Grenzwert für eine Variable an.
TrendLowerLimitColor (Seite 1926) *	P	P	Legt die Farbe fest, die Variablenwerte kennzeichnet, die unterhalb des Werts von "TrendLowerLimit" liegen.
TrendLowerLimitColoring (Seite 1926) *	P	P	Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.
TrendName (Seite 1927) *	P	P	Zeigt den Namen der ausgewählten Kurve an.
TrendPointColor (Seite 1928) *	P	P	Gibt die Farbe der Punkte auf der Kurve an.
TrendPointStyle (Seite 1928) *	P	P	Legt fest, wie die Punkte auf der Kurve dargestellt werden.
TrendPointWidth (Seite 1929) *	P	P	Legt die Punktbreite in Pixel fest.
TrendProvider (Seite 1929) *	P	P	Legt die Datenversorgung der ausgewählten Kurve fest.
TrendProviderCLSID *	P	P	Legt die Provider CLSID für die Daten einer Kurve fest.
TrendRemove (Seite 1931) *	P	P	Entfernt ausgewählte Kurven aus der Liste.
TrendRename (Seite 1931) *	P	P	Ändert den Namen der Kurve, die über die Eigenschaft "TrendIndex" referenziert wird.
TrendRepos (Seite 1932) *	P	P	Ändert die Reihenfolge der ausgewählten Kurve im Kurvenfenster.
TrendSelectTagName *	P	P	Legt eine Variable fest.
TrendTagName *	P	P	Legt den Namen einer Variablen für die Y-Richtung fest.
TrendTimeAxis *	P	P	Legt die Zeitachse fest.
TrendTrendWindow (Seite 1935) *	P	P	Legt fest, in welchem Kurvenfenster die ausgewählte Kurve dargestellt wird.
TrendUncertainColor (Seite 1936) *	P	P	Werte haben einen unsicheren Status, wenn der Anfangswert nach dem Aktivieren von Runtime unbekannt ist oder ein Ersatzwert verwendet wird.
TrendUncertainColoring (Seite 1936) *	P	P	Werte haben einen unsicheren Status, wenn der Anfangswert nach dem Aktivieren von Runtime unbekannt ist oder ein Ersatzwert verwendet wird.
TrendUpperLimit (Seite 1937) *	P	P	Gibt den oberen Grenzwert für eine Variable an. Wenn die Variable den Wert von "TrendUpperLimit" unterschreitet, werden die Werte mit der in "TrendUpperLimitColor" eingestellten Farbe gekennzeichnet.
TrendUpperLimitColor (Seite 1937) *	P	P	Legt die Farbe fest, die Variablenwerte kennzeichnet, die unterhalb des Werts von "TrendLowerLimit" liegen.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
TrendUpperLimitColoring (Seite 1938) *	P	P	Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.
TrendValueAxis *	P	P	Legt die Werteachse fest.
TrendVisible (Seite 1939) *	P	P	In der Liste werden die Kurven aufgelistet, die Sie angelegt haben.
TrendWindowAdd (Seite 1939) *	P	P	Legt eine neue Kurvenanzeige an.
TrendWindowCoarseGrid (Seite 1940) *	P	P	Legt fest, ob die Gitternetzlinien für die Hauptskalierung angezeigt werden.
TrendWindowCoarseGridColor (Seite 1940) *	P	P	Gibt die Farbe der Gitternetzlinien für die Hauptskalierung an.
TrendWindowCount *	P	P	Zeigt die Anzahl der Diagramme an.
TrendWindowFineGrid (Seite 1941) *	P	P	Legt fest, ob die Gitternetzlinien für die Hilfsskalierung angezeigt werden.
TrendWindowFineGridColor (Seite 1941) *	P	P	Gibt die Farbe der Gitternetzlinien für die Hilfsskalierung an.
TrendWindowForegroundTrendGrid (Seite 1942) *	P	P	Legt fest, ob nur die Gitternetzlinien für die Vordergrundkurve im ausgewählten Kurvenfenster angezeigt werden.
TrendWindowGridInTrendColor (Seite 1942) *	P	P	Legt fest, ob die Gitternetzlinien für die Hauptskalierung in der Kurvenfarbe dargestellt werden.
TrendWindowHorizontalGrid (Seite 1943) *	P	P	Legt fest, ob die horizontalen Gitternetzlinien angezeigt werden.
TrendWindowIndex (Seite 1943)	P	P	Referenziert ein projektiertes Kurvenfenster.
TrendWindowName (Seite 1944) *	P	P	Legt den Namen des ausgewählten Kurvenfensters fest.
TrendWindowRemove (Seite 1944) *	P	P	Entfernt die ausgewählte Kurvenanzeige aus der Liste.
TrendWindowRename (Seite 1945) *	P	P	Ändert den Namen des Kurvenfensters, das über die Eigenschaft "TrendWindowIndex" referenziert wird.
TrendWindowRepos (Seite 1945) *	P	P	Ändert die Reihenfolge der Kurvenanzeige.
TrendWindowRulerColor (Seite 1946) *	P	P	Gibt die Farbe des Lineals an.
TrendWindowRulerLayer (Seite 1947) *	P	P	Legt die Darstellungsebene des Lineals in der Kurvenanzeige fest.
TrendWindowRulerStyle (Seite 1947) *	P	P	Legt die Darstellung des Lineals fest.
TrendWindowRulerWidth (Seite 1948) *	P	P	Legt die Stärke des Lineals in Pixel fest.
TrendWindows	-	-	Legt Einstellungen für Diagramme fest.
TrendWindowSpacePortion (Seite 1948) *	P	P	Legt den Anteil des ausgewählten Kurvenfensters bei der Darstellung im Control fest.
TrendWindowStatisticRulerColor *	P	P	Legt die Farbe des Statistiklineals fest.
TrendWindowStatisticRulerStyle *	P	P	Legt den Stil des Statistiklineals fest.
TrendWindowStatisticRulerWidth *	P	P	Legt die Breite des Statistiklineals fest.
TrendWindowVerticalGrid (Seite 1949) *	P	P	Legt fest, ob die vertikalen Gitternetzlinien angezeigt werden.
TrendWindowVisible (Seite 1949) *	P	P	In der Liste werden die Kurvenanzeigen aufgelistet, die Sie angelegt haben.
UseTrendNameAsLabel (Seite 1962)	P	P	Legt fest, ob die Eigenschaft "Name" oder "Beschriftung" oder als Bezeichnung der Kurve in Runtime verwendet wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
ValueAxes	-	-	Legt Einstellungen für Wertachsen fest.
ValueAxisAdd *	P	P	Fügt eine Achse hinzu.
ValueAxisAlignment *	P	P	Legt die Ausrichtung einer Achse fest.
ValueAxisAutoPrecisions *	P	P	Legt fest, dass die Anzahl der dargestellten Nachkommastellen einer Achse automatisch angepasst wird.
ValueAxisAutoRange *	P	P	Legt fest, dass der Wertebereich automatisch ermittelt wird.
ValueAxisBeginValue *	P	P	Legt den Anfangswert einer Achse fest.
ValueAxisColor *	P	P	Legt die Farbe einer Wertachse fest.
ValueAxisCount *	P	P	Zeigt die Anzahl der Wertachsen an.
ValueAxisEndValue *	P	P	Legt den Endwert einer Achse fest.
ValueAxisExponentialFormat *	P	P	Legt fest, dass die Beschriftung einer Achse Exponential-schreibweise verwendet.
ValueAxisIndex	P	P	Der Index legt fest, auf welche Achse sich andere Eigenschaften, wie z.B. Anfangswert, beziehen.
ValueAxisInTrendColor *	P	P	Legt fest, dass für eine Achse die Farbe der Kurve verwendet wird.
ValueAxisLabel (Seite 1970) *	P	P	Legt die Bezeichnung der Wertachse fest.
ValueAxisName *	P	P	Legt den Namen einer Achse fest.
ValueAxisPrecisions *	P	P	Legt die Anzahl der angezeigten Nachkommastellen fest.
ValueAxisRemove *	P	P	Benennt eine Achse um.
ValueAxisRename *	P	P	Benennt eine Achse um.
ValueAxisRepos *	P	P	Legt fest, dass die Wertachse neu positioniert wird.
ValueAxisScalingType (Seite 1970) *	P	P	Legt die Art der Skalierung der Achsen fest.
ValueAxisTrendWindow *	P	P	Legt das Diagramm fest.
ValueAxisVisible *	P	P	Legt fest, ob eine Wertachse sichtbar ist.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

* in ES nicht sichtbar

Tabelle 3-64 Methoden

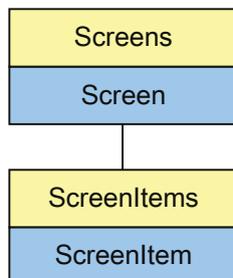
Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
AttachDB (Seite 2163)	P, A	Führt die Funktion der Schaltfläche "Backup verbinden" des Controls aus.
CalculateStatistic (Seite 2164)	P	Führt die Funktion der Schaltfläche "Statistik berechnen" der f(t)-Kurvenanzeige und der Tabellenanzeige aus.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.
DetachDB (Seite 2169)	P, A	Führt die Funktion der Schaltfläche "Backup trennen" des Controls aus.
Export (Seite 2170)	P, A	Führt die Funktion der Schaltfläche "Archiv exportieren" oder "Daten exportieren" des Controls aus.

Methoden	Gült.	Beschreibung
GetRulerData (Seite 2189)	P	Gibt den Wert der aufgerufenen Kurve an der Linealposition zurück.
GetStatusBarElement (Seite 2197)	P, A	Gibt das mit Namen oder Index bezeichnete Element der Statuszeile des Controls als Typ "ICCAxStatusBarElement" zurück.
GetStatusBarElementCollection (Seite 2204)	P, A	Gibt die Auflistung aller Elemente der Statuszeile des Controls als Typ "ICCAxCollection" zurück.
GetTimeAxis (Seite 2200)	P	Gibt das mit Namen oder Index bezeichnete Zeitachsen-Objekt der f(t)-Kurvenanzeige als Typ "ICCAxTimeAxis" zurück.
GetTimeAxisCollection (Seite 2201)	P	Gibt die Auflistung aller Zeitachsen-Objekte der f(t)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
GetToolBarButton (Seite 2205)	P	Gibt die mit Namen oder Index bezeichnete Tastenfunktion der Symbolleiste des Controls als Typ "ICCAxToolBarButton" zurück.
GetToolBarButtonCollection (Seite 2207)	P	Gibt die Auflistung aller Tastenfunktionen der Symbolleiste des Controls als Typ "ICCAxCollection" zurück.
GetTrend (Seite 2208)	P, A	Gibt die mit Namen oder Index bezeichnete Kurve der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxTrend" bzw. "ICCAxFunctionTrend" zurück.
GetTrendCollection (Seite 2209)	P, A	Gibt die Auflistung aller Kurven der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
GetTrendWindow (Seite 2210)	P, A	Gibt das mit Namen oder Index bezeichnete Kurvenfenster-Objekt der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxTrendWindow" zurück.
GetTrendWindowCollection (Seite 2211)	P, A	Gibt die Auflistung aller Kurvenfenster-Objekte der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
GetValueAxis (Seite 2212)	P	Gibt das mit Namen oder Index bezeichnete Wertachsen-Objekt der f(t)-Kurvenanzeige als Typ "ICCAxValueAxis" zurück.
GetValueAxisCollection (Seite 2214)	P	Gibt die Auflistung aller Wertachsen-Objekte der f(t)-Kurvenanzeige als Typ "ICCAxCollection" zurück.
MoveAxis (Seite 2226)	P, A	Führt die Funktion der Schaltfläche "Achsenbereich verschieben" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
MoveToFirst (Seite 2227)	P	Führt die Funktion der Schaltfläche "Erste Zeile" des Controls aus.
MoveToLast (Seite 2228)	P	Führt die Funktion der Schaltfläche "Letzter Datensatz" des Controls aus.
MoveToNext (Seite 2230)	P	Führt die Funktion der Schaltfläche "Nächster Datensatz" des Controls aus.
MoveToPrevious (Seite 2231)	P	Führt die Funktion der Schaltfläche "Vorhergehender Datensatz" des Controls aus.
NextTrend (Seite 2233)	P, A	Führt die Funktion der Schaltfläche "Nächste Kurve" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
OneToOneView (Seite 2234)	P, A	Führt die Funktion der Schaltfläche "Originalansicht" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
PreviousTrend (Seite 2235)	P, A	Führt die Funktion der Schaltfläche "Vorhergehende Kurve" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
Print (Seite 2236)	P, A	Führt die Funktion der Schaltfläche "Drucken" des Controls aus.
ShowHelp (Seite 2255)	P, A	Führt die Funktion der Schaltfläche "Hilfe" des Controls aus.
ShowPercentageAxis (Seite 2259)	P	Führt die Funktion der Schaltfläche "Relative Achse" der f(t)-Kurvenanzeige aus.
ShowPropertyDialog (Seite 2260)	P, A	Führt die Funktion der Schaltfläche "Konfigurationsdialog" des Controls aus.
ShowTagSelection (Seite 2264)	P, A	Führt die Funktion der Schaltfläche "Datenanbindung wählen" des Controls aus.

Methoden	Gült.	Beschreibung
ShowTimeSelection (Seite 2265)	P, A	Führt die Funktion der Schaltfläche "Zeitbereich wählen" des Controls aus.
ShowTrendSelection (Seite 2265)	P, A	Führt die Funktion der Schaltfläche "Kurven wählen" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
StartStopUpdate (Seite 2266)	P, A	Führt die Funktion der Schaltfläche "Start" oder "Stopp" des Controls aus.
ZoomArea (Seite 2274)	P, A	Führt die Funktion der Schaltfläche "Zoomen Ausschnitt" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
ZoomInOut (Seite 2274)	P, A	Führt die Funktion der Schaltfläche "Zoomen +/-" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.
ZoomInOutTime (Seite 2275)	P	Führt die Funktion der Schaltfläche "Zeitachse zoomen +/-" der f(t)-Kurvenanzeige aus.
ZoomInOutValues (Seite 2275)	P	Führt die Funktion der Schaltfläche "Wertachse zoomen +/-" der f(t)-Kurvenanzeige aus.
ZoomMove (Seite 2277)	P, A	Führt die Funktion der Schaltfläche "Kurvenbereich verschieben" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

OptionGroup (RT Professional)

Beschreibung



Stellt das Objekt "Optionsschaltflächen" dar. Das OptionGroup-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIOptionGroup

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-65 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AdaptBorder (Seite 1553)	P	P	Legt fest, ob sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CheckmarkAlignment (Seite 1616)	P	P	Legt fest, ob die Felder rechtsbündig angeordnet sind.
CheckmarkCount (Seite 1616)	P	P	Legt die Anzahl der Felder fest.
CornerStyle (Seite 1628)	-	-	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
EdgeStyle (Seite 1642)	-	-	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P	P	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalAlignment (Seite 1694)	P	P	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Index (Seite 1698)	P	P	Legt den Hintergrund der Raster-Steuerelemente fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	-	-	Legt die Form der Linienenden fest.
LogOperation (Seite 1725)	P	P	Legt fest, ob nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
ProcessValue (Seite 1779)	P	P	Legt die Voreinstellung für den anzuzeigenden Wert fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowBadTagState (Seite 1817)	P	P	Legt fest, ob das Objekt grau dargestellt wird, wenn die verbundene Variable einen schlechten QualityCode oder Variablenstatus besitzt.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
Text (Seite 1859)	P	P	Legt die Beschriftung für das Textfeld fest.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
Texts	-	-	Legt die Texte für die Kontrollkästchen fest.

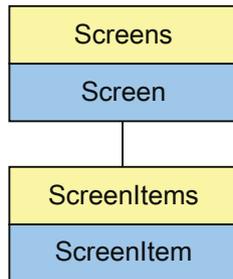
Eigenschaften	Read	Write	Beschreibung
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
VerticalAlignment (Seite 1971)	P	P	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-66 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Polygon (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Polygon" dar. Das Polygon-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIPolygon

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-67 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
ActualPointIndex (Seite 1551)	P	P	Legt die Nummer des aktuellen Eckpunkts des angegebenen Objekts fest.
ActualPointLeft (Seite 1551)	P	P	Legt die X-Koordinate des aktuellen Eckpunktes in Bezug auf den Bildursprung fest.
ActualPointTop (Seite 1552)	P	P	Legt die Y-Koordinate des aktuellen Eckpunkts in Bezug auf den Bildursprung fest.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.

Eigenschaften	Read	Write	Beschreibung
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P, A	P, A	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P, A	P, A	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P, A	P, A	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
EdgeStyle (Seite 1642)	A	A	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
Points	-	-	Legt die Eckpunkte fest.
PointsCount (Seite 1777)	P	-	Legt die Anzahl der Eckpunkte der Polylinie oder des Vielecks fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
RotationAngle (Seite 1786)	P	P	Legt den Rotationswinkel in Grad fest.
RotationCenterLeft (Seite 1786)	P	P	Legt die X-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
RotationCenterTop (Seite 1787)	P	P	Legt die Y-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

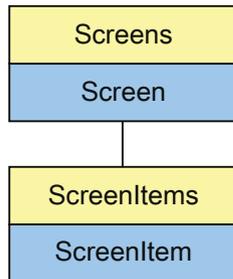
Eigenschaften	Read	Write	Beschreibung
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-68 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Polyline (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Polygonzug" dar. Das Polyline-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIPolyline

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-69 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
ActualPointIndex (Seite 1551)	P	P	Legt die Nummer des aktuellen Eckpunkts des angegebenen Objekts fest.
ActualPointLeft (Seite 1551)	P	P	Legt die X-Koordinate des aktuellen Eckpunktes in Bezug auf den Bildursprung fest.
ActualPointTop (Seite 1552)	P	P	Legt die Y-Koordinate des aktuellen Eckpunkts in Bezug auf den Bildursprung fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BorderEndStyle (Seite 1592)	P; A	P	Legt die Art der Linienenden des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P, A	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Color (Seite 1618)	P, A	P, A	Legt die Linienfarbe des angegebenen Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndStyle (Seite 1647)	A	A	Legt fest, wie das Linienende des angegebenen Objekts dargestellt wird.
FillStyle	A	A	Legt das Füllmuster des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
LineWidth (Seite 1723)	P, A	P, A	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
Points	-	-	Legt die Eckpunkte fest.
PointsCount (Seite 1777)	P	-	Legt die Anzahl der Eckpunkte der Polylinie oder des Vielecks fest.
RotationAngle (Seite 1786)	P	P	Legt den Rotationswinkel in Grad fest.
RotationCenterLeft (Seite 1786)	P	P	Legt die X-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
RotationCenterTop (Seite 1787)	P	P	Legt die Y-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
StartStyle	A	A	Legt fest, wie der Linienanfang des angegebenen Objekts dargestellt wird.
Style (Seite 1848)	A	A	Legt den Linienstil des angegebenen Objekts fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.

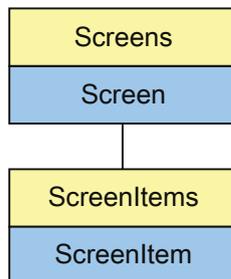
Eigenschaften	Read	Write	Beschreibung
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-70 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

ProjectName (RT Professional)

Beschreibung



Stellt das Objekt "Projektname" dar. Das ProjectName-Objekt ist Element der ScreenItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-71 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
BackColor	-	-	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle	-	-	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	-	-	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor	-	-	Legt die Linienfarbe des Objekts fest.
BorderWidth	-	-	Legt die Linienstärke des Objekts fest.
EdgeStyle (Seite 1642)	-	-	Legt den Linienstil des angegebenen Objekts fest.
FillPatternColor (Seite 1658)	-	-	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	-	-	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	-	-	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	-	-	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	-	-	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	-	-	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor	-	-	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Format	-	-	Legt fest, dass das Format des Textblocks angezeigt wird.
Height (Seite 1680)	-	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalAlignment	-	-	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	-	-	Legt den Wert der X-Koordinate des Objekts fest.
LineWrap	-	-	Legt den Zeilenumbruch innerhalb des Objekts fest.
Name (Seite 1749)	-	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	-	-	Legt den Wert der Y-Koordinate des Objekts fest.
VerticalAlignment	-	-	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	-	-	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	-	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

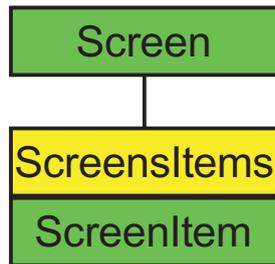
Keine der Eigenschaften im ES sichtbar

Tabelle 3-72 Methoden

Methoden	Gültigkeit	Beschreibung

ProtectedAreaName (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Wirkbereichs-Name (RFID)" dar. Das ProtectedAreaName-Objekt ist Element der ScreenItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-73 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.

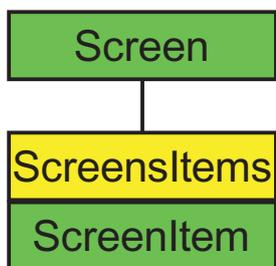
Eigenschaften	Read	Write	Beschreibung
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-74 Methoden

Methoden	Gült.	Beschreibung

RangeLabelView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Wirkbereichs-Name" dar. Das RangeLabelView-Objekt ist Element der ScreenItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-75 Eigenschaften

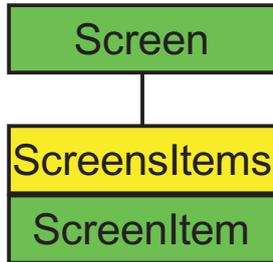
Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-76 Methoden

Methoden	Gült.	Beschreibung

RangeQualityView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Wirkbereichs-Signal" dar. Das RangeQualityView-Objekt ist Element der ScreensItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-77 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

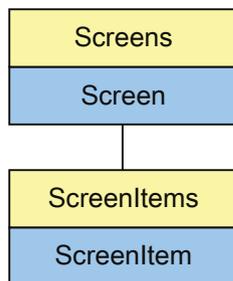
Tabelle 3-78 Methoden

Methoden	Gült.	Beschreibung

Methoden	Gült.	Beschreibung

RecipeView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Rezeptanzeige" dar. Das RecipeView-Objekt ist Element der ScreenItems-Auflistung.

Wenn Sie die Einstellungen für dieses Objekt mit einer benutzerdefinierten Funktion ändern, bleiben die geänderten Einstellungen auch nach einem Neuaufruf des Bildes mit diesem Objekt erhalten.

Hinweis

Das Objekt "Einfache Rezeptanzeige" kann nicht mit einer benutzerdefinierten Funktion dynamisiert werden.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-79 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AllowEdit	-	-	Legt fest, dass das Bearbeiten möglich ist.
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackButtonVisible	-	-	Legt fest, dass die Schaltfläche "Zurück" angezeigt wird.
BackColor	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
ComboBoxFont	-	-	Legt die Schriftart für die Auswahlliste fest.
DataRecordNameCaption	-	-	Legt die Beschriftung für den Namen des Datensatzes fest.
DataRecordNrCaption	-	-	Legt die Beschriftung für die Nummer des Datensatzes fest.
Display3D	-	-	Legt die 3D-Ansicht fest.
DisplayButton2Plc	-	-	Legt fest, dass die Schaltfläche "Schreiben in Steuerung" angezeigt wird.
DisplayButtonComparison	-	-	Legt fest, dass die Schaltfläche "Synchronisation" angezeigt wird.
DisplayButtonDelete	-	-	Legt fest, dass die Schaltfläche "Datensatz löschen" angezeigt wird.
DisplayButtonFromPlc	-	-	Legt fest, dass die Schaltfläche "Lesen aus Steuerung" angezeigt wird.
DisplayButtonHelp	-	-	Legt fest, dass die Schaltfläche "Tooltip" angezeigt wird.
DisplayButtonNew	-	-	Legt fest, dass die Schaltfläche "Datensatz hinzufügen" angezeigt wird.
DisplayButtonSave	-	-	Legt fest, dass die Schaltfläche "Speichern" angezeigt wird.
DisplayButtonSaveAs	-	-	Legt fest, dass die Schaltfläche "Speichern unter" angezeigt wird.
DisplayComboBox	-	-	Legt fest, dass die Auswahlliste angezeigt wird.
DisplayGridLines	-	-	Legt fest, dass die Rasterlinien angezeigt werden.
DisplayLabeling	-	-	Legt fest, dass die Beschriftungen angezeigt werden.
DisplayNumbers	-	-	Legt fest, dass die Nummern angezeigt werden.
DisplaySize *	-	-	
DisplayStatusBar	-	-	Legt fest, dass die Statusleiste angezeigt wird.
DisplayTable	-	-	Legt fest, dass die Tabelle angezeigt wird.
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EntryNameCaption	-	-	Legt die Beschriftung für den Namen des Rezepturelements fest.
EntryValueColFirst	-	-	Legt fest, dass die Spalte "Eintragswert" die Erste ist.
EntryValueFieldLength	-	-	Legt die Feldlänge für den Elementwert fest.
EntryValuePos	-	-	Legt die Position des ersten Wertes fest.
FitToSize	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.

Eigenschaften	Read	Write	Beschreibung
FixedRecipeNumber *	-	-	
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FocusColor	A	A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth	A	A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
ForeColor	A	A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
HeaderFont	-	-	Legt die Schriftart der Überschrift fest.
Height (Seite 1680)	A	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
KeyboardOnline	-	-	Legt fest, dass die Tastenkombinationen verwendet werden können.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
ListAreaHeight	-	-	
ListAreaWidth	-	-	
MenuButtonVisible	-	-	Legt fest, dass die Schaltflächen des Menüs sichtbar sind.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
NameColumnWidth	-	-	
NumberOfLines	-	-	Legt für das Objekt Textliste fest, wie viele Zeilen die Auswahlliste enthalten soll oder gibt den Wert zurück.
Recipe	-	-	Legt die Rezeptur fest.
RecipeNameCaption	-	-	Legt die Beschriftung für den Namen der Rezeptur fest.
RecipeNameSelected *	-	-	
RecipeNrCaption	-	-	Legt fest, dass die Spalte "Rezeptnummer" die erste ist.
RecipeNrColFirst	-	-	Legt fest, dass die Spalte "Rezeptnummer" die erste ist.
RecipeVarSelected *	-	-	
Record *	-	-	
RecordNameSelected *	-	-	
RecordNrColFirst	-	-	Legt fest, dass die Spalte "Datensatznummer" die Erste ist.
RecordVarSelected *	-	-	
RenameButtonVisible	-	-	Legt fest, dass die Schaltfläche "Umbenennen" angezeigt wird.
SelectionBackColor (Seite 1808)	A	A	Legt die Hintergrundfarbe der selektierten Zellen fest.
SelectionForeColor (Seite 1809)	A	A	Legt die Vordergrundfarbe der selektierten Zellen fest.
StatusFont	-	-	
TableBackColor (Seite 1850)	A	A	Legt die Hintergrundfarbe der Tabellenzellen des angegebenen Objekts fest.
TableForeColor	A	A	
TableGridlineColor (Seite 1854)	A	A	Legt die Farbe der Rasterlinien fest.
TableHeaderBackColor (Seite 1854)	A	A	Legt die Hintergrundfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
TableHeaderForeColor (Seite 1855)	A	A	Legt die Textfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
ValueCaption	-	-	Legt die Beschriftung für die Werte fest.
ValueColumnWidth	-	-	
VerticalScrolling	-	-	Legt fest, dass das Objekt eine vertikale Bildlaufleiste besitzt.
ViewType	-	-	Legt den Typ der Meldeanzeige fest.
ViewTypeForSaveStream *	-	-	Legt den Anzeigetyp für den Save Stream fest.
Visible (Seite 1974)	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
VisibleItems	-	-	Legt die Anzahl der sichtbaren Einträge fest.
Width (Seite 1983)	A	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

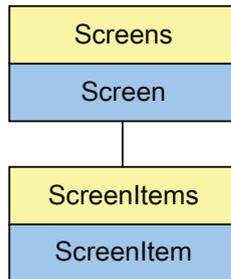
* im ES nicht sichtbar

Tabelle 3-80 Methoden

Methoden	Gült.	Beschreibung

Rectangle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Rechteck" dar. Das Rectangle-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIRectangle

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-81 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P, A	P, A	Legt das Füllmuster des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P, A	P, A	Legt die Linienfarbe des Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P, A	P, A	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	A	A	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
RoundCornerHeight (Seite 1788)	P	P	Legt den Eckradius fest oder gibt ihn zurück.
RoundCornerWidth (Seite 1788)	P	P	Legt den Eckradius fest oder gibt ihn zurück.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.

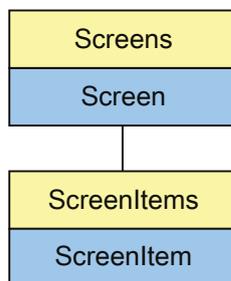
Eigenschaften	Read	Write	Beschreibung
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-82 Methoden

Methoden	Gültigkeit	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

RoundButton (RT Professional)

Beschreibung



Stellt das Objekt "Roundbutton" dar. Das RoundButton-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIRoundButton

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-83 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderBrightColor3D (Seite 1589)	P	P	Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderShadeColor3D (Seite 1599)	P	P	Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
BorderWidth3D (Seite 1603)	P	P	Legt die Breite des Rahmens für die 3D-Darstellung des angegebenen Objekts fest.
CornerStyle (Seite 1628) *	-	-	Legt die Art der Rahmenlinien des angegebenen Objekts fest.

Eigenschaften	Read	Write	Beschreibung
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	P	P	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P	P	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalAlignment (Seite 1694)	P	P	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722) *	-	-	Legt die Form der Linienenden fest.
Mode (Seite 1747)	P	P	Legt den Feldtyp des angegebenen Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
PictureAlignment (Seite 1774)	P	P	Legt die Darstellungsart des Hintergrundbildes im Prozessbild fest oder gibt sie zurück.
PictureDeactivated (Seite 1775)	P	P	Legt das Bild fest, das im Zustand "Deaktiviert" angezeigt wird.
PictureOff (Seite 1775)	P	P	Legt das Bild fest, das im Zustand "Aus" angezeigt wird.
PictureOn (Seite 1776)	P	P	Legt das Bild fest, das im Zustand "Ein" angezeigt wird.
Pressed (Seite 1778)	P	P	Legt fest, ob das angegebene Objekt gedrückt dargestellt ist.
Radius	P	P	Legt den Radius des angegebenen Objekts "Circle" fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
StyleSettings (Seite 1849)	P	P	Legt fest, in welchem Stil das Objekt dargestellt wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
Text (Seite 1859)	P	P	Legt die Beschriftung für das Textfeld fest.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
Toggle (Seite 1888)	P	P	Legt fest, ob das angegebene Objekt in Runtime nach dem Betätigen einrastet.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
TransparentColorDeactivatedPicture (Seite 1915)	P	P	Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Deaktiviert" auf "transparent" gesetzt wird.
TransparentColorPictureOff (Seite 1915)	P	P	Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Aus" auf "transparent" gesetzt wird.
TransparentColorPictureOn (Seite 1916)	P	P	Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Ein" auf "transparent" gesetzt wird.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
UseTransparentColorDeactivatedPicture (Seite 1961)	P	P	Legt fest, ob die mit der Eigenschaft "TransparentColorDeactivatedPicture" festgelegte Transparentfarbe für den Zustand "Deaktiviert" verwendet wird.
UseTransparentColorPictureOff (Seite 1961)	P	P	Legt fest, ob die mit der Eigenschaft "TransparentColorPictureOff" festgelegte Transparentfarbe für den Zustand "Aus" verwendet wird.
UseTransparentColorPictureOn (Seite 1962)	P	P	Legt fest, ob die mit der Eigenschaft "TransparentColorPictureOn" festgelegte Transparentfarbe für den Zustand "Ein" verwendet wird.
VerticalAlignment (Seite 1971)	P	P	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

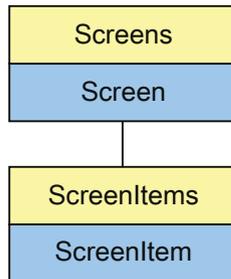
* im ES nicht sichtbar

Tabelle 3-84 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

ScreenWindow (RT Professional)

Beschreibung



Stellt das Objekt "Bildfenster" dar. Das ScreenWindow-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIScreenWindow

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-85 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AdaptScreenToWindow (Seite 1553)	-	-	Legt fest, ob das in einem Bildfenster dargestellte Bild sich in Runtime an die Größe des Bildfensters anpasst oder nicht.
AdaptWindowToScreen (Seite 1554)	-	-	Legt fest, ob das Bildfenster sich dem in ihm dargestellten Bild in Runtime anpasst oder nicht.
BorderEnabled (Seite 1591)	-	-	TRUE, wenn das Fenster in Runtime mit Rahmen dargestellt wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
CaptionText (Seite 1610)	P	P	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
Flashing *	-	-	Legt fest, dass der Textblock blinkt.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalScrollBarPosition (Seite 1696)	P	P	Legt die horizontale Verschiebung der Bildlaufleiste in einem Bildfenster mit Rollbalken fest oder gibt den Wert zurück.
IndependentWindow	-	-	Legt fest, ob das Bildfenster im Bild eingebettet ist oder als unabhängiges Bildfenster angezeigt wird.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LeftOffset (Seite 1711)	P	P	Legt die horizontale Verschiebung der Anzeige einer Grafik fest, die größer ist als das Bildfenster.
MenuToolBarConfig (Seite 1729)	-	-	Lädt die angegebene Konfigurationsdatei mit benutzerdefiniertem Menü und Symbolleisten oder gibt den Namen der Konfigurationsdatei zurück.
MonitorNumber	-	-	Legt die Monitornummer fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
ScreenName (Seite 1799)	P	P	Legt das Bild fest, das im Bildfenster in Runtime angezeigt wird oder gibt den Bildnamen zurück.
ShowCaption (Seite 1818)	-	-	Legt fest, ob die Titelzeile ein- oder ausgeblendet wird.
ShowScrollBars (Seite 1824)	-	-	Legt fest, ob die Rollbalken angezeigt werden.
TagPrefix (Seite 1856)	P	P	Legt das Präfix fest, das allen Variablen, die im Bild vorhanden sind, vorangestellt wird.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
TopOffset (Seite 1912)	P	P	Legt den Abstand des Bildes vom oberen Rand des Bildfensters fest oder gibt ihn zurück.
VerticalScrollBarPosition (Seite 1973)	P	P	Legt die vertikale Verschiebung der Bildlaufleiste in einem Bildfenster mit Rollbalken fest oder gibt den Wert zurück.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.
WindowCloseEnabled (Seite 1986)	-	-	Legt fest, ob ein Fenster in Runtime geschlossen werden kann.
WindowMaximizeEnabled (Seite 1986)	-	-	TRUE, wenn das Objekt in Runtime maximiert werden kann.
WindowMovingEnabled (Seite 1987)	-	-	TRUE, wenn das Objekt in Runtime verschoben werden kann.
WindowOnTop (Seite 1987)	-	-	TRUE, wenn das Objekt in Runtime immer im Vordergrund bleibt.
WindowSizingEnabled (Seite 1988)	-	-	Legt fest, ob in Runtime die Größe des angegebenen Objekts verändert werden kann.
WindowStartupPosition	-	-	??
ZoomFactor (Seite 2008)	P	P	Setzt den Zoomfaktor eines Bildes bzw. Bildfensters oder liest ihn aus.

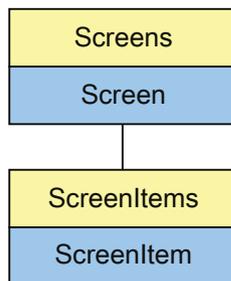
* im ES nicht sichtbar

Tabelle 3-86 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

ScriptDiagnostics (RT Professional)

Beschreibung



Stellt das Objekt "Applikationsfenster" dar. Das ApplicationWindow-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIApplicationWindow

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-87 Eigenschaften

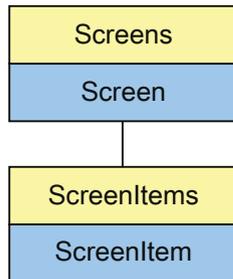
Eigenschaften	Read	Write	Beschreibung
BorderEnabled (Seite 1591)	-	-	TRUE, wenn das Fenster in Runtime mit Rahmen dargestellt wird.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	P	P	Gibt den Objektnamen als STRING zurück.
ShowCaption (Seite 1818)	-	-	Legt fest, ob die Titelzeile ein- oder ausgeblendet wird.
Template (Seite 1858)	-	-	Gibt die Vorlage für die Anzeige des Fensterinhalts des Objektes "Applikationsfenster" zurück.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.
WindowCloseEnabled (Seite 1986)	-	-	Legt fest, ob ein Fenster in Runtime geschlossen werden kann.
WindowMaximizeEnabled (Seite 1986)	-	-	TRUE, wenn das Objekt in Runtime maximiert werden kann.
WindowMovingEnabled (Seite 1987)	-	-	TRUE, wenn das Objekt in Runtime verschoben werden kann.
WindowOnTop (Seite 1987)	-	-	TRUE, wenn das Objekt in Runtime immer im Vordergrund bleibt.
WindowsContents (Seite 1988)	-	-	Gibt den Inhalt des Applikationsfenster wieder.
WindowSizingEnabled (Seite 1988)	-	-	Legt fest, ob in Runtime die Größe des angegebenen Objekts verändert werden kann.

Tabelle 3-88 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Slider (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Schieberegler" dar. Das Slider-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMISlider

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-89 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackPicture	-	-	Legt eine Grafik für den Hintergrund fest.
BarBackColor (Seite 1581)	P, A	P, A	Legt die Farbe des Balkenhintergrunds im angegebenen Objekt fest.
BarColor (Seite 1584)	P, A	P, A	Legt die Farbe des Reglers im Objekt "Slider" fest.

Eigenschaften	Read	Write	Beschreibung
BorderBrightColor3D (Seite 1589)	P, A	P, A	Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest.
BorderInnerStyle3D (Seite 1597)	P	P	Legt die Darstellung des inneren Teils des Objektrahmens fest.
BorderInnerWidth3D (Seite 1597)	P, A	P, A	Legt die Breite des Innenrahmens für die 3D-Darstellung des angegebenen Objekts fest.
BorderOuterStyle3D (Seite 1598)	P	P	Legt die Darstellung des äußeren Teils des Objektrahmens fest.
BorderOuterWidth3D (Seite 1598)	P, A	P, A	Legt die Breite des Außenrahmens für die 3D-Darstellung des angegebenen Objekts fest.
BorderShadeColor3D (Seite 1599)	P, A	P, A	Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P, A	P, A	Legt die Linienstärke des Objekts fest.
Caption (Seite 1608)	P, A	P, A	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
ContinuousChange (Seite 1627)	P	P	Legt fest, ob der Wert der Eigenschaft "ProcessValue" beim Loslassen der Maustaste oder sofort beim Ändern der Schieberposition in Runtime übergeben wird.
Enabled (Seite 1644)	P, A	P, A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FocusColor (Seite 1668)	P, A	P, A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth (Seite 1669)	P, A	P, A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Font	-	-	Legt die Schriftart fest oder gibt sie zurück.
ForeColor (Seite 1675)	P, A	P, A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
LabelColor (Seite 1702)	P, A	P, A	Legt die Farbe der Skalenbeschriftung im Objekt "Slider" fest.
Layer (Seite 1707)	-	-	Legt den Wert der X-Koordinate des Objekts fest.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
MaximumValue (Seite 1729)	P, A	P, A	Legt den Maximalwert der Skala im angegebenen Objekt fest.
MinimumValue (Seite 1745)	P, A	P, A	Legt den Minimalwert der Skala im angegebenen Objekt fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
PositionFont	-	-	Legt die Schriftart für die Beschriftung fest.
ProcessValue (Seite 1779)	P, A	P, A	Legt die Voreinstellung für den anzuzeigenden Wert fest.
ShowBar (Seite 1818)	P, A	P, A	Legt fest, ob der angezeigte Prozesswert im Objekt "Slider" zusätzlich durch einen ausgefüllten Balken angezeigt wird.
ShowPosition (Seite 1822)	P, A	P, A	Legt fest, ob der Wert der aktuellen Reglerposition zusätzlich numerisch angezeigt wird.
ShowThumb (Seite 1827)	P, A	P, A	Legt fest, ob der Schieberegler des Objekts "Slider" angezeigt wird.
ShowTickLabels (Seite 1828)	P, A	P, A	Legt fest, ob in der Skala die Beschriftung angezeigt wird.

Eigenschaften	Read	Write	Beschreibung
ShowTicks (Seite 1828)	P, A	P, A	Legt fest, ob die Teilstriche in der Skala des Objekts angezeigt werden.
ThumbBackColor (Seite 1862)	P, A	P, A	Legt die Hintergrundfarbe des Reglers im Objekt "Slider" fest.
ThumbPicture	-	-	Legt eine Grafik für den Schieber fest.
TickStyle (Seite 1863)	P	P	Legt die Darstellung der Skala fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-90 Methoden

Methoden	Gültigkeit	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

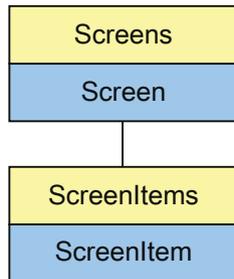
Siehe auch

ObjectName (Seite 2036)

Font (Seite 1669)

SmartClientView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Sm@rtClient-Anzeige" dar. Das SmartClientView-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMISmartClientView

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-91 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AllowMenu (Seite 1562)	-	-	Legt fest, ob das Kontextmenü zur Steuerung der Sm@rtClient-Anzeige aktiviert wird.
ConnectionType (Seite 1626)	-	-	Legt die Art des Verbinders fest. Sie können zwischen zwei Verbindungstypen wählen.
ConnectOnStart	-	-	Legt fest, dass Verbindung bei Start hergestellt wird.

Eigenschaften	Read	Write	Beschreibung
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
Height (Seite 1680)	A	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
LocalCursor (Seite 1724)	-	-	Legt fest, ob die Cursordaten separat übertragen werden, um die Performance zu steigern.
MachineName (Seite 1728)	A	A	Legt die Netzwerkennung des zu überwachenden Geräts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Password (Seite 1773)	A	A	Legt das Passwort für den Aufbau der Fernüberwachung fest.
ScaleDenominator (Seite 1797)	-	-	Legt den Zähler für die Skalierung auf dem Client fest.
ScaleNumerator (Seite 1794)	-	-	Legt den Zähler für die Skalierung auf dem Client fest.
Scaling (Seite 1797)	-	-	TRUE, wenn zusätzlich eine Skala zur Darstellung der Werte verwendet wird.
ServerScale (Seite 1815)	-	-	Legt fest, ob sich die Sm@rtClient-Anzeige vergrößern oder verkleinern lässt.
Shared (Seite 1816)	-	-	Legt fest, dass ein Bediengerät von mehreren Sm@rtClient-Anzeigen gemeinsam genutzt wird.
ShowControls	-	-	Legt fest, dass eine Symbolleiste angezeigt wird.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
UseCursorKeyScroll	-	-	Legt fest, ob die Hintergrundfarbe des Objekts beim Drucken verwendet wird.
ViewOnly (Seite 1973)	A	A	Legt fest, ob die Sm@rtClient-Anzeige zur Fernüberwachung oder zur Fernwartung eingesetzt wird.
Visible (Seite 1974) *	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

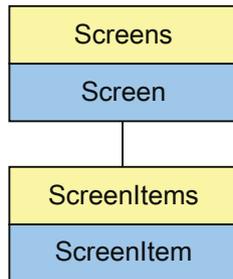
* im ES nicht sichtbar

Tabelle 3-92 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	A	Aktiviert das Permanentfenster oder das Grundbild.

StatusForce (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Status/Steuern" dar. Das StatusForce-Objekt ist Element der ScreenItems-Auflistung.

Wenn Sie die Einstellungen für dieses Objekt mit einer benutzerdefinierten Funktion ändern, bleiben die geänderten Einstellungen auch nach einem Neuaufruf des Bildes mit diesem Objekt erhalten.

Typkennzeichnung in VBS

HMIStatusForce

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-93 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Appearance	-	-	Legt die Gestaltung für dieses NC-Teilprogramm fest.
BackColor	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
ColumnsMoveable	-	-	Legt fest, dass die Spalten verschiebbar sind.
CountVisibleItems (Seite 1630) *	-	-	Legt fest, wie viele Zeilen die Auswahlliste enthält.

Eigenschaften	Read	Write	Beschreibung
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FitToSize	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FocusColor	A	A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth	A	A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
PLCFilter	-	-	Legt den Steuerungstyp fest.
SelectionBackColor (Seite 1808)	A	A	Legt die Hintergrundfarbe der selektierten Zellen fest.
SelectionForeColor (Seite 1809)	A	A	Legt die Vordergrundfarbe der selektierten Zellen fest.
SetOfVisibleColumns	-	-	Legt die sichtbaren Spalten fest.
ShowReadButton	-	-	Legt fest, dass die Schaltfläche "Lesen" angezeigt wird.
ShowTableGridlines (Seite 1827)	A	A	Legt fest, ob in der Tabelle des angegebenen Objekts Gitterlinien angezeigt werden.
ShowWriteButton	-	-	Legt fest, dass die Schaltfläche "Schreiben" angezeigt wird.
TableBackColor (Seite 1850)	A	A	Legt die Hintergrundfarbe der Tabellenzellen des angegebenen Objekts fest.
TableFont	-	-	Legt die Schriftart in der Tabelle fest.
TableForeColor	A	A	??
TableHeaderBackColor (Seite 1854)	A	A	Legt die Hintergrundfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
TableHeaderFont	-	-	Legt die Textfarbe der Überschrift fest.
TableHeaderForeColor (Seite 1855)	A	A	Legt die Textfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974) *	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

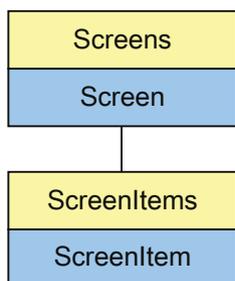
* in ES nicht sichtbar

Tabelle 3-94 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	A	Aktiviert das Permanentfenster oder das Grundbild.

Switch (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Schalter" dar. Das Switch-Objekt ist Element der ScreenItems-Auflistung.

Die Verfügbarkeit der folgenden Objekteigenschaften hängen vom gewählten Modus ab:

Eigenschaft	"Schalter mit Text"	"Schalter mit Grafik"	"Schalter"
CaptionColor	--	--	x
CaptionText	--	--	x
HorizontalAlignment	x	--	--
InnerBackColorOff	--	--	x
InnerBackColorOn	--	--	x
TextOn	x	--	x
TextOff	x	--	x
VerticalAlignment	x	--	--

Typkennzeichnung in VBS

HMISwitch

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-95 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AboveUpperLimitColor (Seite 1545)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Oberer Grenzwert überschritten" fest.
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
AutoSizing	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.
BackColor (Seite 1573)	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577) *	-	-	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578) *	-	-	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579) *	-	-	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580) *	-	-	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BelowLowerLimitColor (Seite 1585)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Unterer Grenzwert unterschritten" fest.
BorderColor (Seite 1590) *	-	-	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592) *	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594) *	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595) *	-	-	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596) *	-	-	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle3D	A	A	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
BorderWidth (Seite 1601) *	-	-	Legt die Linienstärke des Objekts fest.
BorderWidth3D (Seite 1603) *	-	-	Legt die Breite des Rahmens für die 3D-Darstellung des angegebenen Objekts fest.
CaptionColor (Seite 1609)	A	A	Legt die Farbe des Texts fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CaptionFont	-	-	Legt den Zeichensatz für die Beschriftung fest.
CaptionText (Seite 1610)	A	A	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CornerStyle (Seite 1628) *	-	-	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
EdgeStyle *	-	-	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
FlashingColorOff (Seite 1662) *	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663) *	-	-	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665) *	-	-	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingOnLimitViolation	-	-	Legt fest, dass Blinken bei Grenzwertverletzung aktiviert ist.
FlashingRate (Seite 1666) *	-	-	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
FocusColor (Seite 1668)	A	A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth (Seite 1669)	A	A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
ForeColor (Seite 1675)	A	A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HelpText	A	-	Gibt den Tooltip zurück, der in Runtime als Bedienungshilfe für das angegebene Objekt angezeigt wird.
HorizontalAlignment (Seite 1694)	A	A	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
InnerBackColorOff (Seite 1698)	A	A	Legt die Farbe unter dem Schieber des Objekts "Switch" fest, wenn sich das Objekt im Zustand AUS befindet.
InnerBackColorOn (Seite 1699)	A	A	Legt die Farbe unter dem Schieber des Objekts "Switch" fest, wenn sich das Objekt im Zustand EIN befindet.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722) *	-	-	Legt die Form der Linieneinde fest.
Mode (Seite 1747)	-	-	Legt den Feldtyp des angegebenen Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
OnValue	-	-	Legt den Wert für den Zustand "EIN" im "Zwei-Zustände"-Modus fest.
PictureOff (Seite 1775)	-	-	Legt das Bild fest, das im Zustand "Aus" angezeigt wird.
PictureOn (Seite 1776)	-	-	Legt das Bild fest, das im Zustand "Ein" angezeigt wird.
ProcessValue	-	-	Legt die Voreinstellung für den anzuzeigenden Wert fest.
SwitchOrientation	-	-	Legt die Schaltrichtung des Objekts fest.
TextOff (Seite 1860)	A	A	Legt den Text fest, der im Zustand "Aus" des angegebenen Objekts angezeigt wird.
TextOn	A	A	Legt den Text fest, der im Zustand "Ein" des angegebenen Objekts angezeigt wird.
TextOrientation (Seite 1861) *	-	-	Legt die Textrichtung des angegebenen Objekts fest.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
VerticalAlignment (Seite 1971)	A	A	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.

Eigenschaften	Read	Write	Beschreibung
Visible (Seite 1974) *	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

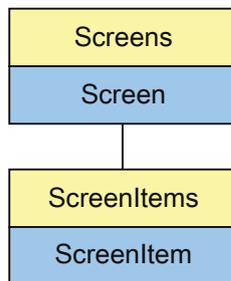
* im ES nicht sichtbar

Tabelle 3-96 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.

SymbolicIOField (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Symbolisches E/A-Feld" dar. Das SymbolicIOField-Objekt ist Element der ScreenItems-Auflistung.

Die Verfügbarkeit der folgenden Objekteigenschaften hängen vom gewählten Modus ab:

Eigenschaft	"Eingabe"	"Ausgabe"	"Eingabe/Ausgabe"	"Zwei Zustände"
BackColor	x	x	x	x
BorderColor	--	x	--	x
BorderWidth	--	--	--	x
BorderStyle3D	--	x	--	--

Eigenschaft	"Eingabe"	"Ausgabe"	"Eingabe/Ausgabe"	"Zwei Zustände"
Enabled	x	--	x	--
HelpText	x	--	x	--
VerticalAlignment	--	x	--	x
HorizontalAlignment	--	x	--	x

Typkennzeichnung in VBS

HMISymbolicIOField

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-97 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AboveUpperLimitColor (Seite 1545)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Oberer Grenzwert überschritten" fest.
AcceptOnExit (Seite 1546)	P	P	Legt fest, ob das Eingabefeld automatisch bestätigt wird, wenn es verlassen wird.
AdaptBorder (Seite 1553)	-	-	Legt fest, ob sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.
AskOperationMotive (Seite 1565)	P	P	Legt fest, ob der Grund für die Bedienung dieses Objekts mitprotokolliert wird.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BelowLowerLimitColor (Seite 1585)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Unterer Grenzwert unterschritten" fest.
BitNumber (Seite 1585)	P	P	Legt das Bit fest, dessen Zustand sich ändern muss, um eine Wertänderung auszulösen.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P, A	P, A	Legt die Linienfarbe des Objekts fest.

Eigenschaften	Read	Write	Beschreibung
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderStyle3D	A	A	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
BorderWidth (Seite 1601)	P, A	P, A	Legt die Linienstärke des Objekts fest.
BottomMargin	-	-	Legt den Abstand des Texts zum unteren Objektrand fest.
CaptionBackColor (Seite 1609)	P	P	Legt die Hintergrundfarbe der Titelzeile des angegebenen Objekts fest.
CaptionColor (Seite 1609)	P	P	Legt die Farbe des Texts fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
CountVisibleItems (Seite 1630)	P	P	Legt fest, wie viele Zeilen die Auswahlliste enthält.
CursorControl (Seite 1632)	P	P	Legt fest, ob der Mauszeiger nach dem Verlassen des Feldes auf das nächste Feld der TAB-Reihenfolge springt.
DrawInsideFrame (Seite 1640)	-	-	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	P	P	Legt den Linienstil des angegebenen Objekts fest.
EditOnFocus (Seite 1644)	P	P	Legt fest, ob die Dateneingabe sofort möglich ist, wenn das Eingabefeld mit der Taste "Tab" ausgewählt ist.
Enabled (Seite 1644)	P, A	P, A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FieldLength	-	-	Legt fest, dass das Feld "Feldlänge Zeichenkette" nur gelesen werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingOnLimitViolation	-	-	Legt fest, dass Blinken bei Grenzwertverletzung aktiviert ist.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
FocusColor (Seite 1668)	-	-	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
FocusWidth (Seite 1669)	-	-	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P, A	P, A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HelpText	A	-	Gibt den Tooltip zurück, der in Runtime als Bedienhilfe für das angegebene Objekt angezeigt wird.
HorizontalAlignment (Seite 1694)	P, A	P, A	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
InputValue (Seite 1700)	-	-	Definiert den im EA-Feld vom Benutzer eingegebenen Wert.
ItemBorderStyle (Seite 1701)	P	P	Legt die Linienart der Trennlinien in der Auswahlliste des angegebenen Objekts fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LeftMargin	-	-	Legt den Abstand des Texts zum linken Objektrand fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
LogOperation (Seite 1725)	P	P	Legt fest, ob nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.
Mode (Seite 1747)	P	P	Legt den Feldtyp des angegebenen Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
OnValue	-	-	Legt den Wert für den Zustand "EIN" im "Zwei-Zustände"-Modus fest.
ProcessValue (Seite 1779)	P	P	Legt die Voreinstellung für den anzuzeigenden Wert fest.
RightMargin	-	-	Legt den Abstand des Texts zum rechten Objektrand fest.
SelectBackColor (Seite 1801)	P	P	Legt die Hintergrundfarbe der selektierten Zellen fest.
SelectForeColor (Seite 1807)	P	P	Legt die Vordergrundfarbe der selektierten Zellen fest.
SeparatorBackColor (Seite 1812)	P	P	Legt die Hintergrundfarbe der durchbrochenen Trennlinien in der Auswahlliste des angegebenen Objekts fest.
SeparatorColor (Seite 1813)	P	P	Legt die Farbe der Trennlinien in der Auswahlliste des angegebenen Objekts fest.
SeparatorCornerStyle (Seite 1814)	P	P	Legt die die Form der Ecken für das Objekt vom Typ "ScreenItem" mit der Ausprägung "SymbolicIOField" fest.
SeparatorLineEndShapeStyle (Seite 1815)	P	P	Legt die die Form der Linienenden für das Objekt vom Typ "ScreenItem" mit der Ausprägung "SymbolicIOField" fest.

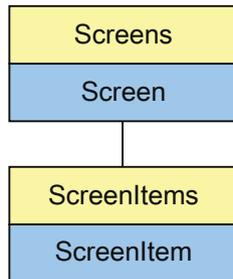
Eigenschaften	Read	Write	Beschreibung
SeparatorStyle (Seite 1814)	P	P	Legt die Linienart der Trennlinien in der Auswahlliste des angegebenen Objekts fest.
SeparatorWidth (Seite 1814)	P	P	Legt die Breite der Trennlinien in der Auswahlliste des angegebenen Objekts fest.
ShowBadTagState (Seite 1817)	P	P	Legt fest, ob das Objekt grau dargestellt wird, wenn die verbundene Variable einen schlechten QualityCode oder Variablenstatus besitzt.
ShowDropDownButton	-	-	Legt fest, dass eine Schaltfläche für die Auswahlliste angezeigt wird.
ShowDropDownList	-	-	Legt fest, dass der Eintrag in einer Auswahlliste gewählt wird.
TextList (Seite 1859)	-	-	Legt die Beschriftung für das Textfeld fest.
TextOff (Seite 1860)	-	-	Legt den Text fest, der im Zustand "Aus" des angegebenen Objekts angezeigt wird.
TextOn	-	-	Legt den Text fest, der im Zustand "Ein" des angegebenen Objekts angezeigt wird.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
TopMargin	-	-	Legt den Abstand des Texts zum oberen Objektrand fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
VerticalAlignment (Seite 1971)	P, A	P, A	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-98 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

SymbolLibrary (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Symbolbibliothek" dar. Das SymbolLibrary-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMISymbolLibrary

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-99 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AboveUpperLimitColor (Seite 1545)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Oberer Grenzwert überschritten" fest.
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.

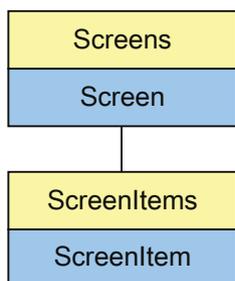
Eigenschaften	Read	Write	Beschreibung
BelowLowerLimitColor (Seite 1585)	-	-	Legt die Farbe des angegebenen Objekts für den Fall "Unterer Grenzwert unterschritten" fest.
BlinkColor (Seite 1586)	P, A	P, A	Legt die Farbe fest, mit der das Objekt "SymbolLibrary" blinkt.
BlinkMode (Seite 1586)	P	P	Legt die Art des Blinkbildes für das angegebene Objekt fest.
BlinkSpeed (Seite 1587)	P	P	Legt die Blinkfrequenz fest.
ChangeMouseCursor (Seite 1615)	P	P	Legt fest, wie sich das Aussehen des Mauszeigers in Runtime verändert, wenn er sich über dem Symbol befindet.
Enabled (Seite 1644)	P, A	P, A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillColorMode (Seite 1657)	P	P	Legt die Art des Vordergrundes für das angegebene Objekt fest.
FixedAspectRatio (Seite 1662)	P	P	Legt fest, ob beim Ändern der Symbolgröße das Seitenverhältnis erhalten bleibt oder veränderlich ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingOnLimitViolation	-	-	Legt fest, dass Blinken bei Grenzwertverletzung aktiviert ist.
Flip (Seite 1667)	P	P	Spiegelt das Symbol an der vertikalen und/oder an der horizontalen Mittelachse des Symbols.
ForeColor (Seite 1675)	P, A	P, A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
ProcessValue	-	-	Legt die Voreinstellung für den anzuzeigenden Wert fest.
Rotation (Seite 1785)	P	P	Legt den Drehwinkel des angegebenen Objekts in Grad fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-100 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

SystemDiagnoseView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "System-Diagnoseanzeige" dar. Das SystemDiagnoseView-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMISysDiagView

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-101 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	A	A	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackgroundColor	A	A	Legt die Hintergrundfarbe fest.
ComponentInfoText	A	A	Legt den Tooltip der System-Diagnoseanzeige in Runtime fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
InfoArea_BackgroundColor	A	A	Legt den Hintergrund der Raster-Steuerelemente fest.
InfoArea_ColumnsMovable	A	A	Legt die Beweglichkeit der Spalten fest.
InfoArea_DefaultTextColor	A	A	Legt die Textfarbe fest.
InfoArea_ErrorTextBackgroundColor	A	A	Legt die Hintergrundfarbe einer Zelle im Fehlerfall fest.
InfoArea_ErrorTextColor	A	A	Legt die Textfarbe einer Zelle im Fehlerfall fest.
InfoArea_FocusFrameColor	A	A	Legt die Farbe des Fokusrahmens fest.
InfoArea_FocusFrameWidth	A	A	Legt die Linienstärke des Focus-Rechtecks fest.
InfoArea_Font	A	A	Legt die Schriften der Rasteransichten fest.
InfoArea_ShowGridLines	A	A	Legt die Rasterlinien fest.
InfoArea_TableHeaderBackgroundCo- lor	A	A	Legt die Hintergrundfarbe des Tabellenkopfs fest.
InfoArea_TableHeaderTextColor	A	A	Legt die Textfarbe des Tabellenkopfs fest.
InputAddressText	A	A	Legt die Eingabe-Adresse fest.
ItemText_AKZ	A	A	Legt die Beschriftung der Spalte "AKZ" fest.
ItemText_Descriptor	A	A	Legt die Beschriftung der Spalte "Descriptor" fest.
ItemText_ErrorText	A	A	Legt die Beschriftung der Spalte "ErrorText" fest.
ItemText_InstallationDate	A	A	Legt die Beschriftung der Spalte "InstallationDate" fest.
ItemText_LADDR	A	A	Legt die Beschriftung der Spalte "LADDR" fest.
ItemText_Name	A	A	Legt die Beschriftung der Spalte "Name" fest.
ItemText_OKZ	A	A	Legt die Beschriftung der Spalte "OKZ" fest.
ItemText_OperationState	A	A	Legt die Beschriftung der Spalte "OperationState" fest.
ItemText_OrderID	A	A	Legt die Beschriftung der Spalte "OrderID" fest.
ItemText_Rack	A	A	Legt die Beschriftung der Spalte "Rack" fest.
ItemText_Slot	A	A	Legt die Beschriftung der Spalte "Slot" fest.
ItemText_SoftwareRevision	A	A	Legt die Beschriftung der Spalte "SoftwareRevision" fest.
ItemText_State	A	A	Legt die Beschriftung der Spalte "State" fest.
ItemText_Station	A	A	Legt die Beschriftung der Spalte "Station" fest.
ItemText_SubAddress	A	A	Legt die Beschriftung der Spalte "SubAddress" fest.
ItemText_SubSlot	A	A	Legt die Beschriftung der Spalte "SubSlot" fest.
ItemText_SubSystem	A	A	Legt die Beschriftung der Spalte "SubSystem" fest.
ItemText_Type	A	A	Legt die Beschriftung der Spalte "Type" fest.
IV_ShowItem_AKZ	A	A	Legt fest, ob die Spalte "AKZ" angezeigt wird.
IV_ShowItem_Descriptor	A	A	Legt fest, ob die Spalte "Descriptor" angezeigt wird.
IV_ShowItem_ErrorText	A	A	Legt fest, ob die Spalte "ErrorText" angezeigt wird.
IV_ShowItem_InstallationDate	A	A	Legt fest, ob die Spalte "InstallationDate" angezeigt wird.
IV_ShowItem_LADDR	A	A	Legt fest, ob die Spalte "LADDR" angezeigt wird.
IV_ShowItem_Name	A	A	Legt fest, ob die Spalte "Name" angezeigt wird.
IV_ShowItem_OKZ	A	A	Legt fest, ob die Spalte "OKZ" angezeigt wird.
IV_ShowItem_OperationState	A	A	Legt fest, ob die Spalte "OperationState" angezeigt wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
IV_ShowItem_OrderID	A	A	Legt fest, ob die Spalte "OrderID" angezeigt wird.
IV_ShowItem_Rack	A	A	Legt fest, ob die Spalte "Rack" angezeigt wird.
IV_ShowItem_Slot	A	A	Legt fest, ob die Spalte "Slot" angezeigt wird.
IV_ShowItem_SoftwareRevision	A	A	Legt fest, ob die Spalte "SoftwareRevision" angezeigt wird.
IV_ShowItem_State	A	A	Legt fest, ob die Spalte "State" angezeigt wird.
IV_ShowItem_Station	A	A	Legt fest, ob die Spalte "Station" angezeigt wird.
IV_ShowItem_SubAddress	A	A	Legt fest, ob die Spalte "SubAddress" angezeigt wird.
IV_ShowItem_SubSlot	A	A	Legt fest, ob die Spalte "SubSlot" angezeigt wird.
IV_ShowItem_SubSystem	A	A	Legt fest, ob die Spalte "SubSystem" angezeigt wird.
IV_ShowItem_Type	A	A	Legt fest, ob die Spalte "Type" angezeigt wird.
Layer (Seite 1707)	A	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
NavigationPath_Font	A	A	Legt die Schriftart des Navigationspfads fest.
NavigationPath_TextColor	A	A	Legt die Textfarbe des Navigationspfads fest.
ObjectName (Seite 2036)	A	-	Gibt den Objektnamen als STRING zurück.
OutputAddressText	A	A	Legt den Text für Ausgabe-Adresse fest.
ShowNavigationButtons	A	A	Legt fest, ob Navigationsschaltflächen angezeigt werden.
ShowPathInformation	A	A	Legt fest, ob der Navigationspfad angezeigt wird.
ShowSplittedView	A	A	Legt fest, ob die Anzeige geteilt wird.
TableHeaderFont	A	A	Legt die Textfarbe der Überschrift fest.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Type (Seite 2140)	A	-	Gibt den Typ des angegebenen Objekts als STRING zurück.
Parent (Seite 2122)	A	-	Gibt das Bild zurück, in dem das angegebene Objekt eingebettet ist.
UV_ShowItem_AKZ	A	A	Legt fest, ob die Spalte "AKZ" angezeigt wird.
UV_ShowItem_Descriptor	A	A	Legt fest, ob die Spalte "Descriptor" angezeigt wird.
UV_ShowItem_InstallationDate	A	A	Legt fest, ob die Spalte "InstallationDate" angezeigt wird.
UV_ShowItem_LADDR	A	A	Legt fest, ob die Spalte "LADDR" angezeigt wird.
UV_ShowItem_Name	A	A	Legt fest, ob die Spalte "Name" angezeigt wird.
UV_ShowItem_OKZ	A	A	Legt fest, ob die Spalte "OKZ" angezeigt wird.
UV_ShowItem_OperationState	A	A	Legt fest, ob die Spalte "OperationState" angezeigt wird.
UV_ShowItem_OrderID	A	A	Legt fest, ob die Spalte "OrderID" angezeigt wird.
UV_ShowItem_Rack	A	A	Legt fest, ob die Spalte "Rack" angezeigt wird.
UV_ShowItem_Slot	A	A	Legt fest, ob die Spalte "Slot" angezeigt wird.
UV_ShowItem_SoftwareRevision	A	A	Legt fest, ob die Spalte "SoftwareRevision" angezeigt wird.
UV_ShowItem_State	A	A	Legt fest, ob die Spalte "State" angezeigt wird.
UV_ShowItem_Station	A	A	Legt fest, ob die Spalte "Station" angezeigt wird.
UV_ShowItem_SubAddress	A	A	Legt fest, ob die Spalte "SubAddress" angezeigt wird.
UV_ShowItem_SubSlot	A	A	Legt fest, ob die Spalte "SubSlot" angezeigt wird.
UV_ShowItem_SubSystem	A	A	Legt fest, ob die Spalte "SubSystem" angezeigt wird.
UV_ShowItem_Type	A	A	Legt fest, ob die Spalte "Type" angezeigt wird.

Eigenschaften	Read	Write	Beschreibung
Visible (Seite 1974)	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	A	Legt fest, ob die Spalte "Slot" angezeigt wird.

Tabelle 3-102 Methoden

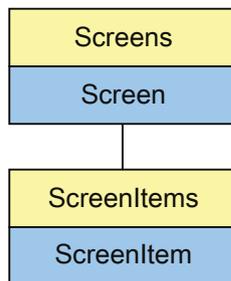
Methoden	Gült.	Beschreibung
Activate (Seite 2157)	A	Aktiviert das Permanentfenster oder das Grundbild.

Siehe auch

Name (Seite 1749)

SystemDiagnoseWindow (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "System-Diagnosefenster" dar. Das SystemDiagnoseWindow-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMISysDiagWindow

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-103 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	A	A	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackgroundColor	A	A	Legt die Hintergrundfarbe fest.
Caption (Seite 1608)	A	A	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CaptionActive	A	A	Legt fest, dass die Beschriftung angezeigt wird.
Closable	A	A	Legt fest, dass Control in Runtime geschlossen werden kann.
ComponentInfoText	A	A	Legt den Tooltip der System-Diagnoseanzeige in Runtime fest.
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
InfoArea_BackgroundColor	A	A	Legt den Hintergrund der Raster-Steuerelemente fest.
InfoArea_ColumnsMovable	A	A	Legt die Beweglichkeit der Spalten fest.
InfoArea_DefaultTextColor	A	A	Legt die Textfarbe fest.
InfoArea_ErrorTextBackgroundColor	A	A	Legt die Hintergrundfarbe einer Zelle im Fehlerfall fest.
InfoArea_ErrorTextColor	A	A	Legt die Textfarbe einer Zelle im Fehlerfall fest.
InfoArea_FocusFrameColor	A	A	Legt die Farbe des Fokusrahmens fest.
InfoArea_FocusFrameWidth	A	A	Legt die Linienstärke des Focus-Rechtecks fest.
InfoArea_Font	A	A	Legt die Schriften der Rasteransichten fest.
InfoArea_ShowGridLines	A	A	Legt die Rasterlinien fest.
InfoArea_TableHeaderBackgroundColor	A	A	Legt die Hintergrundfarbe des Tabellenkopfs fest.
InfoArea_TableHeaderTextColor	A	A	Legt die Textfarbe des Tabellenkopfs fest.
InputAddressText	A	A	Legt die Eingabe-Adresse fest.
ItemText_AKZ	A	A	Legt die Beschriftung der Spalte "AKZ" fest.
ItemText_Descriptor	A	A	Legt die Beschriftung der Spalte "Descriptor" fest.
ItemText_ErrorText	A	A	Legt die Beschriftung der Spalte "ErrorText" fest.
ItemText_InstallationDate	A	A	Legt die Beschriftung der Spalte "InstallationDate" fest.
ItemText_LADDR	A	A	Legt die Beschriftung der Spalte "LADDR" fest.
ItemText_Name	A	A	Legt die Beschriftung der Spalte "Name" fest.
ItemText_OKZ	A	A	Legt die Beschriftung der Spalte "OKZ" fest.
ItemText_OperationState	A	A	Legt die Beschriftung der Spalte "OperationState" fest.
ItemText_OrderID	A	A	Legt die Beschriftung der Spalte "OrderID" fest.

Eigenschaften	Read	Write	Beschreibung
ItemText_Rack	A	A	Legt die Beschriftung der Spalte "Rack" fest.
ItemText_Slot	A	A	Legt die Beschriftung der Spalte "Slot" fest.
ItemText_SoftwareRevision	A	A	Legt die Beschriftung der Spalte "SoftwareRevision" fest.
ItemText_State	A	A	Legt die Beschriftung der Spalte "State" fest.
ItemText_Station	A	A	Legt die Beschriftung der Spalte "Station" fest.
ItemText_SubAddress	A	A	Legt die Beschriftung der Spalte "SubAddress" fest.
ItemText_SubSlot	A	A	Legt die Beschriftung der Spalte "SubSlot" fest.
ItemText_SubSystem	A	A	Legt die Beschriftung der Spalte "SubSystem" fest.
ItemText_Type	A	A	Legt die Beschriftung der Spalte "Type" fest.
IV_ShowItem_AKZ	A	A	Legt fest, ob die Spalte "AKZ" angezeigt wird.
IV_ShowItem_Descriptor	A	A	Legt fest, ob die Spalte "Descriptor" angezeigt wird.
IV_ShowItem_ErrorText	A	A	Legt fest, ob die Spalte "ErrorText" angezeigt wird.
IV_ShowItem_InstallationDate	A	A	Legt fest, ob die Spalte "InstallationDate" angezeigt wird.
IV_ShowItem_LADDR	A	A	Legt fest, ob die Spalte "LADDR" angezeigt wird.
IV_ShowItem_Name	A	A	Legt fest, ob die Spalte "Name" angezeigt wird.
IV_ShowItem_OKZ	A	A	Legt fest, ob die Spalte "OKZ" angezeigt wird.
IV_ShowItem_OperationState	A	A	Legt fest, ob die Spalte "OperationState" angezeigt wird.
IV_ShowItem_OrderID	A	A	Legt fest, ob die Spalte "OrderID" angezeigt wird.
IV_ShowItem_Rack	A	A	Legt fest, ob die Spalte "Rack" angezeigt wird.
IV_ShowItem_Slot	A	A	Legt fest, ob die Spalte "Slot" angezeigt wird.
IV_ShowItem_SoftwareRevision	A	A	Legt fest, ob die Spalte "SoftwareRevision" angezeigt wird.
IV_ShowItem_State	A	A	Legt fest, ob die Spalte "State" angezeigt wird.
IV_ShowItem_Station	A	A	Legt fest, ob die Spalte "Station" angezeigt wird.
IV_ShowItem_SubAddress	A	A	Legt fest, ob die Spalte "SubAddress" angezeigt wird.
IV_ShowItem_SubSlot	A	A	Legt fest, ob die Spalte "SubSlot" angezeigt wird.
IV_ShowItem_SubSystem	A	A	Legt fest, ob die Spalte "SubSystem" angezeigt wird.
IV_ShowItem_Type	A	A	Legt fest, ob die Spalte "Type" angezeigt wird.
Layer (Seite 1707)	A	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Modal	A	A	Legt fest, dass das Meldfenster quittiert werden muss, bevor weitergearbeitet wird.
NavigationPath_Font	A	A	Legt die Schriftart des Navigationspfads fest.
NavigationPath_TextColor	A	A	Legt die Textfarbe des Navigationspfads fest.
ObjectName (Seite 2036)	A	-	Gibt den Objektnamen als STRING zurück.
OutputAddressText	A	A	Legt den Text für Ausgabe-Adresse fest.
Resizable	A	A	Legt fest, dass die Größe des Fensters veränderbar ist.
ShowNavigationButtons	A	A	Legt fest, ob Navigationsschaltflächen angezeigt werden.
ShowPathInformation	A	A	Legt fest, ob der Navigationspfad angezeigt wird.
ShowSplittedView	A	A	Legt fest, ob die Anzeige geteilt wird.

Eigenschaften	Read	Write	Beschreibung
TableHeaderFont	A	A	Legt die Textfarbe der Überschrift fest.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Type (Seite 2140)	A	-	Gibt den Typ des angegebenen Objekts als STRING zurück.
Parent (Seite 2122)	A	-	Gibt das Bild zurück, in dem das angegebene Objekt eingebettet ist.
UV_ShowItem_AKZ	A	A	Legt fest, ob die Spalte "AKZ" angezeigt wird.
UV_ShowItem_Descriptor	A	A	Legt fest, ob die Spalte "Descriptor" angezeigt wird.
UV_ShowItem_InstallationDate	A	A	Legt fest, ob die Spalte "InstallationDate" angezeigt wird.
UV_ShowItem_LADDR	A	A	Legt fest, ob die Spalte "LADDR" angezeigt wird.
UV_ShowItem_Name	A	A	Legt fest, ob die Spalte "Name" angezeigt wird.
UV_ShowItem_OKZ	A	A	Legt fest, ob die Spalte "OKZ" angezeigt wird.
UV_ShowItem_OperationState	A	A	Legt fest, ob die Spalte "OperationState" angezeigt wird.
UV_ShowItem_OrderID	A	A	Legt fest, ob die Spalte "OrderID" angezeigt wird.
UV_ShowItem_Rack	A	A	Legt fest, ob die Spalte "Rack" angezeigt wird.
UV_ShowItem_Slot	A	A	Legt fest, ob die Spalte "Slot" angezeigt wird.
UV_ShowItem_SoftwareRevision	A	A	Legt fest, ob die Spalte "SoftwareRevision" angezeigt wird.
UV_ShowItem_State	A	A	Legt fest, ob die Spalte "State" angezeigt wird.
UV_ShowItem_Station	A	A	Legt fest, ob die Spalte "Station" angezeigt wird.
UV_ShowItem_SubAddress	A	A	Legt fest, ob die Spalte "SubAddress" angezeigt wird.
UV_ShowItem_SubSlot	A	A	Legt fest, ob die Spalte "SubSlot" angezeigt wird.
UV_ShowItem_SubSystem	A	A	Legt fest, ob die Spalte "SubSystem" angezeigt wird.
UV_ShowItem_Type	A	A	Legt fest, ob die Spalte "Type" angezeigt wird.
Visible (Seite 1974)	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.
WindowMode	A	A	Legt den Fenstermodus fest.

Tabelle 3-104 Methoden

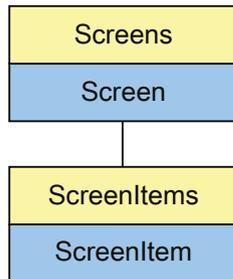
Methoden	Gült.	Beschreibung
Activate (Seite 2157)	A	Aktiviert das Permanentfenster oder das Grundbild.

Siehe auch

Name (Seite 1749)

TextField (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt das Objekt "Textfeld" dar. Das TextField-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMITextField

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-105 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AdaptBorder (Seite 1553)	P	P	Legt fest, ob sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P, A	P, A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P, A	P, A	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderStyle3D	A	A	Legt fest, ob das Objekt eine 3D-Rahmenschattierung besitzt.
BorderWidth (Seite 1601)	P, A	P, A	Legt die Linienstärke des Objekts fest.
BottomMargin	-	-	Legt den Abstand des Texts zum unteren Objektrand fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	A	A	Legt den Linienstil des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
FitToLargest	-	-	Legt fest, dass die Größe des Objekts in allen Sprachen von der Textlänge abhängig ist.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FlashingColorOff (Seite 1662)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
FlashingColorOn (Seite 1663)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
FlashingEnabled (Seite 1665)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
FlashingRate (Seite 1666)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
FontBold (Seite 1671)	P	P	Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.
FontItalic (Seite 1672)	P	P	Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
FontName (Seite 1673)	P	P	Legt die Schriftart des angegebenen Objekts fest.
FontSize (Seite 1673)	P	P	Legt die Schriftgröße des Texts des angegebenen Objekts fest.
FontUnderline (Seite 1674)	P	P	Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.
ForeColor (Seite 1675)	P, A	P, A	Legt die Schriftfarbe des Texts des angegebenen Objekts fest.
Height (Seite 1680)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalAlignment (Seite 1694)	P, A	P, A	Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P, A	P, A	Legt den Wert der X-Koordinate des Objekts fest.
LeftMargin	-	-	Legt den Abstand des Texts zum linken Objektrand fest.
LineEndShapeStyle (Seite 1722)	P	P	Legt die Form der Linienende fest.
LineWrap *	-	-	Legt den Zeilenumbruch innerhalb des Objekts fest.
Name (Seite 1749)	P, A	-	Gibt den Objektnamen als STRING zurück.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
RightMargin	-	-	Legt den Abstand des Texts zum rechten Objektrand fest.
RotationAngle (Seite 1786)	P	P	Legt den Rotationswinkel in Grad fest.
RotationCenterLeft (Seite 1786)	P	P	Legt die X-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
RotationCenterTop (Seite 1787)	P	P	Legt die Y-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
Text (Seite 1859)	P, A	P, A	Legt die Beschriftung für das Textfeld fest.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P, A	P, A	Legt den Wert der Y-Koordinate des Objekts fest.
TopMargin	-	-	Legt den Abstand des Texts zum oberen Objektrand fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
VerticalAlignment (Seite 1971)	P, A	P, A	Legt die vertikale Ausrichtung des Texts des angegebenen Objekts fest.
Visible (Seite 1974)	P, A	P, A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P, A	P, A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

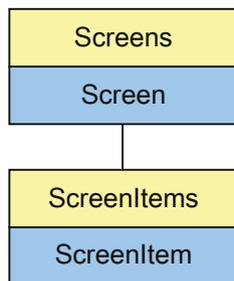
* im ES nicht sichtbar

Tabelle 3-106 Methoden

Methoden	Güt.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

TrendRulerControl (RT Professional)

Beschreibung



Stellt das Objekt "Wertetabelle" dar. Das TrendRulerControl-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMITrendRulerControl

Beispiel

Im folgenden Beispiel wird das Objekt mit dem Namen "Control1" um 16 Pixel nach rechts verschoben:

```
'VBS60
Dim objControl
Set objControl = ScreenItems("Control1")
objControl.Left = objControl.Left + 16
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-107 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
ApplyProjectSettingsForDesignMode	-	-	Legt fest, dass für das Design die Projekteinstellungen verwendet werden.
AutoCompleteColumns (Seite 1568)	P	P	Legt fest, ob Leerspalten angezeigt werden, wenn das Control breiter ist als die projektierten Spalten.
AutoCompleteRows (Seite 1568)	P	P	Legt fest, ob Leerzeilen angezeigt werden, wenn das Control länger ist als die Anzahl der projektierten Zeilen.
AutoPosition	P	P	Legt fest, dass die Position automatisch angepasst wird.
AutoSelectionColors (Seite 1569)	P	P	Legt fest, ob die Markierungsfarben für Zellen und Zeilen mit den vom System festgelegten Farben dargestellt werden.
AutoSelectionRectColor (Seite 1570)	P	P	Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.
AutoShow	P	P	Legt fest, dass die Wertetabelle automatisch eingeblendet wird.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BlockAlign *	P	P	Legt die Ausrichtung fest.
BlockAutoPrecisions *	P	P	Legt fest, dass die Anzahl der dargestellten Nachkommastellen einer Achse automatisch angepasst wird.
BlockCaption *	P	P	Legt die Überschrift fest.
BlockCount *	P	P	Legt die Anzahl der Blöcke fest.
BlockDateFormat *	P	P	Legt das Format für Datumsangaben fest.
BlockExponentialFormat *	P	P	Legt fest, dass Exponentialschreibweise verwendet wird.
BlockHideText *	P	P	Legt fest, ob der Text angezeigt wird.
BlockHideTitleText *	P	P	Legt fest, ob der Text angezeigt wird.
BlockId *	P	P	Legt die ID fest.
BlockIndex	P	P	Der Index legt fest, auf welchen Block sich andere Eigenschaften, wie z.B. Name, beziehen.

Eigenschaften	Read	Write	Beschreibung
BlockLength *	P	P	Legt die Anzahl der angezeigten Zeichen fest.
BlockName *	P	P	Legt den Namen eines Blockes fest.
BlockPrecisions *	P	P	Legt die Anzahl der angezeigten Nachkommastellen fest.
Blocks	-	-	Legt die Meldeblöcke fest.
BlockShowDate *	P	P	Legt fest, dass das Datum angezeigt wird.
BlockShowIcon *	P	P	Legt den Text für den Tooltip fest.
BlockShowTitleIcon *	P	P	Legt fest, dass der Titel als Symbol angezeigt wird.
BlockTimeFormat *	P	P	Legt das Format für Zeitangaben fest.
BlockUseSourceFormat *	P	P	Legt fest, dass das Quellformat verwendet wird.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
Caption (Seite 1608)	P	P	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CellCut (Seite 1611)	P	P	Legt fest, ob die Inhalte der Zellen abgekürzt werden, wenn die Zellen zu schmal sind.
CellSpaceBottom (Seite 1612)	P	P	Legt den Abstand von unten fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceLeft (Seite 1612)	P	P	Legt den Einzug Links fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceRight (Seite 1613)	P	P	Legt den Einzug Rechts fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceTop (Seite 1613)	P	P	Legt den Abstand von oben fest, der in den Zellen der Tabelle verwendet wird.
Closeable (Seite 1617)	P	P	Legt fest, dass Control in Runtime geschlossen werden kann.
ColumnAdd *	P	P	Fügt eine Spalte hinzu.
ColumnCount *	P	P	Zeigt die Anzahl der Spalten an.
ColumnIndex	P	P	Der Index legt fest, auf welche Spalte sich andere Eigenschaften, wie z.B. Anfangswert, beziehen.
ColumnName *	P	P	Legt den Namen einer Spalte fest.
ColumnRemove *	P	P	Löscht eine Spalte.
ColumnRepos *	P	P	Legt fest, dass die Spalte neu positioniert wird.
ColumnResize (Seite 1621)	P	P	Legt fest, ob Sie die Breite der Spalten ändern können.
ColumnScrollbar (Seite 1621)	P	P	Legt den Typ der Bildlaufleiste fest.
ColumnSort *	P	P	Legt die Art der Sortierung fest.
ColumnSortIndex *	P	P	Legt die Reihenfolge bei der Sortierung fest.
ColumnTitleAlignment	P	P	Legt fest, wie die Spaltenüberschriften ausgerichtet werden.
ColumnTitles (Seite 1625)	P	P	Legt fest, ob die Spaltenüberschrift angezeigt wird.
ColumnVisible *	P	P	Legt die Sichtbarkeit einer Spalte fest.
ControlDesignMode (Seite 1831)	P	P	Legt das Design fest.
ExportDirectoryChangeable (Seite 1650)	P	P	Legt fest, ob das Verzeichnis für den Datenexport in Runtime geändert werden kann.
ExportDirectoryname (Seite 1651)	P	P	Legt das Verzeichnis fest, in das die Runtime-Daten exportiert werden.

Eigenschaften	Read	Write	Beschreibung
ExportFileExtension (Seite 1651)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportFilename (Seite 1652)	P	P	Legt den Dateinamen der Datei fest, in welche die Runtime-Daten exportiert werden.
ExportFilenameChangeable (Seite 1652)	P	P	Legt fest, ob der Dateiname der Exportdatei in Runtime geändert werden kann.
ExportFormatGuid (Seite 1653)	P	P	Festgelegte Zuordnung von Ident-Nummer und Export-Provider.
ExportFormatName (Seite 1654)	P	P	Legt das Dateiformat für den Export fest. Zurzeit steht nur das "csv"-Dateiformat für den Export zur Verfügung.
ExportParameters (Seite 1654)	P	P	Legt die Parameter des ausgewählten Formats über den Dialog "Eigenschaften" fest.
ExportSelection (Seite 1655)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportShowDialog (Seite 1655)	P	P	Legt fest, ob der Dialog zum Datenexport in Runtime angezeigt wird.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
GridLineColor (Seite 1678)	P	P	Legt die Farbe der Rasterlinien fest.
GridLineWidth (Seite 1679)	P	P	Legt die Stärke der Trennlinien in Pixel fest.
Height (Seite 1680)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalGridLines (Seite 1695)	P	P	Legt fest, ob horizontale Trennlinien angezeigt werden.
IconSpace (Seite 1697)	P	P	Legt den Abstand zwischen Symbol und Text in den Zellen der Tabelle fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineColor (Seite 1721)	P	P	Gibt die Farbe für die Fenstertrennlinien an.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Moveable (Seite 1748)	P	P	Legt fest, ob das Fenster in Runtime bewegt werden kann.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
PrintJob	P	P	Legt einen Druckauftrag fest
RowScrollbar (Seite 1789)	P	P	Legt fest, ob Zeilen-Rollbalken angezeigt werden.
RowTitleAlignment	P	P	Legt fest, wie die Zeilenüberschriften ausgerichtet werden.
RowTitles	P	P	Legt fest, dass Zeilenköpfe angezeigt werden.
RTPersistence (Seite 1790)	P	P	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
RTPersistencePasswordLevel (Seite 2046)	P	P	Legt die Berechtigung fest, die in Runtime zur Online-Projektierung erforderlich ist.
RTPersistenceType (Seite 1790)	-	-	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
RulerType	P	P	Legt den Typ des Lineals fest.
SelectedCellColor (Seite 1802)	P	P	Gibt die Hintergrundfarbe der markierten Zelle an.
SelectedCellForeColor (Seite 1803)	P	P	Gibt die Schriftfarbe der markierten Zelle an.
SelectedRowColor (Seite 1804)	P	P	Gibt die Hintergrundfarbe der markierten Zeile an.
SelectedRowForeColor (Seite 1805)	P	P	Gibt die Schriftfarbe der markierten Zeile an.

Eigenschaften	Read	Write	Beschreibung
SelectedTitleColor (Seite 1805)	P	P	Gibt die Hintergrundfarbe der markierten Tabellenüberschrift an.
SelectedTitleForeColor (Seite 1806)	P	P	Gibt die Schriftfarbe der markierten Tabellenüberschrift an.
SelectionColoring (Seite 1808)	P	P	Legt fest, ob Markierungsfarben für Zelle oder Zeile verwendet werden.
SelectionRect (Seite 1810)	P	P	Legt fest, ob ein Auswahlrahmen für die markierten Zellen bzw. Zeilen verwendet wird.
SelectionRectColor (Seite 1810)	P	P	Legt die Farbe des Auswahlrechtecks im Meldfenster fest, wenn der SelectionType "1" ist.
SelectionRectWidth (Seite 1811)	P	P	Legt die Linienstärke des Auswahlrechtecks im Meldfenster fest, wenn der SelectionType "1" ist.
SelectionType (Seite 1812)	P	P	Legt fest, wie viele Zeilen Sie markieren können.
ShareSpaceWithSourceControl	P	P	Legt fest, dass die Wertetabelle im Darstellungsbereich der Datenquelle angezeigt wird.
ShowSortButton (Seite 1825)	P	P	Legt fest, ob die Sortiertaste über dem vertikalen Rollballen angezeigt wird.
ShowSortIcon (Seite 1825)	P	P	Legt fest, ob das Sortiersymbol angezeigt wird.
ShowSortIndex (Seite 1826)	P	P	
ShowTitle (Seite 1829)	P	P	Legt die Darstellung der Fensterüberschrift des Controls fest.
Sizeable (Seite 1831)	P	P	Legt fest, ob in Runtime die Größe des Objekts geändert werden kann.
SortSequence (Seite 1833)	P	P	Legt fest, wie die Sortierreihenfolge durch Mausklick verändert werden kann.
SourceControl	P	P	Legt die Datenquelle fest.
SourceControlType	P	P	Legt den Typ der Datenquelle fest.
StatusbarBackColor (Seite 1836)	P	P	Gibt die Hintergrundfarbe für die Statuszeile an.
StatusbarElementAdd (Seite 1836) *	P	P	Legt ein neues, benutzerdefiniertes Element der Statuszeile an.
StatusbarElementAutoSize (Seite 1837) *	P	P	Legt fest, ob die Breite des ausgewählten Elements der Statuszeile automatisch eingestellt wird.
StatusbarElementCount (Seite 1838) *	P	P	Gibt die Anzahl der projektierbaren Elemente der Statuszeile an.
StatusbarElementIconId (Seite 1838) *	P	P	Festgelegte Zuordnung von Ident-Nummer und Symbol eines Elementes der Statuszeile.
StatusbarElementId (Seite 1839) *	P	P	Eindeutige Ident-Nummer für das gewählte Element der Statuszeile.
StatusbarElementIndex (Seite 1840)	P	P	Referenziert ein Element der Statuszeile. Unter Verwendung dieser Eigenschaft können Sie einem bestimmten Element der Statuszeile die Werte anderer Eigenschaften zuweisen.
StatusbarElementName (Seite 1840) *	P	P	Zeigt den Objektnamen für das gewählte Element der Statuszeile an.
StatusbarElementRemove (Seite 1841) *	P	P	Entfernt das ausgewählte Element der Statuszeile.

Eigenschaften	Read	Write	Beschreibung
StatusbarElementRename (Seite 1842) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Statuszeile, das über die Eigenschaft "StatusbarElement-Index" referenziert wird.
StatusbarElementRepos *	P	P	Legt den Index fest, der dem Element zugeordnet werden soll.
StatusbarElements (Seite 1842)	-	-	Legt die in der Statuszeile darzustellenden Elemente fest.
StatusbarElementText	P	P	Legt den Text des momentan gewählten Elements der Statuszeile fest.
StatusbarElementTooltipText (Seite 1842) *	P	P	Legt den Text für den Tooltip des benutzerdefinierten Elements der Statuszeile fest.
StatusbarElementUserDefined (Seite 1843) *	P	P	Zeigt an, ob das Element der Statuszeile vom rojekteur als neues, benutzerdefiniertes Element hinzugefügt wurde.
StatusbarElementVisible (Seite 1844) *	P	P	Aktivieren Sie in der Liste die Elemente der Statuszeile, die sie in Runtime anzeigen wollen.
StatusbarElementWidth (Seite 1844) *	P	P	Gibt die Breite des ausgewählten Elements der Statuszeile in Pixel an.
StatusbarFont	-	-	Statuszeile Schriftart
StatusbarFontColor (Seite 1845)	P	P	Gibt die Schriftfarbe der Texte in der Statuszeile an.
StatusbarShowTooltips (Seite 1846)	P	P	Legt fest, ob in Runtime die Tooltips zu den Elementen der Statuszeile angezeigt werden.
StatusbarText (Seite 1846)	P	P	Festgelegter Text in der Statuszeile.
StatusbarUseBackColor (Seite 1847)	P	P	Legt fest, ob die Hintergrundfarbe der Statuszeile angezeigt wird.
StatusbarVisible (Seite 1848)	P	P	Legt fest, ob die Statuszeile des Controls angezeigt wird.
TableColor (Seite 1851)	P	P	Gibt die Hintergrundfarbe für die Zeilen an. Über die öffnen Sie den Farbauswahl-Dialog.
TableColor2 (Seite 1851)	P	P	Gibt die Hintergrundfarbe der "Zeilenfarbe 2" an.
TableForeColor (Seite 1852)	P	P	
TableForeColor2 (Seite 1853)	P	P	Gibt die Schriftfarbe der "Zeilenfarbe 2" an.
TimeBase (Seite 1866)	P	P	Legt die Zeitbasis fest.
TitleColor	P	P	Legt die Farbe der Titelzeile fest.
TitleCut (Seite 1884)	P	P	Legt fest, ob die Inhalte der Felder einer Titelleiste bei zu kleiner Spaltenbreite abgekürzt werden sollen.
TitleDarkShadowColor (Seite 1884)	P	P	Gibt die Farbe für die dunkle Seite der Schattierung an.
TitleForeColor (Seite 1885)	P	P	Gibt die Schriftfarbe der Tabellenüberschrift für den gewählten Zustand an.
TitleGridLineColor (Seite 1886)	P	P	Gibt die Farbe der Trennlinien in der Tabellenüberschrift an.
TitleLightShadowColor (Seite 1886)	P	P	Gibt die Farbe für die helle Seite der Schattierung an.
TitleSort (Seite 1887)	P	P	Legt fest, wie das Sortieren über Spaltenüberschrift ausgelöst wird.
TitleStyle (Seite 1888)	P	P	Legt fest, ob eine Schattierungsfarbe für die Tabellenüberschrift verwendet wird.
ToolBarAlignment (Seite 1895)	P	P	Legt die Position der Symbolleiste fest oder gibt sie zurück.
ToolBarBackColor (Seite 1895)	P	P	Gibt die Hintergrundfarbe der Symbolleiste an.
ToolBarButtonActive (Seite 1896) *	P	P	Legt fest, ob die mit der Taste verbundene Funktion in Runtime aktiv geschaltet ist.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
ToolBarButtonAdd (Seite 1897) *	P	P	Legt eine neue, benutzerdefinierte Tastenfunktion an.
ToolBarButtonBeginGroup (Seite 1897) *	P	P	Legt fest, ob vor der ausgewählten Tastenfunktion ein Trennzeichen eingefügt wird.
ToolBarButtonClick	P	P	Klickt eine Schaltfläche der Symbolleiste.
ToolBarButtonCount (Seite 1898) *	P	P	Gibt die Anzahl der projektierbaren Tastenfunktionen an.
ToolBarButtonEnabled (Seite 1899) *	P	P	Legt fest, ob eine benutzerdefinierte Taste der Symbolleiste bedienbar ist.
ToolBarButtonHotKey (Seite 1899) *	P	P	Zeigt den Hotkey für die ausgewählte Tastenfunktion an.
ToolBarButtonId (Seite 1900) *	P	P	Eindeutige Ident-Nummer für die gewählte Tastenfunktion.
ToolBarButtonIndex (Seite 1900)	P	P	Referenziert eine Tastenfunktion.
ToolBarButtonLocked (Seite 1901) *	P	P	Legt für eine benutzerdefinierte Taste der Symbolleiste fest, ob der eingerastete, gedrückte Zustand der Taste dargestellt wird.
ToolBarButtonName (Seite 1902) *	P	P	Zeigt den Namen für die ausgewählte Tastenfunktion an.
ToolBarButtonPasswordLevel (Seite 1902) *	P	P	Zeigt die Berechtigung für die ausgewählte Tastenfunktion an.
ToolBarButtonRemove (Seite 1903) *	P	P	Entfernt die ausgewählte Tastenfunktion aus der Liste.
ToolBarButtonRename (Seite 1903) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Symbolleiste, das über die Eigenschaft "ToolBarButtonIndex" referenziert wird.
ToolBarButtonRepos (Seite 1904) *	P	P	Ändert die Reihenfolge der Tastenfunktionen.
ToolBarButtons	-	-	Legt fest, welche Schaltflächen für die Symbolleiste verfügbar sind.
ToolBarButtonTooltipText (Seite 1905) *	P	P	Legt den Text für den Tooltip der Taste fest.
ToolBarButtonUserDefined (Seite 1905) *	P	P	Zeigt, ob die Taste der Symbolleiste vom Projekteur als neue, benutzerdefinierte Taste hinzugefügt wurde.
ToolBarButtonVisible *	P	P	Legt fest, dass die Schaltfläche sichtbar ist.
ToolBarShowTooltips (Seite 1906)	P	P	Legt fest, ob in Runtime die Tooltips zu den Tastenfunktionen angezeigt werden.
ToolBarUseBackColor (Seite 1906)	P	P	Legt fest, ob die Hintergrundfarbe der Symbolleiste angezeigt wird.
ToolBarUseHotKeys (Seite 1907)	P	P	Legt fest, ob die Hotkeys für die Tastenfunktionen in Runtime aktiviert sind.
ToolBarVisible (Seite 1908)	P	P	Legt fest, ob die Symbolleiste des Controls angezeigt wird.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
UseSelectedTitleColor (Seite 1959)	P	P	Legt fest, ob eine Markierungsfarbe für die Tabellenüberschriften von markierten Tabellenzellen verwendet wird.
UseSourceBackColors	P	P	Legt fest, dass die Hintergrundfarbe der Datenquelle verwendet wird.
UseSourceForeColor	P	P	Legt fest, dass die Vordergrundfarbe der Datenquelle verwendet wird.
UseTableColor2 (Seite 1959)	P	P	Legt fest, ob eine zweite Zeilenfarbe bei der Darstellung der Tabelle verwendet wird.
VerticalGridLines (Seite 1972)	P	P	Legt fest, ob vertikale Trennlinien angezeigt werden.

Eigenschaften	Read	Write	Beschreibung
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

* im ES nicht sichtbar

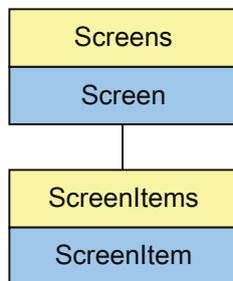
Tabelle 3-108 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.
Export (Seite 2170)	P, A	Führt die Funktion der Schaltfläche "Archiv exportieren" oder "Daten exportieren" des Controls aus.
GetRow (Seite 2182)	P	Gibt das mit der Zeilennummer bezeichnete Zeilen-Objekt der auf Tabellen basierenden Controls als Typ "ICCAxDataRow" zurück.
GetRowCollection (Seite 2183)	P	Gibt die Auflistung aller Zeilen-Objekte der auf Tabellen basierenden Controls als Typ "ICCAxDataRowCollection" zurück.
GetRulerBlock (Seite 2184)	P	Gibt das mit Namen oder Index bezeichnete Block-Objekt der Auswertetabelle als Typ "ICCAxRulerBlock" zurück.
GetRulerBlockCollection (Seite 2185)	P	Gibt die Auflistung aller Block-Objekte der Auswertetabelle als Typ "ICCAxCollection" zurück.
GetRulerColumn (Seite 2187)	P	Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Auswertetabelle als Typ "ICCAxRulerColumn" zurück.
GetRulerColumnCollection (Seite 2188)	P	Gibt die Auflistung aller Spalten-Objekte der Auswertetabelle als Typ "ICCAxCollection" zurück.
GetSelectedRow (Seite 2190)	P	Gibt das markierte Zeilen-Objekt eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.
GetSelectedRows (Seite 2191)	P	Gibt bei Mehrfachselektion die markierten Zeilen-Objekte eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.
GetStatisticAreaColumn (Seite 2193)	P	Gibt das mit Namen oder Index bezeichnete Spalten-Objekt des Statistikbereichfensters der Auswertetabelle als Typ "ICCAxRulerColumn" zurück.
GetStatisticAreaColumnCollection (Seite 2194)	P	Gibt die Auflistung aller Spalten-Objekte des Statistikbereichfensters der Auswertetabelle als Typ "ICCAxCollection" zurück.
GetStatisticResultColumn (Seite 2195)	P	Gibt das mit Namen oder Index bezeichnete Spalten-Objekt des Statistikfensters der Auswertetabelle als Typ "ICCAxRulerColumn" zurück.
GetStatisticResultColumnCollection (Seite 2196)	P	Gibt die Auflistung aller Spalten-Objekte des Statistikfensters der Auswertetabelle als Typ "ICCAxCollection" zurück.
GetStatusBarElement (Seite 2197)	P, A	Gibt das mit Namen oder Index bezeichnete Element der Statuszeile des Controls als Typ "ICCAxStatusBarElement" zurück.
GetStatusBarElementCollection (Seite 2198)	P, A	Gibt die Auflistung aller Elemente der Statuszeile des Controls als Typ "ICCAxCollection" zurück.
GetToolBarButton (Seite 2205)	P, A	Gibt die mit Namen oder Index bezeichnete Tastenfunktion der Symbolleiste des Controls als Typ "ICCAxToolBarButton" zurück.
GetToolBarButtonCollection (Seite 2207)	P, A	Gibt die Auflistung aller Tastenfunktionen der Symbolleiste des Controls als Typ "ICCAxCollection" zurück.

Methoden	Gült.	Beschreibung
Print (Seite 2236)	P, A	Führt die Funktion der Schaltfläche "Drucken" des Controls aus.
SelectAll (Seite 2251)	P	Markiert alle Zeilen in einem tabellenbasierten Control.
SelectRow (Seite 2252)	P	Markiert eine bestimmte Zeile in einem tabellenbasierten Control.
ShowHelp (Seite 2255)	P, A	Führt die Funktion der Schaltfläche "Hilfe" des Controls aus.
ShowPropertyDialog (Seite 2260)	P, A	Führt die Funktion der Schaltfläche "Konfigurationsdialog" des Controls aus.
UnselectAll (Seite 2268)	P	Entfernt alle Markierungen in den Zellen eines tabellenbasierten Controls.
UnselectRow (Seite 2269)	P	Entfernt die Markierungen in einer bestimmten Zelle eines tabellenbasierten Controls.

TrendView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Kurvenanzeige" dar. Das TrendView-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMITrendView

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-109 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
AxisXBunchCount	-	-	Legt die Schrittweite der großen Teilstriche für die X-Achse fest.
AxisXMarkCount	-	-	Legt die Schrittweite der Teilstriche für die X-Achse fest.
AxisXNoOfDigits	-	-	
AxisXShowBunchValues	-	-	Legt fest, dass die Skalenbeschriftung der Zwischenwerte für die X-Achse angezeigt werden.
AxisXStyle *	-	-	
AxisY1BunchCount	-	-	Legt die Schrittweite der großen Teilstriche für die Y-Achse links fest.
AxisY1MarkCount	-	-	Legt die Schrittweite der Teilstriche für die Y-Achse links fest.
AxisY1ShowBunchValues	-	-	Legt fest, dass die Skalenbeschriftung der Zwischenwerte für die Y-Achse links angezeigt werden.
AxisY2BunchCount	-	-	Legt die Schrittweite der großen Teilstriche für die Y-Achse rechts fest.
AxisY2MarkCount	-	-	Legt die Schrittweite der Teilstriche für die Y-Achse rechts fest.
AxisY2ShowBunchValues	-	-	Legt fest, dass die Skalenbeschriftung der Zwischenwerte für Y-Achse rechts angezeigt wird.
BackColor	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
ColumnsMoveable	-	-	Legt fest, dass die Spalten verschiebbar sind.
CountVisibleItems (Seite 1630)	-	-	Legt fest, wie viele Zeilen die Auswahlliste enthält.
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EnableNavigateButtons	-	-	
EnableNavigateKeys	-	-	Legt fest, dass mit Tastaturbedienung navigiert wird.
FitToSize	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
FocusColor	A	A	Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
FocusWidth	A	A	Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	A	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HelpText (Seite 1685)	A	-	Gibt den Tooltip zurück, der in Runtime als Bedienungshilfe für das angegebene Objekt angezeigt wird.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Look3D	-	-	Legt fest, dass 3D-Darstellung aktiviert ist.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
RulerColor (Seite 1791)	A	A	Legt die Farbe für die Skaleneinteilung (Hilfslinie) der Achsbeschriftung im Objekt "OnlineTrendControl" fest.
ScaleColor (Seite 1792)	A	A	Legt die Farbe der Skala des angegebenen Objekts fest.
SelectionBackColor *	-	-	Legt die Hintergrundfarbe der selektierten Zellen fest.
SelectionForeColor *	-	-	Legt die Vordergrundfarbe der selektierten Zellen fest.
ShowRuler	A	A	Legt fest, ob für die Achsbeschriftung des Objekts "OnlineTrendControl" eine Skaleneinteilung (Hilfslinie) eingeblendet wird.
ShowTableGridLines	-	--	Legt fest, dass die Rasterlinien der Tabelle angezeigt werden.
ShowTimeAxis	-	-	Legt fest, dass die X-Achse angezeigt wird.
ShowTimeAxisLabeling	-	-	Legt fest, dass die Beschriftung für die X-Achse angezeigt wird.
ShowValueAxis1	-	-	Legt fest, dass die linke Y-Achse angezeigt wird.
ShowValueAxis1Label	-	-	Legt fest, dass die Beschriftung der linken Y-Achse angezeigt wird.
ShowValueAxis2	-	-	Legt fest, dass die rechte Y-Achse angezeigt wird.
ShowValueAxis2Label	-	-	Legt den Fenstertitel fest.
ShowValueTable	-	-	Legt fest, dass die Wertetabelle angezeigt wird.
ShowY1HlpLine	-	-	Legt fest, dass die Hilfslinie angezeigt wird.
ShowY2HlpLine	-	-	Legt fest, dass die Hilfslinie angezeigt wird.
TableBackColor (Seite 1850)	A	A	Legt die Hintergrundfarbe der Tabellenzellen des angegebenen Objekts fest.
TableFont	-	-	Legt die Schriftart in der Tabelle fest.
TableGridLineColor (Seite 1854)	A	A	Legt die Farbe der Gitterlinien in der Tabelle des angegebenen Objekts fest.
TableHeaderBackColor (Seite 1854)	A	A	Legt die Hintergrundfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
TableHeaderFont	-	-	Legt die Textfarbe der Überschrift fest.
TableHeaderForeColor (Seite 1855)	A	A	Legt die Textfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
TagForExternalTime	-	-	Legt die Variable für die externe Uhrzeit fest.
TimeAxisBegin	-	-	Legt den Anfangswert für die Zeitachse fest.
TimeAxisBeginTime	-	-	Legt den Startzeitpunkt der Darstellung der angegebenen Kurve fest.
TimeAxisCountPoints	-	-	Legt die Anzahl der Werte der Zeitachse fest.
TimeAxisEnd	-	-	Legt den Endwert für die Zeitachse fest.

Eigenschaften	Read	Write	Beschreibung
TimeAxisMode	-	-	Legt den Modus der Zeitachse fest.
TimeAxisRange (Seite 1865)	-	-	Zeitspanne, die von der Zeitachse angezeigt wird.
TimeAxisSide	-	-	Legt die Seite für neue Werte auf der Zeitachse fest.
ToolBarButtons	-	-	Legt fest, welche Schaltflächen für die Symbolleiste verfügbar sind.
ToolBarEnabled *	-	-	Legt fest, ob die Kurvenanzeige eine Symbolleiste hat.
ToolBarStyle	-	-	Legt das Erscheinungsbild der Symbolleiste fest.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
TrendsForPrinting *	-	-	
ValueAxis1AutoRange *	-	-	Legt fest, dass der Datenbereich automatisch angepasst wird.
ValueAxis1Begin	-	-	Legt den Anfangswert für die Achse fest.
ValueAxis1End	-	-	Legt den Endwert für die Achse fest.
ValueAxis1LabelLength	-	-	Legt die Länge der Achsenbeschriftung fest.
ValueAxis1Style *	-	-	
ValueAxis2AutoRange *	-	-	Legt fest, dass der Datenbereich automatisch angepasst wird.
ValueAxis2Begin	-	-	Legt den Anfangswert für die Achse fest.
ValueAxis2End	-	-	Legt den Endwert für die Achse fest.
ValueAxis2LabelLength	-	-	Legt die Länge der Achsenbeschriftung fest.
ValueAxis2Style *	-	-	Legt fest, dass der Datenbereich automatisch angepasst wird.
ValueY1HlpLine	-	-	Legt den Wert für die Hilfslinie fest.
ValueY2HlpLine	-	-	Wert für die Hilfslinie Y-Achse rechts.
Visible (Seite 1974)	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	-	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

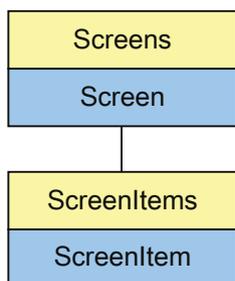
* im ES nicht sichtbar

Tabelle 3-110 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.

TubeArcObject (RT Professional)

Beschreibung



Stellt das Objekt "Rohrbogen" dar. Das TubeArcObject ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMITubeArcObject

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "TubeArcObject1" um 10 Pixel nach rechts verschoben:

```
'VBS24  
Dim objTubeArcObject  
Set objTubeArcObject = ScreenItems("TubeArcObject1")  
objTubeArcObject.Left = objTubeArcObject.Left + 10
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-111 Eigenschaften

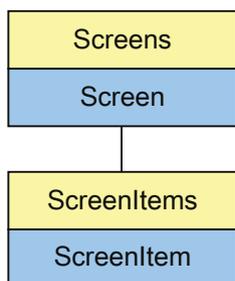
Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
Color (Seite 1618)	P	P	Legt die Linienfarbe des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EndAngle (Seite 1647)	P	P	Legt den Winkel fest, um den der Endpunkt des angegebenen Objekts von der Nulllage (0°) abweicht.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
RadiusHeight	P	P	Legt die Nebenachse des Objekts "Ellipse" fest.
RadiusWidth	P	P	Legt die Hauptachse des Objekts "Ellipse" fest.
StartAngle (Seite 1833)	P	P	Legt das SQL-Statement fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-112 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

TubeDoubleTeeObject (RT Professional)

Beschreibung



Stellt das Objekt "Doppel-T-Stück" dar. Das TubeDoubleTeeObject ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMITubeDoubleTeeObject

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "TubeDoubleTeeObject1" um 10 Pixel nach rechts verschoben:

```
'VBS21  
Dim objTubeDoubleTeeObject  
Set objTubeDoubleTeeObject = ScreenItems("TubeDoubleTeeObject1")  
objTubeDoubleTeeObject.Left = objTubeDoubleTeeObject.Left + 10
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-113 Eigenschaften

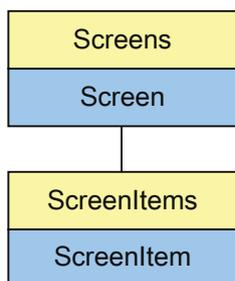
Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
Color (Seite 1618)	P	P	Legt die Linienfarbe des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-114 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

TubePolyline (RT Professional)

Beschreibung



Stellt das Objekt "Polygonrohr" dar. Das TubePolyline-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMITubePolyline

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "TubePolyline1" um 10 Pixel nach rechts verschoben:

```
'VBS24  
Dim objTubePolyline  
Set objTubePolyline = ScreenItems("TubePolyline1")  
objTubePolyline.Left = objTubePolyline.Left + 10
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-115 Eigenschaften

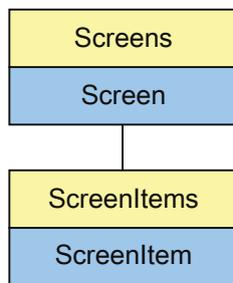
Eigenschaften	Read	Write	Beschreibung
ActualPointIndex (Seite 1551)	P	P	Legt die Nummer des aktuellen Eckpunkts des angegebenen Objekts fest.
ActualPointLeft (Seite 1551)	P	P	Legt die X-Koordinate des aktuellen Eckpunktes in Bezug auf den Bildursprung fest.
ActualPointTop (Seite 1552)	P	P	Legt die Y-Koordinate des aktuellen Eckpunkts in Bezug auf den Bildursprung fest.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
Color (Seite 1618)	P	P	Legt die Linienfarbe des angegebenen Objekts fest.
CornerStyle (Seite 1628)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
Points	-	-	Legt die Eckpunkte fest.
PointsCount (Seite 1777)	P	P	Legt die Anzahl der Eckpunkte der Polylinie oder des Vielecks fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-116 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

TubeTeeObject (RT Professional)

Beschreibung



Stellt das Objekt "T-Stück" dar. Das TubeTeeObject ist Element der ScreenItems-Auflistung. Objekt-Typ des ScreenItem-Objekts. Repräsentiert das Grafikobjekt "T-Stück".

Typkennzeichnung in VBS

HMITubeTeeObject

Verwendung

Im folgenden Beispiel wird das Objekt mit dem Namen "TubeTeeObject1" um 10 Pixel nach rechts verschoben:

```
'VBS21
Dim objTubeTeeObject
Set objTubeTeeObject = ScreenItems("TubeTeeObject1")
objTubeTeeObject.Left = objTubeTeeObject.Left + 10
```

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-117 Eigenschaften

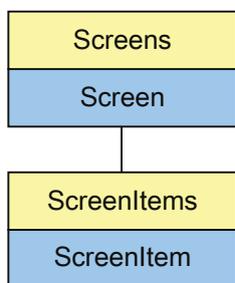
Eigenschaften	Read	Write	Beschreibung
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
Color (Seite 1618)	P	P	Legt die Linienfarbe des angegebenen Objekts fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
RotationAngle (Seite 1786)	P	P	Legt den Rotationswinkel in Grad fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-118 Methoden

Methoden	Güt.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

UserArchiveControl (RT Professional)

Beschreibung



Stellt das Objekt "Rezepturanzeige" dar. Das UserArchiveControl-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIUserArchiveControl

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-119 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
ApplyProjectSettingsForDesignMode	-	-	Legt fest, dass für das Design die Projekteinstellungen verwendet werden.
ArchiveName *	P	P	Legt den Archivnamen fest.
ArchiveType	P	-	Legt den Archivtyp fest.
AutoCompleteColumns (Seite 1568)	P	P	Legt fest, ob Leerspalten angezeigt werden, wenn das Control breiter ist als die projektierten Spalten.
AutoCompleteRows (Seite 1568)	P	P	Legt fest, ob Leerzeilen angezeigt werden, wenn das Control länger ist als die Anzahl der projektierten Zeilen.
AutoSelectionColors (Seite 1569)	P	P	Legt fest, ob die Markierungsfarben für Zellen und Zeilen mit den vom System festgelegten Farben dargestellt werden.
AutoSelectionRectColor (Seite 1570)	P	P	Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
Blocks	-	-	Legt die Meldeblöcke fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
Caption (Seite 1608)	P	P	Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
CellCut (Seite 1611)	P	P	Legt fest, ob die Inhalte der Zellen abgekürzt werden, wenn die Zellen zu schmal sind.
CellSpaceBottom (Seite 1612)	P	P	Legt den Abstand von unten fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceLeft (Seite 1612)	P	P	Legt den Einzug Links fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceRight (Seite 1613)	P	P	Legt den Einzug Rechts fest, der in den Zellen der Tabelle verwendet wird.
CellSpaceTop (Seite 1613)	P	P	Legt den Abstand von oben fest, der in den Zellen der Tabelle verwendet wird.
Closeable (Seite 1617)	P	P	Legt fest, dass Control in Runtime geschlossen werden kann.
ColumnAlign (Seite 1620) *	P	P	Legt fest, wie die ausgewählte Spalte ausgerichtet wird.
ColumnAutoPrecisions *	P	P	Legt fest, dass die Anzahl der dargestellten Nachkommastellen einer Achse automatisch angepasst wird.

Eigenschaften	Read	Write	Beschreibung
ColumnCaption *	P	P	Legt die Überschrift fest.
ColumnCount *	P	-	Zeigt die Anzahl der Spalten an.
ColumnDateFormat (Seite 1620) *	-	P	Legt das Anzeigeformat für Datumsangaben fest.
ColumnExponentialFormat *	P	P	Legt fest, dass in einer Spalte Exponentialschreibweise verwendet wird.
ColumnHideText *	P	P	Legt fest, ob der Text angezeigt wird.
ColumnHideTitleText *	P	P	Legt fest, ob der Text angezeigt wird.
ColumnIndex	P	P	Der Index legt fest, auf welche Spalte sich andere Eigenschaften, wie z.B. Anfangswert, beziehen.
ColumnLeadingZeros *	P	P	Legt fest, dass führende Nullstellen angezeigt werden.
ColumnLength *	P	P	Legt die Anzahl der angezeigten Zeichen fest.
ColumnName *	P	-	Zeigt den Namen der ausgewählten Spalte an.
ColumnPrecisions *	P	P	Legt die Anzahl der Nachkommastellen fest.
ColumnReadOnly *	P	P	Legt fest, dass die Werte der Spalte nur gelesen werden können.
ColumnRepos *	P	P	Legt fest, dass die Spalte neu positioniert wird.
ColumnResize (Seite 1621)	P	P	Legt fest, ob Sie die Breite der Spalten ändern können.
ColumnScrollbar (Seite 1621)	P	P	Legt den Typ der Bildlaufleiste fest.
ColumnShowDate *	P	P	Legt fest, dass das Datum angezeigt wird.
ColumnShowIcon *	P	P	Legt fest, dass ein Symbol angezeigt wird.
ColumnShowTitleIcon *	P	P	Legt fest, dass ein Symbol angezeigt wird.
ColumnSort *	P	P	Legt die Art der Sortierung fest.
ColumnSortIndex *	P	P	Legt die Reihenfolge bei der Sortierung fest.
ColumnTimeFormat *	-	P	Legt das Anzeigeformat für Zeitangaben fest.
ColumnTitleAlignment	P	P	Legt fest, wie die Spaltenüberschriften ausgerichtet werden.
ColumnTitles (Seite 1625)	P	P	Legt fest, ob die Spaltenüberschrift angezeigt wird.
ColumnVisible *	P	P	Legt die Sichtbarkeit einer Spalte fest.
ControlDesignMode (Seite 1831)	-	P	Legt das Design fest.
DataSource	-	-	Legt die Datenquelle fest.
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
EnableDelete	P	P	Legt fest, dass Daten gelöscht werden können.
EnableEdit (Seite 1646)	P	P	Legt fest, ob Sie die angezeigten Daten in Runtime ändern können.
EnableInsert	P	P	Legt fest, dass Daten eingefügt werden können.
ExportDirectoryChangeable (Seite 1650)	P	P	Legt fest, ob das Verzeichnis für den Datenexport in Runtime geändert werden kann.
ExportDirectoryname (Seite 1651)	P	P	Legt das Verzeichnis fest, in das die Runtime-Daten exportiert werden.
ExportFileExtension (Seite 1651)	P	P	Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.
ExportFilename (Seite 1652)	P	P	Legt den Dateinamen der Datei fest, in welche die Runtime-Daten exportiert werden.

Eigenschaften	Read	Write	Beschreibung
ExportFilenameChangeable (Seite 1652)	P	P	Legt fest, ob der Dateiname der Exportdatei in Runtime geändert werden kann.
ExportFormatGuid (Seite 1653)	P	P	Festgelegte Zuordnung von Ident-Nummer und Export-Provider.
ExportFormatName (Seite 1654)	P	P	Legt das Dateiformat für den Export fest. Zurzeit steht nur das "csv"-Dateiformat für den Export zur Verfügung.
ExportParameters (Seite 1654)	-	-	Legt die Parameter des ausgewählten Formats über den Dialog "Eigenschaften" fest.
ExportSelection (Seite 1655)	P	P	Legt fest, welche Runtime-Daten des Controls exportiert werden.
ExportShowDialog (Seite 1655)	P	P	Legt fest, ob der Dialog zum Datenexport in Runtime angezeigt wird.
Filters	-	-	Legt Kriterien in SQL-Syntax für die Datenbank fest.
FilterSQL	P	-	Legt ein SQL-Statement für das Filterkriterium fest.
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
GridLineColor (Seite 1678)	P	P	Legt die Farbe der Rasterlinien fest.
GridLineWidth (Seite 1679)	P	P	Legt die Stärke der Trennlinien in Pixel fest.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HorizontalGridLines (Seite 1695)	P	P	Legt fest, ob horizontale Trennlinien angezeigt werden.
IconSpace (Seite 1697)	P	P	Legt den Abstand zwischen Symbol und Text in den Zellen der Tabelle fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineColor (Seite 1721)	P	P	Gibt die Farbe für die Fenstertrennlinien an.
LineWidth (Seite 1723)	P	P	Legt die Linienstärke des angegebenen Objekts fest.
Moveable (Seite 1748)	P	P	Legt fest, ob das Fenster in Runtime bewegt werden kann.
ObjectName (Seite 2036)	P	-	Gibt den Objektnamen als STRING zurück.
PrintJobName (Seite 2043)	P	P	Legt einen Druckauftrag fest.
RowScrollbar (Seite 1789)	P	P	Legt fest, ob Zeilen-Rollbalken angezeigt werden.
RowTitleAlignment	P	P	Legt fest, wie die Zeilenüberschriften ausgerichtet werden.
RowTitles	P	P	Legt fest, dass Zeilenköpfe angezeigt werden.
RTPersistence (Seite 1790)	P	P	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
RTPersistencePasswordLevel (Seite 2046)	P	P	Legt die Berechtigung fest, die in Runtime zur Online-Projektierung erforderlich ist.
RTPersistenceType (Seite 1790)	-	-	Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.
SelectArchiveName *	P	P	Legt fest, dass das Archiv über den Namen definiert wird.
SelectedCellColor (Seite 1802)	P	P	Gibt die Hintergrundfarbe der markierten Zelle an.
SelectedCellForeColor (Seite 1803)	P	P	Gibt die Schriftfarbe der markierten Zelle an.
SelectedRowColor (Seite 1804)	P	P	Gibt die Hintergrundfarbe der markierten Zeile an.
SelectedRowForeColor (Seite 1805)	P	P	Gibt die Schriftfarbe der markierten Zeile an.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

Eigenschaften	Read	Write	Beschreibung
SelectedTitleColor (Seite 1805)	P	P	Gibt die Hintergrundfarbe der markierten Tabellenüberschrift an.
SelectedTitleForeColor (Seite 1806)	P	P	Gibt die Schriftfarbe der markierten Tabellenüberschrift an.
SelectionColoring (Seite 1808)	P	P	Legt fest, ob Markierungsfarben für Zelle oder Zeile verwendet werden.
SelectionRect (Seite 1810)	P	P	Legt fest, ob ein Auswahlrahmen für die markierten Zellen bzw. Zeilen verwendet wird.
SelectionRectColor (Seite 1810)	P	P	Legt die Farbe des Auswahlrechtecks im Meldfenster fest, wenn der SelectionType "1" ist.
SelectionRectWidth (Seite 1811)	P	P	Legt die Linienstärke des Auswahlrechtecks im Meldfenster fest, wenn der SelectionType "1" ist.
SelectionType (Seite 1812)	P	P	Legt fest, wie viele Zeilen Sie markieren können.
ShowSortButton (Seite 1825)	P	P	Legt fest, ob die Sortiertaste über dem vertikalen Rollballen angezeigt wird.
ShowSortIcon (Seite 1825)	P	P	Legt fest, ob das Sortiersymbol angezeigt wird.
ShowSortIndex (Seite 1826)	P	P	Legt fest, ob ein Sortierindex angezeigt wird.
ShowTitle (Seite 1829)	P	P	Legt die Darstellung der Fensterüberschrift des Controls fest.
Sizeable	P	P	Legt fest, ob in Runtime die Größe des Objekts geändert werden kann.
SortSequence	P	P	Legt fest, wie die Sortierreihenfolge durch Mausklick verändert werden kann.
StatusbarBackColor (Seite 1836)	P	P	Gibt die Hintergrundfarbe für die Statuszeile an.
StatusbarElementAdd (Seite 1836) *	-	P	Legt ein neues, benutzerdefiniertes Element der Statuszeile an.
StatusbarElementAutoSize (Seite 1837) *	P	P	Legt fest, ob die Breite des ausgewählten Elements der Statuszeile automatisch eingestellt wird.
StatusbarElementCount (Seite 1838) *	P	-	Gibt die Anzahl der projektierbaren Elemente der Statuszeile an.
StatusbarElementIconId (Seite 1838) *	P	P	Festgelegte Zuordnung von Ident-Nummer und Symbol eines Elementes der Statuszeile.
StatusbarElementId (Seite 1839) *	P	-	Gibt die Ident-Nummer für das gewählte Element der Statuszeile zurück.
StatusbarElementIndex (Seite 1840)	P	P	Referenziert ein Element der Statuszeile. Unter Verwendung dieser Eigenschaft können Sie einem bestimmten Element der Statuszeile die Werte anderer Eigenschaften zuweisen.
StatusbarElementName (Seite 1840) *	P	P	Zeigt den Objektnamen für das gewählte Element der Statuszeile an.
StatusbarElementRemove (Seite 1841) *	P	P	Entfernt das ausgewählte Element der Statuszeile.
StatusbarElementRename (Seite 1842) *	P	P	Ändert den Namen des benutzerdefinierten Elements der Statuszeile, das über die Eigenschaft "StatusbarElementIndex" referenziert wird.
StatusbarElementRepos *	P	P	Legt den Index fest, der dem Element zugeordnet werden soll.
StatusbarElements (Seite 1842)	-	-	Legt die in der Statuszeile darzustellenden Elemente fest.

Eigenschaften	Read	Write	Beschreibung
StatusbarElementText *	P	P	Legt den Text des momentan gewählten Elements der Statuszeile fest.
StatusbarElementTooltipText (Seite 1842) *	P	P	Legt den Text für den Tooltip des benutzerdefinierten Elements der Statuszeile fest.
StatusbarElementUserDefined (Seite 1843) *	P	-	Zeigt an, ob das Element der Statuszeile vom Projektneur als neues, benutzerdefiniertes Element hinzugefügt wurde.
StatusbarElementVisible (Seite 1844) *	P	P	Aktivieren Sie in der Liste die Elemente der Statuszeile, die sie in Runtime anzeigen wollen.
StatusbarElementWidth (Seite 1844) *	P	P	Gibt die Breite des ausgewählten Elements der Statuszeile in Pixel an.
StatusbarFont	-	-	Statuszeile Schriftart
StatusbarFontColor (Seite 1845)	P	P	Gibt die Schriftfarbe der Texte in der Statuszeile an.
StatusbarShowTooltips (Seite 1846)	P	P	Legt fest, ob in Runtime die Tooltips zu den Elementen der Statuszeile angezeigt werden.
StatusbarText (Seite 1846)	P	-	Gibt den Standardtext der Statuszeile zurück.
StatusbarUseBackColor (Seite 1847)	P	P	Legt fest, ob die Hintergrundfarbe der Statuszeile angezeigt wird.
StatusbarVisible (Seite 1848)	P	P	Legt fest, ob die Statuszeile des Controls angezeigt wird.
TableColor (Seite 1851)	P	P	Gibt die Hintergrundfarbe für die Zeilen an. Über die öffnen Sie den Farbauswahl-Dialog.
TableColor2 (Seite 1851)	P	P	Gibt die Hintergrundfarbe der "Zeilenfarbe 2" an.
TableForeColor (Seite 1852)	P	P	
TableForeColor2 (Seite 1853)	P	P	Gibt die Schriftfarbe der "Zeilenfarbe 2" an.
TimeBase (Seite 1866)	P	P	Legt die Zeitbasis fest.
TitleColor	P	P	Legt die Farbe der Titelzeile fest.
TitleCut (Seite 1884)	P	P	Legt fest, ob die Inhalte der Felder einer Titelleiste bei zu kleiner Spaltenbreite abgekürzt werden sollen.
TitleDarkShadowColor (Seite 1884)	P	P	Gibt die Farbe für die dunkle Seite der Schattierung an.
TitleForeColor (Seite 1885)	P	P	Gibt die Schriftfarbe der Tabellenüberschrift für den gewählten Zustand an.
TitleGridLineColor (Seite 1886)	P	P	Gibt die Farbe der Trennlinien in der Tabellenüberschrift an.
TitleLightShadowColor (Seite 1886)	P	P	Gibt die Farbe für die helle Seite der Schattierung an.
TitleSort (Seite 1887)	P	P	Legt fest, wie das Sortieren über Spaltenüberschrift ausgelöst wird.
TitleStyle (Seite 1888)	P	P	Legt fest, ob eine Schattierungsfarbe für die Tabellenüberschrift verwendet wird.
ToolBarAlignment (Seite 1895)	P	P	Legt die Position der Symbolleiste fest oder gibt sie zurück.
ToolBarBackColor (Seite 1895)	P	P	Gibt die Hintergrundfarbe der Symbolleiste an.
ToolBarButtonActive (Seite 1896) *	P	P	Legt fest, ob die mit der Taste verbundene Funktion in Runtime aktiv geschaltet ist.
ToolBarButtonAdd (Seite 1897) *	P	P	Legt eine neue, benutzerdefinierte Tastenfunktion an.
ToolBarButtonBeginGroup (Seite 1897) *	P	P	Legt fest, ob vor der ausgewählten Tastenfunktion ein Trennzeichen eingefügt wird.
ToolBarButtonClick	P	P	Klickt eine Schaltfläche der Symbolleiste.

Eigenschaften	Read	Write	Beschreibung
ToolBarButtonCount (Seite 1898) *	P	-	Gibt die Anzahl der in der Symbolleiste enthaltenen Schaltflächen zurück.
ToolBarButtonEnabled (Seite 1899) *	P	P	Legt fest, ob eine benutzerdefinierte Taste der Symbolleiste bedienbar ist.
ToolBarButtonHotKey (Seite 1899) *	P	P	Zeigt den Hotkey für die ausgewählte Tastenfunktion an.
ToolBarButtonId (Seite 1900) *	P	P	Eindeutige Ident-Nummer für die gewählte Tastenfunktion.
ToolBarButtonIndex (Seite 1900)	P	P	Referenziert eine Tastenfunktion.
ToolBarButtonLocked (Seite 1901) *	P	P	Legt für eine benutzerdefinierte Taste der Symbolleiste fest, ob der eingerastete, gedrückte Zustand der Taste dargestellt wird.
ToolBarButtonName (Seite 1902) *	P	-	Zeigt den Namen der ausgewählten Schaltfläche der Symbolleiste an.
ToolBarButtonPasswordLevel (Seite 1902) *	P	P	Zeigt die Berechtigung für die ausgewählte Tastenfunktion an.
ToolBarButtonRemove (Seite 1903) *	-	P	Entfernt die ausgewählte Tastenfunktion aus der Liste.
ToolBarButtonRename (Seite 1903) *	-	P	Ändert den Namen des benutzerdefinierten Elements der Symbolleiste, das über die Eigenschaft "ToolBarButtonIndex" referenziert wird.
ToolBarButtonRepos (Seite 1904) *	P	P	Ändert die Reihenfolge der Tastenfunktionen.
ToolBarButtons	-	-	Legt fest, welche Schaltflächen für die Symbolleiste verfügbar sind.
ToolBarButtonTooltipText (Seite 1905) *	P	P	Legt den Text für den Tooltip der Taste fest.
ToolBarButtonUserDefined (Seite 1905) *	P	-	Zeigt, ob die Taste der Symbolleiste vom Projektteur als neue, benutzerdefinierte Taste hinzugefügt wurde.
ToolBarButtonVisible *	P	P	Legt fest, dass die Schaltfläche sichtbar ist.
ToolBarShowTooltips (Seite 1906)	P	P	Legt fest, ob in Runtime die Tooltips zu den Tastenfunktionen angezeigt werden.
ToolBarUseBackColor (Seite 1906)	P	P	Legt fest, ob die Hintergrundfarbe der Symbolleiste angezeigt wird.
ToolBarUseHotKeys (Seite 1907)	P	P	Legt fest, ob die Hotkeys für die Tastenfunktionen in Runtime aktiviert sind.
ToolBarVisible (Seite 1908)	P	P	Legt fest, ob die Symbolleiste des Controls angezeigt wird.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
UseSelectedTitleColor (Seite 1959)	P	P	Legt fest, ob eine Markierungsfarbe für die Tabellenüberschriften von markierten Tabellenzellen verwendet wird.
UseTableColor2 (Seite 1959)	P	P	Legt fest, ob eine zweite Zeilenfarbe bei der Darstellung der Tabelle verwendet wird.
VerticalGridLines (Seite 1972)	P	P	Legt fest, ob vertikale Trennlinien angezeigt werden.
Visibility	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

* im ES nicht sichtbar

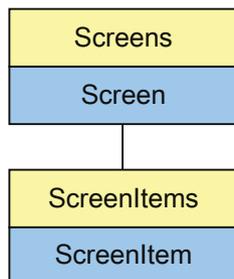
Tabelle 3-120 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft mit VBScript während Runtime.
CopyRows (Seite 2165)	P	Führt die Funktion der Schaltfläche "Zeilen kopieren" des Controls aus.
CutRows (Seite 2166)	P	Führt die Funktion der Schaltfläche "Zeilen ausschneiden" der Rezepturanzeige aus.
DeleteRows (Seite 2169)	P	Führt die Funktion der Schaltfläche "Zeilen löschen" der Rezepturanzeige aus.
Export (Seite 2170)	P, A	Führt die Funktion der Schaltfläche "Archiv exportieren" oder "Daten exportieren" des Controls aus.
GetColumn (Seite 2171)	P	Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Rezepturanzeige als Typ "ICCAxUAColumn" zurück.
GetColumnCollection (Seite 2172)	P	Gibt die Auflistung aller Spalten-Objekte der Rezepturanzeige als Typ "ICCAxCollection" zurück.
GetRow (Seite 2182)	P	Gibt das mit der Zeilennummer bezeichnete Zeilen-Objekt der auf Tabellen basierenden Controls als Typ "ICCAxDataRow" zurück.
GetRowCollection (Seite 2183)	P	Gibt die Auflistung aller Zeilen-Objekte der auf Tabellen basierenden Controls als Typ "ICCAxDataRowCollection" zurück.
GetSelectedRow (Seite 2190)	P	Gibt das markierte Zeilen-Objekt eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.
GetSelectedRows (Seite 2191)	P	Gibt bei Mehrfachselektion die markierten Zeilen-Objekte eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.
GetStatusBarElement (Seite 2197)	P, A	Gibt das mit Namen oder Index bezeichnete Element der Statuszeile des Controls als Typ "ICCAxStatusBarElement" zurück.
GetStatusBarElementCollection (Seite 2198)	P, A	Gibt die Auflistung aller Elemente der Statuszeile des Controls als Typ "ICCAxCollection" zurück.
GetToolBarButton (Seite 2205)	P, A	Gibt die mit Namen oder Index bezeichnete Tastenfunktion der Symbolleiste des Controls als Typ "ICCAxToolBarButton" zurück.
GetToolBarButtonCollection (Seite 2207)	P, A	Gibt die Auflistung aller Tastenfunktionen der Symbolleiste des Controls als Typ "ICCAxCollection" zurück.
MoveToFirst (Seite 2227)	P	Führt die Funktion der Schaltfläche "Erste Zeile" des Controls aus.
MoveToLast (Seite 2228)	P	Führt die Funktion der Schaltfläche "Letzter Datensatz" des Controls aus.
MoveToNext (Seite 2230)	P	Führt die Funktion der Schaltfläche "Nächster Datensatz" des Controls aus.
MoveToPrevious (Seite 2231)	P	Führt die Funktion der Schaltfläche "Vorhergehender Datensatz" des Controls aus.
PasteRows (Seite 2234)	P	Führt die Funktion der Schaltfläche "Zeilen einfügen" der Rezepturanzeige aus.
Print (Seite 2236)	P, A	Führt die Funktion der Schaltfläche "Drucken" des Controls aus.
ReadTags (Seite 2238)	P	Führt die Funktion der Schaltfläche "Variablen lesen" der Rezepturanzeige aus.
SelectAll (Seite 2251)	P	Markiert alle Zeilen in einem tabellenbasierten Control.
SelectRow (Seite 2252)	P	Markiert eine bestimmte Zeile in einem tabellenbasierten Control.
ServerExport (Seite 2252)	P	Führt die Funktion der Schaltfläche "Archiv exportieren" der Rezepturanzeige aus.
ServerImport (Seite 2253)	P	Führt die Funktion der Schaltfläche "Archiv importieren" der Rezepturanzeige aus.
ShowHelp (Seite 2255)	P, A	Führt die Funktion der Schaltfläche "Hilfe" des Controls aus.

Methoden	Gült.	Beschreibung
ShowPropertyDialog (Seite 2260)	P, A	Führt die Funktion der Schaltfläche "Konfigurationsdialog" des Controls aus.
ShowSelectArchive (Seite 2260)	P	Führt die Funktion der Schaltfläche "Datenanbindung wählen" der Rezepturanzeige aus.
ShowSelection (Seite 2261)	P	Führt die Funktion der Schaltfläche "Selektions-Dialog" der Rezepturanzeige aus.
ShowSelectTimeBase (Seite 2262)	P	Führt die Funktion der Schaltfläche "Zeitbasis-Dialog" der Rezepturanzeige aus.
ShowSort (Seite 2263)	P	Führt die Funktion der Schaltfläche "Sortier-Dialog" der Rezepturanzeige aus.
UnselectAll (Seite 2268)	P	Entfernt alle Markierungen in den Zellen eines tabellenbasierten Controls.
UnselectRow (Seite 2269)	P	Entfernt die Markierungen in einer bestimmten Zelle eines tabellenbasierten Controls.
WriteTags (Seite 2273)	P	Führt die Funktion der Schaltfläche "Variablen schreiben" der Rezepturanzeige aus.

UserView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Benutzeranzeige" dar. Das UserView-Objekt ist Element der ScreenItems-Auflistung.

Hinweis

Das Objekt "Einfache Benutzeranzeige" kann nicht mit einer benutzerdefinierten Funktion dynamisiert werden.

Typkennzeichnung in VBS

HMIUserView

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-121 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Appearance (Seite 1320)	-	-	Legt die Gestaltung für dieses NC-Teilprogramm fest.
Authorization (Seite 1566) *	-	-	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor	A	A	Legt die Hintergrundfarbe des angegebenen Objekts fest.
Columns	-	-	Legt fest, welche Spalten angezeigt werden.
ColumnsMoveable	-	-	Legt fest, dass die Spalten verschiebbar sind.
CountVisibleItems (Seite 1630)	-	-	Legt fest, wie viele Zeilen die Auswahlliste enthält.
Enabled	A	A	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FitToSize	-	-	Legt fest, dass die Objektgröße automatisch an den Inhalt angepasst wird.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
SelectionBackColor (Seite 1808)	A	A	Legt die Hintergrundfarbe der selektierten Zellen fest.
SelectionForeColor (Seite 1809)	A	A	Legt die Vordergrundfarbe der selektierten Zellen fest.
ShowTableGridlines (Seite 1827)	-	-	Legt fest, ob in der Tabelle des angegebenen Objekts Gitterlinien angezeigt werden.
TableBackColor (Seite 1850)	A	A	Legt die Hintergrundfarbe der Tabellenzellen des angegebenen Objekts fest.
TableFont	-	-	Legt die Schriftart in der Tabelle fest.
TableForeColor	A	A	
TableGridlineColor (Seite 1854)	A	A	Legt die Farbe der Rasterlinien fest.
TableHeaderBackColor (Seite 1854)	A	A	Legt die Hintergrundfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
TableHeaderFont	-	-	Legt die Textfarbe der Überschrift fest.

Eigenschaften	Read	Write	Beschreibung
TableHeaderForeColor (Seite 1855)	A	A	Legt die Textfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Visible (Seite 1974) *	A	A	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

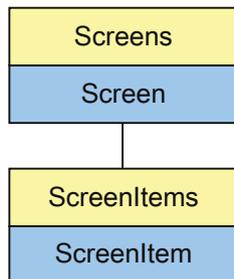
* im ES nicht sichtbar.

Tabelle 3-122 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

WindowSlider (RT Professional)

Beschreibung



Stellt das Objekt "Bildlaufleiste" dar. Das WindowSlider-Objekt ist Element der ScreenItems-Auflistung.

Typkennzeichnung in VBS

HMIWindowSlider

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-123 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
AskOperationMotive (Seite 1565)	P	P	Legt fest, ob der Grund für die Bedienung dieses Objekts mitprotokolliert wird.
Authorization (Seite 1566)	P	P	Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
BackColor (Seite 1573)	P	P	Legt die Hintergrundfarbe des angegebenen Objekts fest.
BackColorBottom (Seite 1574)	P	P	Legt die Farbe für den unteren/rechten Teil des Sliders fest.
BackColorTop (Seite 1575)	P	P	Legt die Farbe für den oberen/linken Teil des Sliders fest.
BackFillStyle (Seite 1576)	P	P	Legt das Füllmuster des angegebenen Objekts fest.
BackFlashingColorOff (Seite 1577)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.
BackFlashingColorOn (Seite 1578)	P	P	Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.
BackFlashingEnabled (Seite 1579)	P	P	Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.
BackFlashingRate (Seite 1580)	P	P	Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.
BorderBackColor (Seite 1587)	P	P	Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.
BorderColor (Seite 1590)	P	P	Legt die Linienfarbe des Objekts fest.
BorderFlashingColorOff (Seite 1592)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.
BorderFlashingColorOn (Seite 1594)	P	P	Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.
BorderFlashingEnabled (Seite 1595)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
BorderFlashingRate (Seite 1596)	P	P	Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.
BorderStyle (Seite 1600)	P	P	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
BorderWidth (Seite 1601)	P	P	Legt die Linienstärke des Objekts fest.
CornerStyle (Seite 1628)	-	-	Legt die Art der Rahmenlinien des angegebenen Objekts fest.
DrawInsideFrame (Seite 1640)	P	P	Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.
EdgeStyle (Seite 1642)	-	-	Legt den Linienstil des angegebenen Objekts fest.

3.5 VBS Objektmodell (Panels, Comfort Panels, RT Advanced, RT Professional)

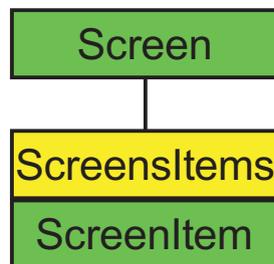
Eigenschaften	Read	Write	Beschreibung
Enabled (Seite 1644)	P	P	Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
FillPatternColor (Seite 1658)	P	P	Legt die Farbe des Füllmusters des angegebenen Objekts fest.
Flashing	-	-	Legt fest, dass der Textblock blinkt.
Height (Seite 1680)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
HighLimitColor (Seite 1686)	P	P	Legt die Farbe der oberen oder rechten Schaltfläche zum Scrollen in einer Bildlaufleiste fest.
JumpToLimitsAfterMouseClicked (Seite 1702)	P	P	Legt die Länge der langen Teilstriche einer Skala fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	P	P	Legt den Wert der X-Koordinate des Objekts fest.
LineEndShapeStyle (Seite 1722)	-	-	Legt die Form der Linienende fest.
LogOperation (Seite 1725)	P	P	Legt fest, ob nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.
LowLimitColor (Seite 1727)	P	P	Legt die Farbe der unteren oder linken Schaltfläche zum Scrollen in einer Bildlaufleiste fest.
MarginToBorder (Seite 1728)	P	P	Legt die Breite des 3D-Rahmens in Pixel fest.
MaximumValue (Seite 1729)	P	P	Legt den Maximalwert der Skala im angegebenen Objekt fest.
MinimumValue (Seite 1745)	P	P	Legt den Minimalwert der Skala im angegebenen Objekt fest.
Name (Seite 1749)	P	-	Gibt den Objektnamen als STRING zurück.
OperationSteps (Seite 1758)	P	P	Legt fest, um wie viele Schritte der Schieber der Bildlaufleiste mit einem Mausklick verschoben wird.
ProcessValue (Seite 1779)	P	P	Legt die Voreinstellung für den anzuzeigenden Wert fest.
RelativeFillLevel (Seite 1784)	P	P	Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.
ShowBadTagState (Seite 1817)	P	P	Legt fest, ob das Objekt grau dargestellt wird, wenn die verbundene Variable einen schlechten QualityCode oder Variablenstatus besitzt.
ShowFillLevel (Seite 1819)	P	P	Legt fest, ob das angegebene Objekt gefüllt wird.
StyleSettings (Seite 1849)	P	P	Legt fest, in welchem Stil das Objekt dargestellt wird.
TextOrientation (Seite 1861)	P	P	Legt die Textrichtung des angegebenen Objekts fest.
ThumbBackColor (Seite 1862)	P	P	Legt die Hintergrundfarbe des Reglers im Objekt "Slider" fest.
ToolTipText (Seite 1908)	P	P	Legt den Text des Tooltips fest.
Top (Seite 1909)	P	P	Legt den Wert der Y-Koordinate des Objekts fest.
Transparency (Seite 1913)	P	P	Legt den Prozentsatz der Transparenz des Objekts fest und gibt ihn zurück.
UseDesignColorSchema (Seite 1954)	P	P	Legt die Schriftart fest, die beim Drucken verwendet wird.
UseDesignShadowSettings (Seite 1955)	P	P	Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.
Visible (Seite 1974)	P	P	Legt fest, ob das angegebene Objekt sichtbar ist.
Width (Seite 1983)	P	P	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.
WindowsStyle (Seite 1988)	P	P	Legt fest, ob das Objekt im allgemeinen Windows-Stil dargestellt wird.

Tabelle 3-124 Methoden

Methoden	Gült.	Beschreibung
Activate (Seite 2157)	P, A	Aktiviert das Permanentfenster oder das Grundbild.
ActivateDynamic (Seite 2160)	P	Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime.
DeactivateDynamic (Seite 2167)	P	Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

WLANQualityView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "WLAN-Empfang" dar. Das WLANQualityView-Objekt ist Element der ScreenItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-125 Eigenschaften

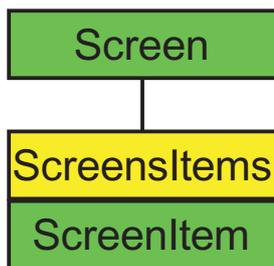
Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-126 Methoden

Methoden	Gült.	Beschreibung
???		

ZoneLabelView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Zonen-Name" dar. Das ZoneLabelView-Objekt ist Element der ScreenItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-127 Eigenschaften

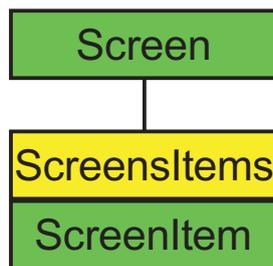
Eigenschaften	Read	Write	Beschreibung
Font (Seite 1669)	-	-	Legt die Schriftart fest oder gibt sie zurück.
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-128 Methoden

Methoden	Gült.	Beschreibung
???		

ZoneQualityView (Panels, Comfort Panels, RT Advanced)

Beschreibung



Stellt das Objekt "Zonen-Signal" dar. Das ZoneQualityView-Objekt ist Element der ScreensItems-Auflistung.

Abkürzung	Gültigkeit
Pa	Panels
A	Runtime Advanced
P	Runtime Professional

Tabelle 3-129 Eigenschaften

Eigenschaften	Read	Write	Beschreibung
Height (Seite 1680)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Höhe fest.
Layer (Seite 1707)	-	-	Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet.
Left (Seite 1711)	A	A	Legt den Wert der X-Koordinate des Objekts fest.
Name (Seite 1749)	A	-	Gibt den Objektnamen als STRING zurück.
Top (Seite 1909)	A	A	Legt den Wert der Y-Koordinate des Objekts fest.
Width (Seite 1983)	A	A	Legt bei den angegebenen Objekten "Ellipse", "Circle" und "Rectangle" die Weite fest.

Tabelle 3-130 Methoden

Methoden	Gült.	Beschreibung
?????		

3.5.5 Eigenschaften (Panels, Comfort Panels, RT Advanced, RT Professional)

3.5.5.1 Eigenschaften A (Panels, Comfort Panels, RT Advanced, RT Professional)

AboveUpperLimitColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des angegebenen Objekts für den Fall "Oberer Grenzwert überschritten" fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**AboveUpperLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- GraphicIOField
- Switch
- SymbolLibrary
- SymbolicIOField

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des angegebenen Objekts für den Fall "Oberer Grenzwert überschritten" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

AcceptOnExit (RT Professional)

Beschreibung

Legt fest, ob das Eingabefeld automatisch bestätigt wird, wenn es verlassen wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AcceptOnExit**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField
- SymbolicIOField

BOOLEAN

Optional. TRUE, wenn das Eingabefeld automatisch bestätigt wird, wenn es verlassen wird.

AcceptOnFull (RT Professional)

Beschreibung

Legt fest, ob das Eingabefeld automatisch verlassen und bestätigt wird, wenn die festgelegte Anzahl an Werten eingegeben wurde.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AcceptOnFull**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

BOOLEAN

Optional. TRUE, wenn das Eingabefeld automatisch verlassen und bestätigt wird, wenn die festgelegte Anzahl an Werten eingegeben wurde.

AccessPath (RT Professional)

Beschreibung

Gibt den Ablagepfad eines Bildes zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**AccessPath**

Object

Erforderlich. Ein Objekt vom Typ "Screen".

Beispiel

Im folgenden Beispiel wird der Pfad des Bildes "ScreenWindow1" ausgegeben:

```
'VBS67
Dim objScreen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
MsgBox objScreen.AccessPath
```

Siehe auch

Screen (Seite 1320)

ActiveProject (RT Professional)

Beschreibung

Gibt das angegebene Projekt zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**ActiveProject**

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

Siehe auch

HMIRuntime (Seite 1313)

Project (Seite 1319)

ActiveScreen (Panels, Comfort Panels, RT Advanced)

Beschreibung

Gibt ein Objekt vom Typ "Screen" zurück, welches das Bild darstellt, das gerade den Fokus besitzt.

Hinweis

Wenn Sie in einer benutzerdefinierten Funktion die Eigenschaft "ActiveScreen" abfragen, so kann es vorkommen, dass die Eigenschaft kein gültiges "Screen"-Objekt zurückliefert, sondern "Nothing", z. B. aufgrund eines ScreenSavers. Eine Systemmeldung dazu wird ausgegeben.

Zugriff in Runtime: Lesen

Syntax

Object.ActiveScreen

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

Bemerkungen

Welches Bild zurückgegeben wird, hängt davon ab, ob das Grundbild oder das Permanentfenster den Fokus besitzt.

Die ActiveScreen-Eigenschaft gibt NOTHING zurück, wenn kein Bild den Fokus enthält. Das ist z.B. der Fall, wenn ein anderes Fenster den Fokus besitzt. Mit der Anweisung "If not [Ausdruck] Is Nothing" können Sie abfragen, ob ein Bild zurückgegeben wird:

```
'VBS_Example_ActiveScreen
Dim objActiveScreen
Set objActiveScreen = HmiRuntime.ActiveScreen
If Not objActiveScreen Is Nothing Then
'found an active screen
HmiRuntime.Trace("There is an active screen." & vbCrLf)
Else
'found NO active screen
HmiRuntime.Trace("There is NO active screen." & vbCrLf)
End If
```

ActiveScreen (RT Professional)

Beschreibung

Gibt ein Objekt vom Typ "Screen" zurück, welches das Bild darstellt, das gerade den Fokus besitzt.

Hinweis

Wenn Sie in einer Funktion die Eigenschaft "ActiveScreen" abfragen, kann es z.B. aufgrund eines ScreenSavers vorkommen, dass die Eigenschaft kein gültiges "Screen"-Objekt sondern "Nothing" zurückliefert und eine Systemmeldung angezeigt wird.

Zugriff in Runtime: Lesen

Syntax

Object.ActiveScreen

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

Bemerkungen

Welches Bild zurückgegeben wird, hängt davon ab, ob das Grundbild oder das Permanentfenster den Fokus besitzt.

Die ActiveScreen-Eigenschaft gibt NOTHING zurück, wenn kein Bild den Fokus enthält. Das ist z.B. der Fall, wenn ein anderes Fenster den Fokus besitzt. Mit der Anweisung "If not [Ausdruck] Is Nothing" können Sie abfragen, ob ein Bild zurückgegeben wird:

```
'VBS_Example_ActiveScreen
Dim objActiveScreen
Set objActiveScreen = HmiRuntime.ActiveScreen
If Not objActiveScreen Is Nothing Then
'found an active screen
HmiRuntime.Trace("There is an active screen." & vbCrLf)
Else
'found NO active screen
HmiRuntime.Trace("There is NO active screen." & vbCrLf)
End If
```

ActiveScreenItem (RT Professional)

Beschreibung

Referenziert das Bildobjekt, das gerade den Fokus hat.

Nur wenn das Bild des entsprechenden "Screen"-Objektes gerade angewählt und über ein Eingabefeld verfügt, ist die "ActiveScreenItem"-Eigenschaft des "Screen"-Objektes zur Zeit mit einem gültigen "ScreenItem"-Objekt belegt. In allen anderen Fällen, wenn z.B. ein anderes Bild aus der "Screens"-Auflistung, ein unabhängiges Fenster innerhalb von WinCC oder eine andere Applikation angewählt ist, wird diese Eigenschaft an den Bildern nicht versorgt, d.h. mit "Nothing" belegt.

Verwendung

Sie verwenden das Objekt "ActiveScreenItem", um in Runtime die Eigenschaften des Objekts anzusprechen, das den Fokus hat.

ActiveScreenItem (Panels, Comfort Panels, RT Advanced)

Beschreibung

Referenziert das Bildobjekt, das gerade den Fokus hat.

Nur wenn das Bild des entsprechenden "Screen"-Objekts gerade angewählt ist und über ein Eingabefeld verfügt, ist die "ActiveScreenItem"-Eigenschaft des "Screen"-Objekts zur Zeit mit einem gültigen "ScreenItem"-Objekt belegt. In allen anderen Fällen, wenn z. B. ein anderes Bild aus der "Screens"-Auflistung, ein unabhängiges Fenster innerhalb von WinCC oder eine andere Applikation angewählt ist, wird diese Eigenschaft an den Bildern nicht versorgt, d.h. mit "Nothing" belegt.

Verwendung

Sie verwenden das Objekt "ActiveScreenItem", um in Runtime die Eigenschaften des Objekts anzusprechen, das den Fokus hat.

ActualPointIndex (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Nummer des aktuellen Eckpunkts des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ActualPointIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Polygon
- Polyline
- Tubepolyline

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Line

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer des aktuellen Eckpunkts des angegebenen Objekts festlegt.

ActualPointLeft (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die X-Koordinate des aktuellen Eckpunktes in Bezug auf den Bildursprung fest. Der Bildursprung liegt links oben im Objekt. Jeder Eckpunkt wird über einen Index identifiziert, der sich aus der Anzahl ("PointCount") der vorhandenen Eckpunkte ableitet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ActualPointLeft**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Polygon
- Polyline
- Tubepolyline

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Line

Int32

Optional. Ein Wert oder eine Konstante, welche die X-Koordinate des aktuellen Eckpunktes in Bezug auf den Bildursprung festlegt.

Bemerkungen

Eine Änderung des Werts kann sich auf die Eigenschaften "Width" (Objektbreite) und "Left" (X-Koordinate der Objektposition) auswirken.

ActualPointTop (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Y-Koordinate des aktuellen Eckpunktes in Bezug auf den Bildursprung fest. Der Bildursprung liegt links oben im Objekt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ActualPointTop**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Polygon
- Polyline
- Tubepolyline

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Line

Int32

Optional. Ein Wert oder eine Konstante, welche die Y-Koordinate des aktuellen Eckpunkts in Bezug auf den Bildursprung festlegt.

Bemerkungen

Eine Änderung des Werts kann sich auf die Eigenschaften "Height" (Objekthöhe) und "Top" (Y-Koordinate der Objektposition) auswirken.

AdaptBorder (RT Professional)

Beschreibung

Legt fest, ob sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AdaptBorder**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- IOField
- OptionGroup
- SymbolicIOField
- TextField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Checkbox

BOOLEAN

Optional. TRUE, wenn sich der Rahmen des angegebenen Objekts dynamisch an die Textgröße anpasst.

AdaptPicture (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AdaptScreenToWindow (RT Professional)

Beschreibung

Gibt an, ob das in einem Bildfenster dargestellte Bild sich in Runtime an die Größe des Bildfensters anpasst oder nicht.

Zugriff in Runtime: Lesen

Syntax

Object.**AdaptScreenToWindow**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Screenwindow

BOOLEAN

TRUE, wenn sich das Bild an die Bildfenstergröße anpasst.

FALSE, wenn sich das Bild nicht an die Bildfenstergröße anpasst.

AdaptWindowtoScreen (RT Professional)

Beschreibung

Gibt an, ob das Bildfenster sich dem in ihm dargestellten Bild in Runtime anpasst.

Zugriff in Runtime: Lesen

Syntax

Object.**AdaptWindowtoScreen**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Screenwindow

BOOLEAN

TRUE, wenn sich die Bildfenstergröße an das Bild anpasst.

FALSE, wenn sich die Bildfenstergröße nicht an das Bild anpasst.

Address (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Web-Adresse fest, die im HTML-Browser geöffnet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Address**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- HTML-Browser

STRING

Optional. Ein Wert oder eine Konstante, welche die Web-Adresse enthält.

AddressEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AdressPreview (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

Alarm (RT Professional)

Beschreibung

Legt den Grenzwert für die Speicherplatzanzeige fest, ab dem eine Meldung ausgegeben wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Alarm**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

Int32

Optional. Ein Wert oder eine Konstante, welche den Grenzwert für die Speicherplatzanzeige festlegt, ab dem eine Meldung ausgegeben wird.

AlarmClasses (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AlarmColor (RT Professional)

Beschreibung

Legt die Farbe fest, in der der Balken dargestellt wird, sobald der belegte Speicherplatz den Alarmbereich erreicht hat.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe festlegt, in der der Balken dargestellt wird, sobald der belegte Speicherplatz den Alarmbereich erreicht hat.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0-255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

AlarmID (RT Professional)

Beschreibung

Liefert die AlarmID des Alarm-Objekts zurück. Die AlarmID ist eindeutig und wird vom System vergeben.

AlarmID (readonly)

Siehe auch

Alarms (Auflistung) (Seite 1305)

AlarmLog (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AlarmLogs (RT Professional)

Beschreibung

Gibt ein Objekt vom Typ "AlarmLogs" zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**AlarmLogs**

Object

Erforderlich. Ein Objekt vom Typ "Logging".

Siehe auch

Logging (Seite 1318)

AlarmLowerLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den unteren Grenzwert fest, bei dem Alarm ausgelöst wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmLowerLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den unteren Grenzwert festlegt, bei dem Alarm ausgelöst wird.

Bemerkungen

Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft "AlarmLowerLimitRelative" fest.

Die Eigenschaft "AlarmLowerLimitEnable" legt fest, ob die Überwachung dieses Grenzwertes aktiviert ist.

AlarmLowerLimitColor (RT Professional)

Beschreibung

Legt die Balkenfarbe für den Grenzwert "AlarmLowerLimit" fest.

Die Eigenschaft "AlarmLowerLimitEnable" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmLowerLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Slider

Color

Optional. Ein Wert oder eine Konstante, welche die Balkenfarbe für den Grenzwert "AlarmLowerLimit" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

AlarmLowerLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Grenzwert "AlarmLowerLimit" überwacht wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmLowerLimitEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der Grenzwert "AlarmLowerLimit" überwacht wird.

Bemerkungen

Folgende Werte werden über die Eigenschaften "AlarmLowerLimit", "AlarmLowerLimitColor" und "AlarmLowerLimitRelative" festgelegt:

Grenzwert

Darstellung beim Erreichen des Grenzwerts

Art der Auswertung

AlarmLowerLimitRelative (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert, bei dem Alarm ausgelöst wird, prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmLowerLimitRelative**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der untere Grenzwert, bei dem Alarm ausgelöst wird, prozentual ausgewertet wird.

AlarmSource (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AlarmUpperLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den oberen Grenzwert fest, bei dem Alarm ausgelöst wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmUpperLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den oberen Grenzwert festlegt, bei dem Alarm ausgelöst wird.

Bemerkungen

Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft "AlarmUpperLimitRelative" fest.

Die Eigenschaft "AlarmUpperLimitEnable" legt fest, ob die Überwachung dieses Grenzwertes aktiviert ist.

AlarmUpperLimitColor (RT Professional)

Beschreibung

Legt die Balkenfarbe für den Grenzwert "AlarmUpperLimit" fest.

Die Eigenschaft "AlarmUpperLimitEnable" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmUpperLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Slider

Color

Optional. Ein Wert oder eine Konstante, welche die Balkenfarbe für den Grenzwert "AlarmUpperLimit" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

AlarmUpperLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Grenzwert "AlarmUpperLimit" überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmUpperLimitEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der Grenzwert "AlarmUpperLimit" überwacht wird.

Bemerkungen

Folgende Werte werden über die Eigenschaften "AlarmUpperLimit", "AlarmUpperLimitColor" und "AlarmUpperLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

AlarmUpperLimitRelative (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der obere Grenzwert, bei dem Alarm ausgelöst wird, prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmUpperLimitRelative**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der obere Grenzwert, bei dem Alarm ausgelöst wird, prozentual ausgewertet wird.

AllFilters (RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AllFiltersForHitlist (RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AllowEdit (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AllowMenu (RT Advanced)

Beschreibung

Kein Zugriff in Runtime

Analog (RT Professional)

Beschreibung

Legt fest, ob die Uhr als Analoguhr dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Analog**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

BOOLEAN

Optional. TRUE, wenn die Uhr als Analoguhr dargestellt wird.

AngleMax (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Winkel für das Skalenende des Objekts "Gauge" fest

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AngleMax**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

DOUBLE

Optional. Ein Wert oder eine Konstante, die den Winkel in Grad festlegt.

Bemerkungen

Anfang und Ende der Skalenteilung werden mit den Eigenschaften "AngleMin" und "AngleMax" in Winkelgraden beschrieben.

Der Wert der AngleMin-Eigenschaft muss immer kleiner als der Wert der AngleMax-Eigenschaft sein. Der Null-Grad-Winkel liegt auf der 3-Uhr-Position der Skalenscheibe. Positive Winkelwerte werden im Uhrzeigersinn gezählt.

AngleMin (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Winkel für den Skalenanfang des Objekts "Gauge" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AngleMin**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

DOUBLE

Optional. Ein Wert oder eine Konstante, die den Winkel in Grad festlegt.

Bemerkungen

Anfang und Ende der Skalenteilung werden mit den Eigenschaften "AngleMin" und "AngleMax" in Winkelgraden beschrieben.

Der Wert der AngleMin-Eigenschaft muss immer kleiner als der Wert der AngleMax-Eigenschaft sein.

Der Null-Grad-Winkel liegt auf der 3-Uhr-Position der Skalenscheibe. Positive Winkelwerte werden im Uhrzeigersinn gezählt.

Appearance (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ApplyProjectSettings (RT Professional)

Beschreibung

Legt fest, ob die Projekteinstellungen aus dem Editor "HMI-Meldungen" übernommen werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ApplyProjectSettings**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Die im Editor "HMI-Meldungen" projektierten Meldeblöcke werden mit ihren Eigenschaften in die Meldeanzeige übernommen. Die Meldeblöcke werden mit diesen Eigenschaften in der Meldeanzeige angezeigt.

FALSE: Die Eigenschaften werden nicht übernommen.

ApplyProjectSettingsForDesignMode (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AskOperationMotive (RT Professional)

Beschreibung

Legt fest, ob der Grund für die Bedienung dieses Objekts mitprotokolliert wird. Beim Bedienen des angegebenen Objekts in Runtime öffnet sich ein Dialog, in dem der Bediener den Grund der Bedienung als Text eingibt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AskOperationMotive**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ComboBox
- IOField
- ListBox
- SymbolicIOField
- WindowsSlider

BOOLEAN

Optional. TRUE, wenn der Grund für die Bedienung dieses Objekts mitprotokolliert wird.

Assignments (RT Professional)

Beschreibung

Legt eine Auflistung fest, welche die Zuordnungen zwischen dem Ausgabewert und dem tatsächlich auszugebenden Ausgabertext enthält. Die Zuordnungen sind abhängig von der eingestellten Listenart. Die Listenart legen Sie mit der ListType-Eigenschaft fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Assignments**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicIOField

STRING

Optional. Legt eine Auflistung fest, welche die Zuordnungen zwischen dem Ausgabewert und dem tatsächlich auszugebenden Ausgabertext enthält.

AssociatedS7GraphDBTag (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

Authorization (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Bedienrechte des angegebenen Objekts in Runtime fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Authorization**[=HMIRTAuthorization]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- CircularArc
- ComboBox
- DateTimeField
- Ellipse
- EllipseSegment
- EllipticalArc
- GraphicIOField
- GraphicView
- IOField
- Line
- ListBox
- MultiLineEdit
- OptionGroup
- Polygon
- Polyline
- RecipeView
- Rectangle
- RoundButton
- Slider
- StatusForce

- Switch
- SymbolLibrary
- SymbolicIOField
- SysDiagControl
- TextField
- TrendView
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- UserView
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- AlarmView
- PLCCodeViewer
- ProtectedAreaNameView
- RangeLabelView
- S7GraphOverview

HMIRTAuthorization

Optional. Ein Wert oder eine Konstante, die die Bedienrechte des angegebenen Objekts in Runtime festlegt.

AutoCompleteColumns (RT Professional)

Beschreibung

Legt fest, ob Leerspalten angezeigt werden, wenn das Control breiter ist als die projizierten Spalten.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AutoCompleteColumns**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl

- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Leerspalten werden angezeigt.

FALSE: Die Leerspalten werden nicht angezeigt.

AutoCompleteRows (RT Professional)

Beschreibung

Legt fest, ob Leerzeilen angezeigt werden, wenn das Control länger ist als die Anzahl der projektierten Zeilen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AutoCompleteRows**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Leerzeilen werden angezeigt.

FALSE: Die Leerzeilen werden nicht angezeigt.

AutoScroll (RT Professional)

Beschreibung

Legt fest, wie sich das Meldefenster beim Auftreten neuer Meldungen verhält.

Die gezielte Auswahl von Meldezeilen ist nur möglich, wenn "AutoScroll" nicht aktiviert ist.

Die Eigenschaft "AutoScroll" wird deaktiviert, wenn das Attribut "MsgCtrlFlag" = "-1" gesetzt wird. Die neueste Meldung wird im Meldefenster oben angezeigt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AutoScroll**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

Optional.

TRUE: Eine neu auftretende Meldung wird an die im Meldefenster dargestellte Liste angehängt und ausgewählt. Der sichtbare Bereich des Meldefensters wird bei Bedarf verschoben.

FALSE: Eine neu auftretende Meldung wird nicht ausgewählt. Der sichtbare Bereich des Meldefensters wird nicht verändert.

AutoSelectionColors (RT Professional)

Beschreibung

Legt fest, ob die Markierungsfarben für Zellen und Zeilen mit den vom System festgelegten Farben dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AutoSelectionColors**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Systemfarbe wird verwendet.

FALSE: Die benutzerdefinierte Farbe wird verwendet.

AutoSelectionRectColor (RT Professional)

Beschreibung

Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AutoSelectionRectColors**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Systemfarbe wird verwendet.

FALSE: Die benutzerdefinierte Farbe wird verwendet.

AutoSizeing (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

AverageLast15Values (RT Professional)

Beschreibung

Legt fest, ob ein Mittelwert aus den letzten 15 Werten gebildet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AverageLast15Values**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn ein Mittelwert aus den letzten 15 Werten gebildet wird.

AxisXBunchCount (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisXMarkCount (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisXShowBunchValues (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisY1BunchCount (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisY1MarkCount (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisY1ShowBunchValues (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisY2BunchCount (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisY2MarkCount (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

AxisY2ShowBunchValues (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

3.5.5.2 Eigenschaften B (Panels, Comfort Panels, RT Advanced, RT Professional)

BackColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Hintergrundfarbe des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- AlarmView
- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- CircularArc
- ComboBox
- DateTimeField
- Ellipse
- EllipseSegment
- EllipticalArc

- FunctionTrendControl
- Gauge
- GraphicView
- IOField
- Line
- ListBox
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- Polygon
- Polyline
- RecipeView
- Rectangle
- RoundButton
- Slider
- StatusForce
- Switch
- SymbolLibrary
- SymbolicIOField
- TextField
- TrendRulerControl
- TrendView
- TubeArcObject
- UserArchiveControl
- UserView
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- GraphicIOField

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Die Hintergrundfarbe ist nicht sichtbar, wenn die Eigenschaft "BorderStyle" den Wert "0" hat.

BackColorBottom (RT Professional)

Beschreibung

Legt die Farbe für den unteren/rechten Teil des Objekts Slider fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BackColorBottom**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- WindowsSlider

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den unteren/rechten Teil des Objekts Slider festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0-255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BackColorTop (RT Professional)

Beschreibung

Legt die Farbe für den oberen/linken Teil des Objekts Slider fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BackColorTop**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- WindowsSlider

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den oberen/linken Teil des Objekts Slider festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0-255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BackFillStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt das Füllmuster des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BackFillStyle**[=FillStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- Clock
- ComboBox
- Ellipse
- EllipseSegment
- Gauge
- GraphicView
- IOField

- ListBox
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- Slider
- SymbolLibrary
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- DateTimeField
- GraphicIOField
- Switch

FillStyle

Optional. Ein Wert oder eine Konstante, die das Füllmuster festlegt.

hmiFillStyleTransparent (65536): Transparente Füllung

hmiFillStyleSolid (0): Objekt wird mit der angegebenen Farbe gefüllt

Standardeinstellung: hmiFillStyleSolid

BackFlashingColorOff (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Hintergrundes für den Blinkzustand "Aus" fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BackFlashingColorOff**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- GraphicView

- IOField
- OptionGroup
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Hintergrundes für den Blinkzustand "Aus" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

BackFlashingColorOn (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Hintergrundes für den Blinkzustand "Ein" fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BackFlashingColorOn**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- GraphicView
- IOField

- OptionGroup
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Hintergrundes für den Blinkzustand "Ein" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

BackFlashingEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Hintergrund des angegebenen Objekts in Runtime blinkt.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BackFlashingEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- GraphicView
- IOField
- OptionGroup

- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch

BOOLEAN

Optional. TRUE, wenn der Hintergrund des angegebenen Objekts in Runtime blinkt.

BackFlashingRate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Blinkfrequenz des Hintergrundes des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BackFlashingRate**[=FlashingRate]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- GraphicView
- IOField
- OptionGroup
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch

FlashingRate

hmiFlashingRateSlow (0): Die Länge des Blinkintervalls ist 250 ms.

hmiFlashingRateMedium (1): Die Länge des Blinkintervalls ist 500 ms.

hmiFlashingRateFast (2): Die Länge des Blinkintervalls ist 1000 ms.

BackgroundColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BarBackColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Balkenhintergrunds im angegebenen Objekt fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BarBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- Slider

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Balkenhintergrunds festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

BarBackFillStyle (RT Professional)

Beschreibung

Legt das Füllmuster des Balkens fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BarBackFillStyle**[=FillStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

FillStyle

hmiFillStyleTransparent (65536): Transparentes Füllmuster

hmiFillStyleSolid (0): Solides Füllmuster

hmiFillStyleBackwardDiagonal (131075): Diagonal, nach rechts oben gestreiftes Füllmuster

hmiFillStyleCross (131076): Kariertes Füllmuster

hmiFillStyleDiagonalCross (131077): Diagonal kariertes Füllmuster

hmiFillStyleForwardDiagonal (131074): Diagonal, nach links oben gestreiftes Füllmuster

hmiFillStyleHorizontal (131072): Horizontal gestreiftes Füllmuster

hmiFillStyleVertical (131073): Vertikal gestreiftes Füllmuster

hmiFillStylePattern1 (196608): Vordefiniertes Füllmuster

hmiFillStylePattern2 (196609): Vordefiniertes Füllmuster

hmiFillStylePattern3 (196610): Vordefiniertes Füllmuster

hmiFillStylePattern4 (196611): Vordefiniertes Füllmuster

hmiFillStylePattern5 (196612): Vordefiniertes Füllmuster

hmiFillStylePattern6 (196613): Vordefiniertes Füllmuster

hmiFillStylePattern7 (196614): Vordefiniertes Füllmuster

hmiFillStylePattern8 (196615): Vordefiniertes Füllmuster

hmiFillStylePattern9 (196616): Vordefiniertes Füllmuster

hmiFillStylePattern10 (196617): Vordefiniertes Füllmuster

hmiFillStylePattern11 (196618): Vordefiniertes Füllmuster

hmiFillStylePattern12 (196619): Vordefiniertes Füllmuster

hmiFillStylePattern13 (196620): Vordefiniertes Füllmuster

hmiFillStylePattern14 (196621): Vordefiniertes Füllmuster

hmiFillStylePattern15 (196622): Vordefiniertes Füllmuster

hmiFillStylePattern16 (196623): Vordefiniertes Füllmuster

hmiFillStylePattern17 (196624): Vordefiniertes Füllmuster

hmiFillStylePattern18 (196625): Vordefiniertes Füllmuster

hmiFillStylePattern19 (196626): Vordefiniertes Füllmuster

hmiFillStylePattern20 (196627): Vordefiniertes Füllmuster
hmiFillStylePattern21 (196628): Vordefiniertes Füllmuster
hmiFillStylePattern22 (196629): Vordefiniertes Füllmuster
hmiFillStylePattern23 (196630): Vordefiniertes Füllmuster
hmiFillStylePattern24 (196631): Vordefiniertes Füllmuster
hmiFillStylePattern25 (196632): Vordefiniertes Füllmuster
hmiFillStylePattern26 (196633): Vordefiniertes Füllmuster
hmiFillStylePattern27 (196634): Vordefiniertes Füllmuster
hmiFillStylePattern28 (196635): Vordefiniertes Füllmuster
hmiFillStylePattern29 (196636): Vordefiniertes Füllmuster
hmiFillStylePattern30 (196637): Vordefiniertes Füllmuster
hmiFillStylePattern31 (196638): Vordefiniertes Füllmuster
hmiFillStylePattern32 (196639): Vordefiniertes Füllmuster
hmiFillStylePattern33 (196640): Vordefiniertes Füllmuster
hmiFillStylePattern34 (196641): Vordefiniertes Füllmuster
hmiFillStylePattern35 (196642): Vordefiniertes Füllmuster
hmiFillStylePattern36 (196643): Vordefiniertes Füllmuster
hmiFillStylePattern37 (196644): Vordefiniertes Füllmuster

BarBackFlashingColorOff (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BarBackFlashingColorOn (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BarBackFlashingEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BarBackFlashingRate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BarColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Balkens im Objekt "Slider" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BarColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Slider

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Reglers festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

Der Bereich erstreckt sich vom "RangeMin" bis zur Position des Reglers.

BarEdgeStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BaseScreenName (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Liest den Namen des aktuellen Grundbildes oder löst über das Setzen eines neuen Bildnamens einen Grundbildwechsel aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BaseScreenName**[= STRING]

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

STRING

Optional. Ein Wert oder eine Konstante, die den Bildnamen enthält.

Bemerkungen

Sie können die Eigenschaft auch verwenden, um festzustellen, welches Bild gerade angezeigt wird.

BelowLowerLimitColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BitNumber (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt das Bit fest, dessen Zustand sich ändern muss, um eine Wertänderung auszulösen.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BitNumber**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- SymbolicOField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Button
- GraphicOField

Int32

Optional. Ein Wert oder eine Konstante, welche das Bit festlegt, dessen Zustand sich ändern muss, um eine Wertänderung auszulösen.

Bemerkungen

Die verwendete Variable muss vom Typ BYTE, WORD oder DWORD sein.

BlinkColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe fest, mit der das Objekt "SymbolLibrary" blinkt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

```
Object.BlinkColor[=Color]
```

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

Color

Optional. Ein Wert oder eine Konstante, welche die Blinkfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

BlinkMode (RT Professional)

Beschreibung

Legt die Art der Blinkgrafik für das angegebene Objekt fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BlinkMode**[=SymbolLibraryBlinkMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

SymbolLibraryBlinkMode

hmiSymbolLibraryFlashingNone (0): Das Blinken ist ausgeschaltet.

hmiSymbolLibraryFlashingInvisible (1): Die Blinkgrafik ist unsichtbar.

hmiSymbolLibraryFlashingShaded (2): Die Blinkgrafik erhält eine farbige, schattierte Oberfläche. Die Farbe der Oberfläche entspricht der Einstellung der Eigenschaft "BlinkColor".

hmiSymbolLibraryFlashingSolid (3): Die Blinkgrafik erhält eine farbige, unshattierte Oberfläche. Die Farbe der Oberfläche entspricht der Einstellung der Eigenschaft "BlinkColor".

BlinkSpeed (RT Professional)

Beschreibung

Legt die Blinkfrequenz fest.

Fast - 250: Die Länge des Blinkintervalls ist 250 ms. Medium - 500: Die Länge des Blinkintervalls ist 500 ms.

Slow - 1000: Die Länge des Blinkintervalls ist 1000 ms. Der Standardwert ist Medium - 500.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BlinkSpeed**[=FlashingRate]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

FlashingRate

hmiFlashingRateSlow (0): Die Länge des Blinkintervalls ist 250 ms.

hmiFlashingRateMedium (1): Die Länge des Blinkintervalls ist 500 ms.

hmiFlashingRateFast (2): Die Länge des Blinkintervalls ist 1000 ms.

Blocks (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BorderColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BorderColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- ComboBox
- Ellipse
- EllipseSegment
- GraphicIOField
- GraphicView
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- Polygon

- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- AlarmView
- Clock
- DateTimeField
- Gauge
- RecipeView
- Slider
- StatusForce
- Switch
- SysDiagControl
- TrendView
- UserView

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der durchbrochenen Rahmenlinie des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BorderBrightColor3D (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest:

- Äußere Rahmenteile oben und unten
- Innere Rahmenteile unten und rechts

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderBrightColor3D**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- RoundButton
- Slider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Rahmenfarbe festlegt. Standardeinstellung ist weiß.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BorderColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Linienfarbe des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- AlarmView
- Bar
- Button
- CheckBox
- Circle

- CircleSegment
- ComboBox
- DateTimeField
- Ellipse
- EllipseSegment
- FunctionTrendControl
- GraphicIOField
- GraphicView
- IOField
- ListBox
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- TrendRulerControl
- UserArchiveControl
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Clock
- Gauge
- RecipeView
- Slider
- StatusForce
- Switch
- SysDiagControl
- TrendView
- UserView

Color

Optional. Ein Wert oder eine Konstante, welche die Linienfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BorderEnabled (RT Professional)

Beschreibung

Gibt zurück, ob das Fenster in Runtime mit einem Rahmen dargestellt wird.

Zugriff in Runtime: Lesen

Syntax

Object.**BorderEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ApplicationWindow
- Screenwindow

BOOLEAN

Optional. TRUE, wenn das Fenster in Runtime mit einem Rahmen dargestellt wird.

BorderEndStyle (RT Professional)

Beschreibung

Legt die Art der Linienenden des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderEndStyle**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Line
- Polyline

Int32

Optional. Ein Wert oder eine Konstante, welche die Art der Linienenden des angegebenen Objekts festlegt.

BorderFlashingColorOff (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BorderFlashingColorOff**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- Circle
- CircleSegment
- Ellipse
- EllipseSegment
- GraphicIOField
- GraphicView
- IOField
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BorderFlashingColorOn (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BorderFlashingColorOn**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- Circle
- CircleSegment
- Ellipse
- EllipseSegment
- GraphicIOField
- GraphicView
- IOField
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- SymbolicIOField

- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BorderFlashingEnabled (RT Professional)

Beschreibung

Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderFlashingEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "Ellipse", "Circle", "EllipseSegment", "CircleSegment", "Rectangle", "Polygon", "TextField", "IOField", "SymbolicIOField", "Button", "Roundbutton", "Switch", "GraphicView", "GraphicIOField", "Bar", "Checkbox", "OptionGroup" oder "WindowSlider".

BOOLEAN

Optional. TRUE, wenn die Rahmenlinie des Objekts blinkt.

Siehe auch

Ellipse (Seite 1386)

Circle (Seite 1369)

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

- Rectangle (Seite 1471)
- Polygon (Seite 1456)
- TextField (Seite 1505)
- IOField (Seite 1418)
- SymbolicIOField (Seite 1491)
- Button (Seite 1360)
- Switch (Seite 1488)
- GraphicView (Seite 1412)
- GraphicIOField (Seite 1409)
- Bar (Seite 1353)
- CheckBox (Seite 1366)
- OptionGroup (Seite 1452)
- WindowSlider (Seite 1538)
- RoundButton (Seite 1473)

BorderFlashingRate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen/Schreiben

Syntax

Object.**BorderFlashingRate**[=FlashingRate]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- Circle
- CircleSegment
- Ellipse
- EllipseSegment
- GraphicIOField
- GraphicView

- IOField
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch

FlashingRate

hmiFlashingRateSlow (0): Die Länge des Blinkintervalls ist 250 ms.

hmiFlashingRateMedium (1): Die Länge des Blinkintervalls ist 500 ms.

hmiFlashingRateFast (2): Die Länge des Blinkintervalls ist 1000 ms.

BorderInnerStyle3D (RT Professional)

Beschreibung

Legt die Darstellung des inneren Teils des Objektrahmens fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderInnerStyle3D**[=SliderBorder3DStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge
- Slider

SliderBorder3DStyle

hmiGaugeBorder3DStyleNone (0): Es gibt keinen inneren Teil des Objektrahmens.

hmiGaugeBorder3DStyleRecessed (1): Der Objektrahmen ist vertieft dargestellt.

hmiGaugeBorder3DStyleRaised (2): Der Objektrahmen ist erhaben dargestellt.

hmiGaugeBorder3DStyleGray (3): Der Objektrahmen hat einen einheitlich grauen Rahmen.

hmiGaugeBorder3DStyleColored (4): Der Objektrahmen hat einen einheitlich farbigen Rahmen. Die Rahmenfarbe entspricht der Hintergrundfarbe.

BorderInnerWidth3D (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Breite des Innenrahmens für die 3D-Darstellung des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderInnerWidth3D**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des Innenrahmens in Pixeln festlegt.

BorderOuterStyle3D (RT Professional)

Beschreibung

Legt die Darstellung des äußeren Teils des Objektrahmens fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderOuterStyle3D**[=GaugeBorder3DStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge
- Slider

GaugeBorder3DStyle

hmiGaugeBorder3DStyleNone (0): Es gibt keinen inneren Teil des Objektrahmens.

hmiGaugeBorder3DStyleRecessed (1): Der Objektrahmen ist vertieft dargestellt.

hmiGaugeBorder3DStyleRaised (2): Der Objektrahmen ist erhaben dargestellt.

hmiGaugeBorder3DStyleGray (3): Der Objektrahmen hat einen einheitlich grauen Rahmen.

hmiGaugeBorder3DStyleColored (4): Der Objektrahmen hat einen einheitlich farbigen Rahmen. Die Rahmenfarbe entspricht der Hintergrundfarbe.

BorderOuterWidth3D (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Breite des Außenrahmens für die 3D-Darstellung des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderOuterWidth3D**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des Außenrahmens in Pixeln festlegt.

BorderShadeColor3D (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Rahmenfarbe folgender Rahmenteile bei 3D-Darstellung des angegebenen Objekts fest:

- Innere Rahmenteile oben und unten
- Äußere Rahmenteile unten und rechts

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderShadeColor3D**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- RoundButton
- Slider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für die Schattierung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

BorderStyle (RT Professional)

Beschreibung

Legt die Art der Rahmenlinien des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderStyle**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- Button
- CheckBox
- ComboBox
- GraphicIOField
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- RoundButton
- SymbolicIOField
- WindowsSlider

Int32

Optional. Ein Wert oder eine Konstante, welche die Art der Rahmenlinien des angegebenen Objekts festlegt.

0 = durchgezogene Linie

1 = gestrichelte Linie

2 = gepunktete Linie

3 = strichpunktierte Linie

4 = Strich-Punkt-Punkt-Linie

BorderStyle3D (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob das angegebene Objekt eine 3D-Rahmenschattierung besitzt.

Zugriff in Runtime:

- RT Advanced: Lesen und Schreiben
- RT Professional: kein Zugriff

Syntax

Object.**BorderStyle3D**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- DateTimeField
- IOField
- Switch
- TextField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Bar
- Button
- GraphicIOField
- GraphicView
- SymbolicIOField

BOOLEAN

Optional. TRUE, wenn das Objekt eine 3D-Rahmenschattierung besitzt.

BorderWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Linienstärke des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BorderWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- AlarmView
- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- ComboBox
- Ellipse
- EllipseSegment
- FunctionTrendControl
- Gauge
- GraphicIOField
- GraphicView
- IOField
- ListBox
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- Slider
- SymbolicIOField
- TextField
- TrendRulerControl
- UserArchiveControl
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Clock
- DateTimeField
- RecipeView
- StatusForce
- Switch
- SysDiagControl
- TrendView
- UserView

Int32

Optional. Ein Wert oder eine Konstante, welche die Linienstärke in Pixeln festlegt.

BorderWidth3D (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Breite des Innenrahmens für die 3D-Darstellung des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**BorderWidth3D**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- Gauge
- RoundButton

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Bar
- IOField
- Switch

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des Innenrahmens für die 3D-Darstellung in Pixel festlegt.

BottomMargin (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BusyText (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonBackColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonBackFillStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonBarElements (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonBarStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonBorderBackColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonBorderColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonBorderWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonCornerRadius (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonEdgeStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonFirstGradientColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonFirstGradientOffset (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonMiddleGradientColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonSecondGradientColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ButtonSecondGradientOffset (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ColumnWidth_Date (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ColumnWidth_Event (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ColumnWidth_EventSeverity (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ColumnWidth_EventState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ColumnWidth_Number (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ColumnWidth_Time (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ItemText_Date (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ItemText_Event (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ItemText_EventSeverity (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ItemText_EventState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ItemText_Number (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ItemText_Time (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ShowItem_Date (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ShowItem_Event (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ShowItem_EventSeverity (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ShowItem_EventState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ShowItem_Number (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

BV_ShowItem_Time (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

3.5.5.3 **Eigenschaften C (Panels, Comfort Panels, RT Advanced, RT Professional)**

Caption (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Caption**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- Slider
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, die den Text enthält, der in der Titelzeile dargestellt wird.

CaptionBackColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe der Titelzeile des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CaptionBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicIOField

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der Titelzeile des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0-255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

CaptionColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Texts fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CaptionColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Gauge
- Switch
- SymbolicIOField

Color

Optional. Ein Wert oder eine Konstante, welche die Textfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

CaptionText (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Text fest, der in der Titelzeile des angegebenen Objekts angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CaptionText**[= STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Gauge
- ScreenWindow
- Switch

STRING

Optional. Ein Wert oder eine Konstante, die den Text enthält, der in der Titelzeile dargestellt wird.

CaptionTop (RT Professional)

Beschreibung

Legt den Abstand der Instrumentenbeschriftung zur Oberkante des angegebenen Objekts fest. Sie können die Instrumentenbeschriftung nur längs des vertikalen Durchmessers der Skalenscheibe positionieren. Der Wert des Attributs bezieht sich auf die Höhe des angegebenen Objekts. Die Höhe legt die Oberkante des angegebenen Objekts und die Unterkante des Schriftzugs fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CaptionTop**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den Abstand der Instrumentenbeschriftung zur Oberkante des angegebenen Objekts festlegt.

Wertebereich 0 bis 1

0: Die Unterkante des Schriftzugs liegt auf der oberen Begrenzung des angegebenen Objekts. Der Text ist nicht mehr sichtbar, da er außerhalb des angegebenen Objekts liegt.

1: Die Unterkante des Schriftzugs liegt auf der unteren Begrenzung des angegebenen Objekts.

CellCut (RT Professional)

Beschreibung

Legt fest, ob die Inhalte der Zellen abgekürzt werden, wenn die Zellen zu schmal sind.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CellCut**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE, wenn die Inhalte abgekürzt werden.

CellSpaceBottom (RT Professional)

Beschreibung

Legt den Abstand von unten fest, der in den Zellen der Tabelle verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CellSpaceBottom**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgenden Ausprägungen:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchivControl

Int32

Optional. Ein Wert oder Konstante, welche den Abstand von unten festlegt, der in den Zellen der Tabelle verwendet wird.

CellSpaceLeft (RT Professional)

Beschreibung

Legt den Einzug von links fest, der in den Zellen der Tabelle verwendet wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CellSpaceLeft**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder ein Parameter, der den Einzug von links innerhalb der Tabellenzellen festlegt.

CellSpaceRight (RT Professional)

Beschreibung

Legt den Einzug von rechts fest, der in den Zellen der Tabelle verwendet wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CellSpaceRight**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgenden Ausprägungen:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder Konstante, welche den Einzug von rechts festlegt, der in den Zellen der Tabelle verwendet wird.

CellSpaceTop (RT Professional)

Beschreibung

Legt den Abstand von oben fest, der in den Zellen der Tabelle verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CellSpaceTop**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgenden Ausprägungen:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder Konstante, welche den Abstand von oben festlegt, der in den Zellen der Tabelle verwendet wird.

CenterColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Mittelpunktes für das Objekt "Gauge" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CenterColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Mittelpunktes festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

CenterSize (RT Professional)

Beschreibung

Legt den Durchmesser des Skalenmittelpunkts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CenterSize**[=SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

SINGLE

Optional. Ein Wert oder eine Konstante, welche den Durchmesser des runden Skalenmittelpunkts festlegt.

Wertebereich ist 0,03 bis 1

1: Der Durchmesser entspricht dem kleineren Wert der Geometrieattribute "Width" oder "Height".

ChangeMouseCursor (RT Professional)

Beschreibung

Legt fest, ob das Aussehen des Mauszeigers in Runtime gleich bleibt, wenn er sich über dem Symbol befindet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ChangeMouseCursor**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

BOOLEAN

Optional. TRUE, wenn der Mauszeiger wie ein Pfeil aussieht, auch wenn er über dem Symbol positioniert wird.

FALSE, wenn der Mauszeiger wie ein 3D-Pfeil mit einem grünen Blitzsymbol aussieht. Damit wird in Runtime signalisiert, dass das betreffende Objekt bedienbar ist.

CheckMarkAlignment (RT Professional)

Beschreibung

Legt fest, ob die Felder rechtsbündig angeordnet sind.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CheckMarkAlignment**[=ContentAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- OptionGroup

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox

ContentAlignment

(0): Die Felder sind linksbündig angeordnet.

(1): Die Felder sind rechtsbündig angeordnet.

CheckMarkCount (RT Professional)

Beschreibung

Legt die Anzahl der Felder fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CheckMarkCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- OptionGroup

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Felder festlegt. Wertebereich von 0 bis 31

ClearOnError (RT Professional)

Beschreibung

Legt fest, ob eine Fehleingabe in diesem Objekt automatisch gelöscht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ClearOnError**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

BOOLEAN

Optional. TRUE, wenn eine Fehleingabe in diesem Objekt automatisch gelöscht wird.

ClearOnFocus (RT Professional)

Beschreibung

Legt fest, ob der Feldeintrag gelöscht wird, sobald das E/A-Feld aktiviert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ClearOnFocus**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

BOOLEAN

Optional. TRUE, wenn der Feldeintrag gelöscht wird, sobald das E/A-Feld aktiviert wird.

Closeable (RT Professional)

Beschreibung

Legt fest, ob das Fenster in Runtime geschlossen werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Closeable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn das Fenster in Runtime geschlossen werden kann.

Siehe auch

AlarmControl (Seite 1338)

UserArchiveControl (Seite 1528)

SystemDiagnoseWindow (Seite 1501)

TrendRulerControl (Seite 1508)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

OnlineTrendControl (Seite 1443)

Color (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Linienfarbe des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Color**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- CircularArc
- EllipticalArc

- Line
- Polyline
- TubeArcObject
- TubeDoubleTeeObject
- Tubepolyline
- TubeTeeObject

Color

Optional. Ein Wert oder eine Konstante, welche die Linienfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

ColorChangeHysteresis (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Hysterese prozentual zum Anzeigewert fest.

Die Eigenschaft "ColorChangeHysteresisEnable" muss den Wert TRUE haben, damit die Hysterese berechnet werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColorChangeHysteresis**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die Hysterese prozentual zum Anzeigewert festlegt.

ColorChangeHysteresisEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob das Objekt mit Hysterese angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColorChangeHysteresisEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn das Objekt mit Hysterese angezeigt wird.

ColumnAlignment (RT Professional)

Beschreibung

Legt fest, wie die ausgewählte Spalte ausgerichtet wird.

Folgende Einstellungen sind möglich:

Wert	Beschreibung	Erklärung
0	links	Die ausgewählte Spalte wird linksbündig angezeigt.
1	zentriert	Die ausgewählte Spalte wird zentriert angezeigt.
2	rechts	Die ausgewählte Spalte wird rechtsbündig angezeigt.

Das Attribut ist mit dem Namen **ColumnAlignment** dynamisierbar. Der Datentyp ist LONG.

ColumnDateFormat (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, welches Datumsformat zur Anzeige verwendet wird. Über die Eigenschaft "ColumnIndex" wird die Datumsspalte der Rezepturanzeige referenziert. Das Attribut ist mit dem Namen **ColumnDateFormat** dynamisierbar.

Zugriff in Runtime: Schreiben

Syntax

Object.**ColumnDateFormat**(=String)

Object

Erforderlich. Ein Object vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

String

Folgende Datumsformate stehen zur Verfügung:

Wert	Erklärung
dd.MM.yy	Tag.Monat.Jahr, z. B. 24.12.13.
dd.yyyd.MM	Tag.Monat.Jahr, z. B. 24.12.2013.
dd/MM/yy	Tag/Monat/Jahr, z. B. 24/12/13
dd/MM/yyyy	Tag/Monat/Jahr, z. B. 24/12/2013.

Siehe auch

UserArchiveControl (Seite 1528)

ColumnResize (RT Professional)

Beschreibung

Legt fest, ob Sie die Breite der Spalten ändern können.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColumnResize**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Sie können die Breite der Spalten ändern.

FALSE: Sie können die Breite der Spalten nicht ändern

Columns (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnScrollbar (RT Professional)

Beschreibung

Legt fest, ob Spalten-Rollbalken angezeigt werden.

Zugriff in Runtime: Lesen

Syntax

Object.**ColumnScrollbar**[=Scrollbar]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "AlarmControl", "TrendRulerControl", "OnlineTableControl", "UserArchivControl".

Scrollbar

0 = (Nein) Die Spalten-Rollbalken werden nicht angezeigt

1 = (bei Bedarf) Die Spalten-Rollbalken werden angezeigt, wenn der Platzbedarf des Controls in vertikaler Richtung größer ist als der zur Verfügung stehende Anzeigebereich.

2 = (immer) Die Spalten-Rollbalken werden immer angezeigt

Siehe auch

UserArchiveControl (Seite 1528)

TrendRulerControl (Seite 1508)

OnlineTableControl (Seite 1434)

AlarmControl (Seite 1338)

ColumnSettings (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnSettingsBufferView (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnsMoveable (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextAckGroup (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextAlarmState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextAlarmText (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextClassName (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextDate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextDevice (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextDiagnosable (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextNumber (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTextTime (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ColumnTimeFormat (RT Professional)

Beschreibung

Legt fest, welches Zeitformat zur Anzeige verwendet wird. Über die Eigenschaft "ColumnIndex" wird die Zeitspalte der Rezepturanzeige referenziert. Das Attribut ist mit dem Namen **ColumnTimeFormat** dynamisierbar.

Zugriff in Runtime: Schreiben

Syntax

Object.**ColumnTimeFormat**(=String)

Object

Erforderlich. Ein Object vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

String

Folgende Zeitformate stehen zur Verfügung:

Wert	Erklärung
HH:mm:ss.ms	Stunden:Minuten:Sekunden, z.B. 15:35:44.240.
hh:mm:ss tt	Stunden:Minuten:Sekunden AM/PM, z.B. 03:35:44 PM.
hh:mm:ss.ms tt	Stunden:Minuten:Sekunden.Millisekunden AM/PM, z.B. 03:35:44.240 PM.

ColumnTitleAlignment (RT Professional)

Beschreibung

Legt fest, wie die Spaltenüberschriften ausgerichtet werden.

Zugriff in Runtime: Lesen

Syntax

Object.**ColumnTitleAlignment**[=GridColumnHeaderHorizontalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "AlarmControl", "TrendRulerControl", "OnlineTableControl", "UserArchivControl".

GridColumnHeaderHorizontalAlignment

0 = Die Spaltenüberschriften werden linksbündig angezeigt.

1 = Die Spaltenüberschriften werden zentriert angezeigt

2 = Die Spaltenüberschriften werden rechtsbündig angezeigt.

3 = Die Spaltenüberschriften werden wie der zugehörige Inhalt der Spalte ausgerichtet.

Siehe auch

AlarmControl (Seite 1338)

ColumnTitles (RT Professional)

Beschreibung

Legt fest, ob die Spaltenüberschrift angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColumnTitles**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchivControl

BOOLEAN

TRUE: Die Spaltenüberschrift wird angezeigt.

FALSE: Die Spaltenüberschrift wird nicht angezeigt.

ComboBoxFont (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

Dublette_CompatibilityMode (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ComputerName (RT Professional)

Beschreibung

Gibt den Namen des Computers zurück, auf dem das Alarm-Objekt ausgelöst wurde.

ComputerName (readonly)

ConnectionType (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime

ConnectOnStart (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ConnectTrendWindows (RT Professional)

Beschreibung

Legt fest, ob die projizierten Kurvenanzeigen verbunden werden. Voraussetzung ist, dass Sie mehrere Kurvenanzeigen projiziert haben.

Die verbundenen Kurvenanzeigen haben folgende Eigenschaften:

- Eine gemeinsame X-Achse
- Einen Rollbalken
- Ein Lineal

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ConnectTrendWindows**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Alle projizierten Kurvenanzeigen werden verbunden.

FALSE: Die Kurvenanzeige werden separat dargestellt.

Context (RT Professional)

Beschreibung

Liest oder setzt das Serverpräfix des Alarm-Objekts.

ContinousChange (RT Professional)

Beschreibung

Legt fest, ob der Wert der Eigenschaft "ProcessValue" beim Loslassen der Maustaste oder sofort beim Ändern der Schieberposition in Runtime übergeben wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ContinousChange**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

BOOLEAN

Optional. TRUE, wenn der Wert der Eigenschaft "ProcessValue" beim Loslassen der Maustaste oder sofort bei Änderung der Schieberposition in Runtime übergeben wird.

CornerRadius (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

CornerStyle (RT Professional)

Beschreibung

Legt die Art der Rahmenlinien des angegebenen Objekts fest.

Zugriff in Runtime: Nur Lesen

Syntax

Object.**CornerStyle**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- CircularArc
- ComboBox
- EllipseSegment
- GraphicIOField
- GraphicView
- IOField
- Line
- ListBox

- MultiLineEdit
- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- Tubepolyline
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Ellipse
- EllipticalArc
- Switch
- TubeArcObject

Int

Optional. Ein Wert oder eine Konstante, welche die Art der Rahmenlinien des angegebenen Objekts festlegt.

0 = durchgezogene Linie

1 = gestrichelte Linie

2 = gepunktete Linie

3 = strichpunktierte Linie

4 = Strich-Punkt-Punkt-Linie

Count (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt die Anzahl der Elemente in der angegebenen Auflistung zurück
Zugriff in Runtime: Lesen

Syntax

Object.Count

Object

Erforderlich. Ein Objekt vom Typ "Collection".

CountDivisions (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Anzahl der Segmente fest, in die der Balken durch die großen Teilstriche der Skala unterteilt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.CountDivisions[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Segmente festlegt, in die der Balken durch die großen Teilstriche der Skala unterteilt wird.

0-100: Objekt kann maximal in 100 Segmente unterteilt werden

= 0: Die optimale Anzahl der Segmente wird automatisch festgelegt

Siehe auch

Bar (Seite 1353)

CountOfLinesPerAlarms (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

CountOfVisibleAlarms (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

CountSubDivisions (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

CountVisibleItems (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, wieviele Zeilen die Auswahlliste enthält. Wenn die Anzahl der projektorientierten Texte größer als dieser Wert ist, dann hat die Auswahlliste eine vertikale Bildlaufleiste.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**CountVisibleItems**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- ComboBox
- ListBox
- SymbolicIOField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- StatusForce
- TrendView
- UserView

Int32

Optional. Ein Wert oder eine Konstante, welche festlegt, wieviele Zeilen die Auswahlliste enthält.

CurrentContext (RT Professional)

Beschreibung

Gibt eine Zeichenkette abhängig von der Verwendung der Funktion zurück.

Wenn die Funktion in einem Bild im Bildfenster enthalten ist, gibt CurrentContext den symbolischen Servernamen zurück, der das Bild liefert. Beispiel:

"WinCCProject_MyComputer::"

Wenn die Funktion im Grundbild enthalten ist, wird eine leere Zeichenkette zurückgegeben.

Zugriff in Runtime: Lesen

Syntax

Object.**CurrentContext**

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

Siehe auch

HMIRuntime (Seite 1313)

CursorControl (RT Professional)

Beschreibung

Legt fest, ob der Mauszeiger nach dem Verlassen des Feldes auf das nächste Feld der TAB-Reihenfolge springt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CursorControl**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField
- SymbolicIOField

BOOLEAN

Optional. TRUE, wenn der Mauszeiger nach dem Verlassen des Feldes auf das nächste Feld der TAB-Reihenfolge springt.

Bemerkungen

Die Eigenschaft "CursorMode" muss dazu auf TRUE gesetzt sein.

Curves (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

3.5.5.4 Eigenschaften D (Panels, Comfort Panels, RT Advanced, RT Professional)

DangerRangeColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Gefahrenbereichs auf der Skala des Objekts "Gauge" fest.
Die Eigenschaft "DangerRangeVisible" muss den Wert TRUE haben, damit der Gefahrenbereich angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DangerRangeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Gefahrenbereichs festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

DangerRangeStart (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ab welchem Skalenwert der Gefahrenbereich des Objekts "Gauge" beginnt.

Die Eigenschaft "DangerRangeColor" muss den Wert TRUE haben, damit der Gefahrenbereich angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DangerRangeStart**[=SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

SINGLE

Optional. Ein Wert oder eine Konstante, welche den Skalenwert für den Beginn des Gefahrenbereichs enthält.

Bemerkungen

Der Bereich erstreckt sich ab dem Wert "Gefahr" bis zum Ende der Skala.

DangerRangeVisible (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Gefahrenbereich in der Skala des Objekts "Gauge" angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DangerRangeVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

BOOLEAN

Optional. TRUE, wenn der Gefahrenbereich in der Skala angezeigt wird.

Bemerkungen

Die Farbe des Gefahrenbereichs legen Sie mit der Eigenschaft "DangerRangeColor" fest.

Den Beginn des Gefahrenbereichs legen Sie mit der Eigenschaft "DangerRangeStart" fest.

DataFormat (RT Professional)

Beschreibung

Gibt den Datentyp des Objekts IOField zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**DataFormat**[=IOFieldDataFormat]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

IOFieldDataFormat

Optional. Ein Wert oder eine Konstante, welche den Datentyp des Objekts IOField zurückgibt.

Wertebereich von 0 bis 3.

- 0: Binär
- 1: Dezimal
- 2: String
- 3: Hexadezimal

DataLogs (RT Professional)

Beschreibung

Gibt ein Objekt vom Typ "DataLogs" zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**DataLogs**

Object

Erforderlich. Ein Objekt vom Typ "Logging".

Siehe auch

Logging (Seite 1318)

DataRecordNameCaption (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DataRecordNrCaption (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DataSet (RT Professional)

Beschreibung

Gibt ein Objekt vom Typ "DataSet" zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**DataSet**

Object

Erforderlich. Ein Objekt vom Typ "Screen".

Siehe auch

HMIRuntime (Seite 1313)

Screen (Seite 1320)

DialColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Zifferblattes im angegebenen Objekt fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DialColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock
- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Zifferblattes festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

DialFillStyle (RT Professional)

Beschreibung

Legt die Art des Hintergrundes des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DialFillStyle**[=GaugeFillStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

GaugeFillStyle

hmiGaugeBackStyleSolid (0): Der rechteckige Hintergrund der Anzeige ist mit der angegebenen Rahmenfarbe gefüllt. Die Skalenscheibe ist mit der angegebenen Hintergrundfarbe gefüllt.

hmiGaugeBackStyleFrameTransparent (1): Der rechteckige Hintergrund der Gauge ist transparent. Die Skalenscheibe ist mit der angegebenen Hintergrundfarbe gefüllt. Damit ist eine kreisförmige Anzeige darstellbar.

hmiGaugeBackStyleTransparent (2): Der rechteckige Hintergrund und die Skalenscheibe sind transparent.

DialSize (RT Professional)

Beschreibung

Legt den Durchmesser der Skalenscheibe in Bezug auf den kleineren Wert der Geometrieattribute "Width" und "Height" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DialSize**[=SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

SINGLE

Optional. Ein Wert oder eine Konstante, welche den Durchmesser der Skalenscheibe in Bezug auf den kleineren Wert der Geometrieattribute "Width" und "Height" festlegt.

DisplayButton2Plc (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayButtonComparison (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayButtonDelete (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayButtonFromPlc (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayButtonHelp (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayButtonNew (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayButtonSave (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayButtonSaveAs (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayComboBox (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayGridLines (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayLabeling (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayNumbers (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayOptions (RT Professional)

Beschreibung

Legt fest, ob Meldungen, deren Anzeige unterdrückt wurde, angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DisplayOptions**[=AlarmDisplayOptions]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl

AlarmDisplayOptions

0: Alle Meldungen

1: Nur eingeblendete Meldungen

2: Nur ausgeblendete Meldungen

DisplayStatusBar (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DisplayTable (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

DoubleClickAction (RT Professional)

Beschreibung

Legt die Aktion fest, die bei Doppelklick auf eine Meldezeile in Runtime ausgeführt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DoubleClickAction**[=AlarmControlActions]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

AlarmControlActions

Wert	Beschreibung	Erklärung
0	keine	Keine Aktion wird ausgeführt.
1	Loop in Alarm	Die Funktion "Loop in Alarm" wird aufgerufen.
2	Kommentar-Dialog öffnen	Die Tastenfunktion "Kommentar-Dialog" wird aufgerufen.
3	Infotext-Dialog öffnen	Die Tastenfunktion "Infotext-Dialog" wird aufgerufen.
4	Spalten abhängig	Die ausgeführte Aktion richtet sich nach der Spalte, in die Sie doppelgeklickt haben.

DrawInsideFrame (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**DrawInsideFrame**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- Circle
- CircleSegment
- CircularArc
- Ellipse
- EllipseSegment
- EllipticalArc
- GraphicIOField
- GraphicView
- OptionGroup

- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox
- Switch
- TubeArcObject

BOOLEAN

Optional. TRUE, wenn die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens gezeichnet wird.

Drive (RT Professional)

Beschreibung

Legt den Buchstaben des Laufwerks fest, das überwacht werden soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Drive**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

STRING

Optional. Ein Wert oder eine Konstante, welche den Buchstaben des Laufwerks festlegt, das überwacht werden soll.

3.5.5.5 Eigenschaften E-F (Panels, Comfort Panels, RT Advanced, RT Professional)

EdgeStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Linienstil des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EdgeStyle**[=LineStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- ComboBox
- Ellipse
- EllipseSegment
- GraphicIOField
- GraphicView
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- AlarmView
- Clock
- DateTimeField
- Gauge
- RecipeView
- Slider
- StatusForce
- Switch
- SysDiagControl

- TrendView
- UserView

LineStyle

Optional. Ein Wert oder eine Konstante, welche den Linienstil festlegt. Wertebereich von -1 bis 4.

Die Objekte "Ellipse", "Circle", "Rectangle" und "Polygon" unterstützen die Linienstile:

- hmiLineStyleSolid (0): durchgezogene Linie
- hmiLineStyleDash (1): gestrichelte Linie
- hmiLineStyleDot (2): gepunktete Linie
- hmiLineStyleDashDot (3): Strich-Punkt-Linie
- hmiLineStyleDashDotDot (4): Strich-Punkt-Punkt-Linie

Die Objekte "TextField" und "IOField" unterstützen nur die Linienstile:

- hmiLineStyleNone (-1): keine Linie
- hmiLineStyleSolid (0): durchgezogene Linie

Standardeinstellung: hmiLineStyleSolid

EditOnFocus (RT Professional)

Beschreibung

Legt fest, ob die Dateneingabe sofort möglich ist, wenn das Eingabefeld mit der Taste <Tab> ausgewählt ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EditOnFocus**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField
- SymbolicIOField

BOOLEAN

Optional. TRUE, wenn die Dateneingabe sofort möglich ist, wenn das Eingabefeld mit der Taste <Tab> ausgewählt ist.

Enabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob das angegebene Objekt in Runtime bedient werden kann.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Enabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- AlarmView
- Bar
- Button
- ChannelDiagnose
- CheckBox
- Circle
- CircleSegment
- CircularArc
- Clock
- ComboBox
- DateTimeField
- DiscSpaceView
- Ellipse
- EllipseSegment
- EllipticalArc
- FunctionTrendControl
- Gauge
- GraphicIOField
- GraphicView
- HTML-Browser
- IOField
- Line
- ListBox
- MediaPlayer

- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- PLCCodeViewer
- Polygon
- Polyline
- RecipeView
- Rectangle
- RoundButton
- S7GraphOverview
- Slider
- SmartClientView
- StatusForce
- Switch
- SymbolicIOField
- SymbolLibrary
- SysDiagControl
- TextField
- TrendRulerControl
- TrendView
- TubeArcObject
- TubeDoubleTeeObject
- Tubepolyline
- TubeTeeObject
- UserArchiveControl
- UserView
- WindowsSlider

BOOLEAN

Optional. TRUE, wenn das angegebene Objekt bedient werden kann.

EnableEdit (RT Professional)

Beschreibung

Legt fest, ob Sie die angezeigten Daten in Runtime ändern können.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EnableEdit**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl
- UserArchiveControl

BOOLEAN

TRUE: Die Daten sind in Runtime änderbar.

FALSE: Die Daten sind nicht änderbar.

EnableNavigateKeys (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

EndAngle (RT Professional)

Beschreibung

Legt den Winkel fest, um den der Endpunkt des angegebenen Objekts von der Nulllage (0°) abweicht.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EndAngle**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- CircleSegment
- CircularArc
- EllipseSegment
- EllipticalArc
- TubeArcObject

Int32

Optional. Ein Wert oder eine Konstante, welche den Winkel festlegt, um den der Endpunkt des angegebenen Objekts von der Nulllage (0°) abweicht.

EndLeft (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

EndStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, wie das Linienende des angegebenen Objekts dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EndStyle**[=LineEndStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Line
- Polyline

LineEndStyle

Optional. Ein Wert oder eine Konstante, die das Linienende festlegt. Wertebereich von 0 bis 6.

hmiLineEndStyleNone (0): Die Linie besitzt kein Endsymbol.

hmiLineEndStyleArrow (1): Die Linie endet mit einer Pfeilspitze.

hmiLineEndStyleFilledArrow (2): Die Linie endet mit einer ausgefüllten Pfeilspitze.

hmiLineEndStyleFilledArrowReversed (3): Die Linie endet mit einer umgekehrten Pfeilspitze.

hmiLineEndStyleLine (4): Die Linie endet mit einer senkrechten Linie.

hmiLineEndStyleCircle (5): Die Linie endet mit einem Kreis.

hmiLineEndStyleFilledCircle (6): Die Linie endet mit einem ausgefüllten Kreis.

EndTop (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

EntryNameCaption (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

ErrorDescription (RT Professional)

Beschreibung

Gibt eine der folgenden Fehlerbeschreibungen in englischer Sprache zurück:

Ausgabe	Beschreibung
" "	OK
"Operation Failed"	Durchführungsfehler
"Variable not found"	Variablenfehler
"Server down"	Server nicht verfügbar
"An error occured for one or several tags"	Multi Tag Error (Fehler bei einer oder mehreren Variablen)

Um eine Fehlerbeschreibung zu erhalten, führen Sie zuerst die Read-Methode aus.

Hinweis

Wenn der Fehler beim Zugriff über das TagSet-Objekt auftritt, werten Sie die ErrorDescription-Eigenschaft für jede Variable des TagSet-Objekts aus.

Zugriff in Runtime: Lesen

Syntax

Object.**ErrorDescription**

Object

Erforderlich. Ein Objekt vom Typ "Tag".

Beispiel

Das folgende Beispiel gibt die Fehlerbeschreibung für die Variable "Tag1" aus:

```
'VBS72
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objtag.Read
MsgBox objTag.ErrorDescription
```

Das folgende Beispiel fügt der TagSet-Auflistung zwei Variablen hinzu und gibt die ErrorDescription-Eigenschaft als Trace aus:

```
'VBS179
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
HMIRuntime.Trace "ErrorDescription: " & group.ErrorDescription & vbNewLine
```

Auf die ErrorDescription-Eigenschaft einer in der Auflistung enthaltenen Variable greifen Sie wie folgt zu:

```
HMIRuntime.Trace "ErrorDescription: " & group("Motor1").ErrorDescription & vbNewLine
```

Siehe auch

[Tag](#) (Seite 1332)

[TagSet \(Auflistung\)](#) (Seite 1336)

EvenRowBackColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ExportDelimiter (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ExportDirectoryChangeable (RT Professional)

Beschreibung

Legt fest, ob das Verzeichnis für den Datenexport in Runtime geändert werden kann.

Syntax

Object.**ExportDirectoryChangeable**=[BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Das Verzeichnis für den Datenexport kann in Runtime geändert werden.

FALSE: Das Verzeichnis für den Datenexport kann nicht in Runtime geändert werden.

ExportDirectoryname (RT Professional)

Beschreibung

Legt das Verzeichnis fest, in das die Runtime-Daten exportiert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExportDirectoryname**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl

- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welches das Verzeichnis festlegt.

ExportFileExtension (RT Professional)

Beschreibung

Legt die Dateierweiterung der Exportdatei fest. Bisher wird nur die Datenerweiterung "csv" unterstützt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExportFileExtension**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Legt die Dateierweiterung der Exportdatei fest.

ExportFilename (RT Professional)

Beschreibung

Legt den Dateinamen der Datei fest, in welche die Runtime-Daten exportiert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExportFilename**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Legt den Dateinamen der Datei fest, in welche die Runtime-Daten exportiert werden.

ExportFilenameChangeable (RT Professional)

Beschreibung

Legt fest, ob der Dateiname der Exportdatei in Runtime geändert werden kann.

Syntax

Object.**ExportFilenameChangeable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Der Dateiname der Exportdatei kann in Runtime geändert werden.

FALSE: Der Dateiname der Exportdatei kann nicht in Runtime geändert werden.

ExportFormatGuid (RT Professional)

Beschreibung

Legt die Zuordnung von Ident-Nummer und Export-Provider fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExportFormatGuid**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Zuordnung von Ident-Nummer und Export-Provider festlegt.

ExportFormatName (RT Professional)

Beschreibung

Legt das Dateiformat für den Export fest. Zurzeit steht nur das "csv"-Dateiformat für den Export zur Verfügung.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExportFormatName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche das Dateiformat für den Export festlegt.

ExportParameters (RT Professional)

Beschreibung

Legt die Parameter des ausgewählten Formats über den Eigenschaften-Dialog fest.
Zugriff in Runtime: Lesen und Schreiben.

Syntax

Object.**ExportParameters**

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

ExportParameters

Optional. Ein Wert oder eine Konstante, welche die Parameter des ausgewählten Formats über den Eigenschaften-Dialog festlegt.

ExportSelection (RT Professional)

Beschreibung

Legt fest, welche Runtime-Daten des Controls exportiert werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExportSelection**[=ExportRange]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl

- TrendRulerControl
- UserArchiveControl

ExportRange

Wert	Beschreibung	Erklärung
0	alles	Alle Runtime-Daten des Controls werden exportiert.
1	Auswahl	Die ausgewählten Runtime-Daten des Controls werden exportiert.

ExportShowDialog (RT Professional)

Beschreibung

Legt fest, ob der Dialog zum Datenexport in Runtime angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExportShowDialog**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

Optional. TRUE, wenn der Dialog in Runtime angezeigt wird.

ExtendedZoomingEnable (RT Professional)

Beschreibung

Legt fest, ob der Bediener das Bild in Runtime durch Drehen des Mausekkrades vergrößern oder verkleinern kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExtendedZoomingEnable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "Screen".

BOOLEAN

Optional. TRUE, wenn der Bediener das Bild in Runtime vergrößern und verkleinern kann.

Beispiel

Das folgende Beispiel zeigt, wie für das Bild NewPDL1 das Erweiterte Zoomen aktiviert wird:

```
'VBS155
Dim objScreen
Set objScreen = HMIRuntime.Screens("NewPDL1")
objScreen.ExtendedZoomingEnable = 1
```

Siehe auch

Screen (Seite 1320)

FieldLength (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, dass das Feld "Feldlänge Zeichenkette" nur gelesen werden kann.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FieldLength**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "IOField".

BOOLEAN

Optional. TRUE, wenn das Feld "Feldlänge Zeichenkette" nur gelesen werden kann.

Genutzt bei folgenden Objekttypen

IOField

Siehe auch

IOField (Seite 1418)

FieldLength (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FillColorMode (RT Professional)

Beschreibung

Legt die Art des Vordergrundes für das angegebene Objekt fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FillColorMode**[=SymbolLibraryColorMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

SymbolLibraryColorMode

hmiSymbolLibraryAppearanceOriginal (0): Die Oberfläche ist grau.

hmiSymbolLibraryAppearanceShaded (1): Die Darstellung ist schattiert.

hmiSymbolLibraryAppearanceSolid (2): Die Darstellung ist solid.

hmiSymbolLibraryAppearanceTransparent (3): Die Darstellung ist grau.

FillPatternColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Füllmusters des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**FillPatternColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- ComboBox
- Ellipse
- EllipseSegment
- GraphicView
- IOField
- ListBox
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- AlarmControl
- DateTimeField
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Füllmusters des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

FillStyle (RT Advanced)

Beschreibung

Legt das Füllmuster des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FillStyle**[=LineFillStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Line
- Polyline

LineFillStyle

Optional. Ein Wert oder eine Konstante, die das Füllmuster festlegt.

hmiLineFillStyleTransparent (0): Transparente Füllung

hmiLineFillStyleSolid (1): Objekt wird mit der angegebenen Farbe gefüllt

Standardeinstellung: hmiLineFillStyleSolid

FillstyleAlignment (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Ausrichtung innerhalb des angegebenen Objekts fest.

Filter (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FilterTag (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FilterText (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FirstConnectedObjectIndex (RT Professional)

Beschreibung

Legt die Indexnummer des oberen Verbinderpunktes fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FirstConnectedObjectIndex**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Connector".

Int

Int

Siehe auch

Connector (Seite 1379)

FirstConnectedObjectName (RT Professional)

Beschreibung

Legt den Objektnamen des Objekts fest, das am oberen Verbinderpunkt angedockt ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FirstConnectedObjectName**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Connector".

String

String

Siehe auch

Connector (Seite 1379)

FitToLargest (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FitToSize (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FixedAspectRatio (RT Professional)

Beschreibung

Legt fest, ob beim Ändern der Symbolgröße das Seitenverhältnis erhalten bleibt oder veränderlich ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FixedAspectRatio**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

BOOLEAN

Optional. TRUE, wenn bei Änderung der Symbolgröße das Seitenverhältnis erhalten bleibt.

FlashingColorOff (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**FlashingColorOff**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- Circle
- CircleSegment
- CircularArc
- Ellipse
- EllipseSegment
- EllipticalArc
- IOField
- Line
- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- SymbolicIOField
- TextField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Bar
- CheckBox
- Switch
- TubeArcObject

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Aus" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

FlashingColorOn (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" fest.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**FlashingColorOn**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- Circle
- CircleSegment
- CircularArc
- Ellipse
- EllipseSegment
- EllipticalArc
- IOField
- Line
- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- SymbolicIOField
- TextField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Bar
- CheckBox
- Switch
- TubeArcObject

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Rahmenlinie des angegebenen Objekts für den Blinkzustand "Ein" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

FlashingEnabled (RT Professional)

Beschreibung

Legt fest, ob die Rahmenlinie des angegebenen Objekts in Runtime blinkt.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FlashingEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "Line", "Polyline", "Ellipse", "Circle", "EllipseSegment", "CircleSegment", "EllipticalArc", "CircularArc", "Rectangle", "Polygon", "TextField", "IOField", "SymbolicIOField", "Button", "Switch", "GraphicIOField", "Bar", "Checkbox", "OptionGroup" oder "Connector".

BOOLEAN

Optional. TRUE, wenn die Rahmenlinie des Objekts blinkt.

Siehe auch

Line (Seite 1422)
Polyline (Seite 1459)
Ellipse (Seite 1386)
Circle (Seite 1369)
EllipseSegment (Seite 1388)
CircleSegment (Seite 1372)
EllipticalArc (Seite 1391)
CircularArc (Seite 1374)
Rectangle (Seite 1471)

Polygon (Seite 1456)
TextField (Seite 1505)
IOField (Seite 1418)
SymbolicIOField (Seite 1491)
Button (Seite 1360)
Switch (Seite 1488)
GraphicIOField (Seite 1409)
Bar (Seite 1353)
CheckBox (Seite 1366)
OptionGroup (Seite 1452)
Connector (Seite 1379)
RoundButton (Seite 1473)

FlashingOnLimitViolation (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FlashingRate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Blinkfrequenz der Rahmenlinie des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen/Schreiben

Syntax

Object.**FlashingRate**[=FlashingRate]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- Circle
- CircleSegment
- CircularArc
- Ellipse

- EllipseSegment
- EllipticalArc
- GraphicIOField
- IOField
- Line
- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- SymbolicIOField
- TextField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Bar
- CheckBox
- Switch
- TubeArcObject

FlashingRate

hmiFlashingRateSlow (0): Die Länge des Blinkintervalls ist 1000 ms.

hmiFlashingRateMedium (1): Die Länge des Blinkintervalls ist 500 ms.

hmiFlashingRateFast (2): Die Länge des Blinkintervalls ist 250 ms.

FlashTransparentColor (RT Professional)

Beschreibung

Legt die Farbe des Bitmap-Objektes eines Blinkbildes fest, die auf "transparent" gesetzt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FlashTransparentColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "GraphicIOField".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Bitmap-Objektes eines Blinkbildes festlegt, die auf "transparent" gesetzt wird.

Siehe auch

GraphicIOField (Seite 1409)

Flip (RT Professional)

Beschreibung

Spiegelt das Symbol an der vertikalen und/oder an der horizontalen Mittelachse des Symbols.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Flip**[=SymbolLibraryFlip]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

SymbolLibraryFlip

hmiSymbolLibraryFlipNone (0): Das Symbol wird nicht gespiegelt.

hmiSymbolLibraryFlipHorizontal (1): Das Symbol wird horizontal gespiegelt.

hmiSymbolLibraryFlipVertical (2): Das Symbol wird vertikal gespiegelt.

hmiSymbolLibraryFlipBoth (3): Das Symbol wird horizontal und vertikal gespiegelt.

FocusColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe für den Fokusrahmen des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FocusColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmView
- RecipeView
- Slider
- StatusForce

- Switch
- TrendView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Button
- GraphicIOField
- SymbolicIOField

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den Fokusrahmen festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

FocusWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Rahmenbreite des angegebenen Objekts fest, wenn das Objekt den Fokus besitzt.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FocusWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmView
- RecipeView
- Slider
- StatusForce
- Switch
- TrendView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Button
- GraphicIOField
- SymbolicIOField

Int32

Optional. Ein Wert oder eine Konstante, welche die Rahmenbreite in Pixeln festlegt. Wertebereich von 1 bis 10.

Font (RT Professional)

Beschreibung

Legt die Schriftart fest oder gibt sie zurück.

Das Font-Objekt hat die Untereigenschaften

- Size (Schriftgröße)
- Bold (Fett ja/nein)
- Name (Schriftart)
- Italic (Kursiv ja/nein)
- Underline (Unterstrichen ja/nein)
- StrikeThrough (Durchgestrichen ja/nein)

Werden zwei Font-Eigenschaften direkt zugewiesen, wird nur die Default-Eigenschaft "Name" übernommen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Font**[=Font]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

AlarmControl

Clock

FunctionTrendControl

OnlineTableControl

OnlineTrendControl

Slider

TrendRulerControl

UserArchiveControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Bar
- Button
- CheckBox
- ComboBox

- DateTimeField
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- ProtectedAreaNameView
- RangeLabelView
- RecipeView
- RoundButton
- SmartClientView
- Switch
- SymbolicIOField
- TextField
- TrendView
- ZoneLabelView

Font

Schriftart

Beispiel

```
'VBS74
Dim objControl1
Dim objControl2
Set objControl1 = ScreenItems("Control1")
Set objControl2 = ScreenItems("Control2")
objControl2.Font = objControl1.Font ' take over only the type of font
```

FontBold (RT Professional)

Beschreibung

Legt fest, ob der Text des angegebenen Objekts fett dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FontBold**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TextField
- Bar
- Button
- CheckBox
- ComboBox
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- RoundButton
- SymbolicIOField

BOOLEAN

Optional. TRUE, wenn der Text des angegebenen Objekts fett dargestellt wird.

FontItalic (RT Professional)

Beschreibung

Legt fest, ob der Text des angegebenen Objekts kursiv dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FontItalic**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TextField
- Button
- CheckBox
- ComboBox
- IOField
- ListBox
- MultiLineEdit
- OptionGroup

- RoundButton
- SymbolicIOField

BOOLEAN

Optional. TRUE, wenn der Text des angegebenen Objekts kursiv dargestellt wird.

FontName (RT Professional)

Beschreibung

Legt die Schriftart des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FontName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TextField
- Bar
- Button
- CheckBox
- ComboBox
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- RoundButton
- SymbolicIOField

STRING

Optional. Ein Wert oder eine Konstante, welche die Schriftart des angegebenen Objekts festlegt.

FontSize (RT Professional)

Beschreibung

Legt die Schriftgröße des Texts des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FontSize**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TextField
- Bar
- Button
- CheckBox
- ComboBox
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- RoundButton
- SymbolicIOField

Int32

Optional. Ein Wert oder eine Konstante, welche die Schriftgröße des Texts des angegebenen Objekts festlegt.

FontUnderline (RT Professional)

Beschreibung

Legt fest, ob der Text des angegebenen Objekts unterstrichen dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FontUnderline**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TextField
- Button
- CheckBox
- ComboBox
- IOField
- ListBox

- MultiLineEdit
- OptionGroup
- RoundButton
- SymbolicIOField

BOOLEAN

Optional. TRUE, wenn der Text des angegebenen Objekts unterstrichen dargestellt wird.

ForeColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Schriftfarbe des Texts des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- CheckBox
- ComboBox
- DateTimeField
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- RecipeView
- RoundButton
- Slider
- Switch
- SymbolLibrary
- SymbolicIOField
- TextField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- AlarmView

Color

Optional. Ein Wert oder eine Konstante, welche die Schriftfarbe des Texts des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

ForeColorTransparency (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

FormatPattern (RT Professional)

Beschreibung

Legt das Format des Ausgabewerts fest.

Zugriff in Runtime: Lesen

Syntax

Object.**FormatPattern**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

STRING

Optional. Ein Wert oder eine Konstante, welche das Format des Ausgabewerts festlegt.

Free (RT Professional)

Beschreibung

Gibt die Größe des freien Speicherplatzes zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Free**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die Größe des freien Speicherplatzes zurück gibt.

FreePercent (RT Professional)

Beschreibung

Gibt die Messwerte für den freien Speicherplatz in Prozent zurück. Die Werte können in Runtime abgefragt werden. Die Werte können nicht vordefiniert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FreePercent**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

Int32

Optional. Ein Wert oder eine Konstante, welche die Messwerte für den freien Speicherplatz in Prozent zurück gibt.

3.5.5.6 Eigenschaften G-H (Panels, Comfort Panels, RT Advanced, RT Professional)

Gradation (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Wertdifferenz zwischen zwei Hauptteilstrichen des Objekts "Gauge" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Gradation**[=SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

SINGLE

Optional. Ein Wert oder eine Konstante, welche die Wertdifferenz festlegt.

GraphDirection (RT Professional)

Beschreibung

Legt fest, an welchem Rand des Kurvenfensters die aktuellen Werte angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**GraphDirection**[=GraphDirection]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

GraphDirection

0: Positive Werte werden nach rechts und nach oben abgetragen.

-1: Positive Werte werden nach links und nach oben abgetragen.

-2: Positive Werte werden nach rechts und nach unten abgetragen.

-3: Positive Werte werden nach links und nach unten abgetragen.

GridLineColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Rasterlinien fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**GridLineColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- AlarmView
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- TrendView

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Rasterlinien festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

GridlineFillColor (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

GridLineStyle (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

GridLineWidth (RT Professional)

Beschreibung

Legt die Stärke der Trennlinien in Pixel fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**GridLineWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Stärke der Trennlinie festlegt.

HeaderFont (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

Height (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Höhe des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben.

Syntax

Object.**Height**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- AlarmView
- ApplicationWindow
- Bar
- BatteryView
- Button
- ChannelDiagnose
- CheckBox

- Circle
- CircleSegment
- CircularArc
- Clock
- ComboBox
- DateTimeField
- DiscSpaceView
- Ellipse
- EllipseSegment
- EllipticalArc
- FunctionTrendControl
- Gauge
- GraphicIOField
- GraphicView
- HTML-Browser
- IOField
- Line
- ListBox
- MediaPlayer
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- PLCCodeViewer
- Polygon
- Polyline
- ProtectedAreaNameView
- RangeLabelView
- RangeQualityView
- RecipeView
- Rectangle
- RoundButton
- Screenwindow
- Slider
- SmartClientView

- StatusForce
- Switch
- SymbolLibrary
- SymbolicIOField
- SysDiagControl
- TextField
- TrendRulerControl
- TrendView
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- UserArchiveControl
- UserView
- WlanQualityView
- WindowsSlider
- ZoneLabelView
- ZoneQualityView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- S7GraphOverview

Int32

Optional. Ein Wert oder eine Konstante, welche die Höhe in Pixel festlegt.

Beispiel

Das folgende Beispiel halbiert die Höhe aller Objekte des Bildes "NewPDL1", deren Name mit "Circle" beginnt:

```
'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
```

Siehe auch

- Line (Seite 1422)
- Polyline (Seite 1459)
- Ellipse (Seite 1386)
- Circle (Seite 1369)
- EllipseSegment (Seite 1388)
- CircleSegment (Seite 1372)
- EllipticalArc (Seite 1391)
- CircularArc (Seite 1374)
- Rectangle (Seite 1471)
- Polygon (Seite 1456)
- TextField (Seite 1505)
- IOField (Seite 1418)
- SymbolicIOField (Seite 1491)
- Button (Seite 1360)
- Switch (Seite 1488)
- GraphicView (Seite 1412)

GraphicIOField (Seite 1409)
Bar (Seite 1353)
Clock (Seite 1377)
Gauge (Seite 1406)
Slider (Seite 1481)
SymbolLibrary (Seite 1496)
OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)
AlarmControl (Seite 1338)
HTMLBrowser (Seite 1416)
CheckBox (Seite 1366)
OptionGroup (Seite 1452)
WindowSlider (Seite 1538)
Connector (Seite 1379)
ScreenWindow (Seite 1477)
DiskSpaceView (Seite 1384)
ChannelDiagnose (Seite 1364)
ScriptDiagnostics (Seite 1479)
Group (Seite 1415)
ForeignControl (Seite 1393)
ProtectedAreaName (Seite 1463)
UserView (Seite 1536)
TubeTeeObject (Seite 1526)
TubePolyline (Seite 1524)
TubeDoubleTeeObject (Seite 1522)
TubeArcObject (Seite 1520)
MultiLineEdit (Seite 1429)
MediaPlayer (Seite 1427)
Listbox (Seite 1424)
DateTimeField (Seite 1382)
UserArchiveControl (Seite 1528)
TrendRulerControl (Seite 1508)
ProjectName (Seite 1461)
AlarmView (Seite 1350)

BatteryView (Seite 1359)
RangeLabelView (Seite 1464)
ZoneQualityView (Seite 1543)
ZoneLabelView (Seite 1542)
WLANQualityView (Seite 1541)
TrendView (Seite 1516)
SystemDiagnoseWindow (Seite 1501)
SystemDiagnoseView (Seite 1498)
StatusForce (Seite 1486)
SmartClientView (Seite 1484)
RecipeView (Seite 1467)
RangeQualityView (Seite 1466)

HelpText (Panels, Comfort Panels, RT Advanced)

Beschreibung

Gibt den Tooltip zurück, der in Runtime als Bedienungshilfe für das angegebene Objekt angezeigt wird.

Zugriff in Runtime: Lesen

Syntax

Object.**HelpText**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- DateTimeField
- GraphicIOField
- IOField
- Switch
- SymbolicIOField
- TrendView

STRING

Optional. Ein Wert oder eine Konstante, welche den Inhalt des Tooltips angibt, der in Runtime als Bedienungshilfe für das angegebene Objekt angezeigt wird.

HiddenInput (RT Professional)

Beschreibung

Legt fest, ob der Eingabewert während der Eingabe angezeigt wird oder ein * für jedes Zeichen.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HiddenInput**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

BOOLEAN

Optional. TRUE, wenn der Eingabewert während der Eingabe nicht angezeigt wird. Für jedes Zeichen wird ein * angezeigt.

HighLimitColor (RT Professional)

Beschreibung

Legt die Farbe der oberen oder rechten Schaltfläche zum Scrollen in einer Bildlaufleiste fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HighLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- WindowsSlider

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der oberen oder rechten Schaltfläche zum Scrollen in einer Bildlaufleiste festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können sie auch die VBS-Farbkonstanten wie vbRed und vbGreen verwenden.

Hitlist (RT Professional)

Beschreibung

Kein Zugriff in Runtime.

HitlistColumnAdd (RT Professional)

Beschreibung

Übernimmt den ausgewählten Meldeblock aus der Liste der vorhandenen Meldeblöcke in die Liste der gewählten Meldeblöcke.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistColumnAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Übernimmt den ausgewählten Meldeblock aus der Liste der vorhandenen Meldeblöcke in die Liste der gewählten Meldeblöcke.

HitlistColumnCount (RT Professional)

Beschreibung

Legt die Anzahl der gewählten Meldeblöcke fest, die in der Hitliste in Runtime angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistColumnCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Legt die Anzahl der gewählten Meldeblöcke fest, die in der Hitliste in Runtime angezeigt werden.

HitlistColumnIndex (RT Professional)

Beschreibung

Referenziert einen für die Hitliste gewählten Meldeblock. Unter Verwendung des Attributs können Sie einem bestimmten Meldeblock der Hitliste die Werte anderer Attribute zuweisen.

Gültige Werte für "HitlistColumnIndex" liegen zwischen 0 und "HitlistColumnCount" minus 1. Das Attribut "HitlistColumnCount" gibt die Anzahl der für die Hitliste gewählten Meldeblöcke an. Das Attribut "HitlistColumnIndex" ist über das Attribut **HitlistColumnRepos** dynamisierbar.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistColumnIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Referenziert einen für die Hitliste gewählten Meldeblock.

HitlistColumnName (RT Professional)

Beschreibung

Legt den Namen des Meldeblocks der Hitliste fest, der mit dem Attribut "HitlistColumnIndex" referenziert wird. Den Namen können Sie nicht ändern.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistColumnName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt den Namen des Meldeblocks der Hitliste fest, der mit dem Attribut "HitlistColumnIndex" referenziert wird.

HitlistColumnRemove (RT Professional)

Beschreibung

Entfernt den markierten Meldeblock aus der Liste der gewählten Meldeblöcke und fügt ihn in die Liste der vorhandenen Meldeblöcke ein.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistColumnRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Entfernt den markierten Meldeblock aus der Liste der gewählten Meldeblöcke und fügt ihn in die Liste der vorhandenen Meldeblöcke ein.

HitlistColumnRepos (RT Professional)

Beschreibung

Ändert die Reihenfolge der Meldeblöcke. "Auf" und "Ab" bewegen den ausgewählten Meldeblock in der Liste nach oben oder unten. Dadurch wird in Runtime der Meldeblock im Control weiter vorne oder hinten platziert.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistColumnRepos**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Ändert die Reihenfolge der Meldeblöcke.

HitlistColumnSort (RT Professional)

Beschreibung

Legt fest, wie der im "HitlistColumnIndex" referenzierte Meldeblock für die Hitliste sortiert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.HitlistColumnSort[=SortMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

SortMode

0: Keine Sortierung.

1: Aufsteigende Sortierung vom kleinsten zum größten Wert

2: Absteigende Sortierung vom größten zum kleinsten Wert

HitlistColumnSortIndex (RT Professional)

Beschreibung

Legt die Sortierreihenfolge des im "HitlistColumnIndex" referenzierten Meldeblocks der Hitliste fest. Wenn Sie den Wert auf "0" setzen, wird das Sortierkriterium in "HitlistColumnSort" entfernt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.HitlistColumnSortIndex[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Legt die Sortierreihenfolge des im "HitlistColumnIndex" referenzierten Meldeblocks der Hitliste fest. Wenn Sie den Wert auf "0" setzen, wird das Sortierkriterium in "HitlistColumnSort" entfernt.

HitlistColumnVisible (RT Professional)

Beschreibung

Legt eine Liste mit den gewählten Meldeblöcke der Hitliste fest, die in Runtime im Control verwendet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistColumnVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

Optional. Legt eine Liste mit den gewählten Meldeblöcke der Hitliste fest, die in Runtime im Control verwendet werden.

HitlistDefaultSort (RT Professional)

Beschreibung

Legt die Standardsortierung in den Tabellenspalten der Hitliste fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistDefaultSort**[=SortMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

SortMode

0: Die Liste wird aufsteigend nach Häufigkeit sortiert.

1: Die Liste wird absteigend nach Häufigkeit sortiert.

HitlistMaxSourceItems (RT Professional)

Beschreibung

Legt die maximale Zahl der Datensätze fest, die aus dem Meldearchiv zum Erstellen der Hitliste verwendet werden. Der Wert ist innerhalb von "0 - 10000" frei wählbar.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.HitlistMaxSourceItems[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Ein Wert oder eine Konstante, welche die maximale Zahl der Datensätze festlegt, die aus dem Meldearchiv zum Erstellen der Hitliste verwendet werden.

HitlistMaxSourceItemsWarn (RT Professional)

Beschreibung

Legt fest, ob ein Warnhinweis ausgegeben wird, wenn die maximale Zahl der Datensätze erreicht ist, die in "HitlistMaxSourceItems" festgelegt wurde.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.HitlistMaxSourceItemsWarn[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn ein Warnhinweis ausgegeben wird, sobald die maximale Zahl der Datensätze erreicht ist, die in "HitlistMaxSourceItems" festgelegt wurde.

Siehe auch

AlarmControl (Seite 1338)

HitListRelTimeFactor (RT Professional)

Beschreibung

Legt den Zeitfaktor fest, der mit dem Zeittyp "HitlistRelTimeFactorType" den Zeitraum bestimmt, für den die Statistik der Hitliste erstellt wird.

Wenn Sie keinen Zeitraum festlegen wollen, setzen Sie den Zeitfaktor auf "0".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistRelTimeFactor**[=Long]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Long

Optional. Ein Wert oder eine Konstante, welche den Zeitfaktor festlegt.

Siehe auch

AlarmControl (Seite 1338)

HitlistRelTimeFactorType (RT Professional)

Beschreibung

Legt den Zeittyp fest, der mit dem Zeitfaktor "HitlistRelTimeFactor" den Zeitraum bestimmt, für den die Statistik der Hitliste erstellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitlistRelTimeFactorType**[=AlarmViewAdvancedTimeFactorUnit]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

AlarmViewAdvancedTimeFactorUnit

hmiAlarmViewAdvancedTimeFactorUnitMinute (0): Der Zeitbereich für die Statistik wird in Minuten gemessen.

hmiAlarmViewAdvancedTimeFactorUnitHour (1): Der Zeitbereich für die Statistik wird in Stunden gemessen.

hmiAlarmViewAdvancedFactorUnitDay (2): Der Zeitbereich für die Statistik wird in Tagen gemessen.

hmiAlarmViewAdvancedTimeFactorUnitWeek (3): Der Zeitbereich für die Statistik wird in Wochen gemessen.

hmiAlarmViewAdvancedTimeFactorUnitMonth (4): Der Zeitbereich für die Statistik wird in Monaten gemessen.

Siehe auch

AlarmControl (Seite 1338)

HitListRelTime (RT Professional)

Beschreibung

Legt fest, ob ein Zeitbereich für die Statistik verwendet wird.

Wenn Sie keinen Zeitraum festlegen wollen, setzen Sie den Zeitfaktor auf "0".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HitListRelTime** [=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Wenn kein Zeitbereich in der Selektion angegeben ist, wird der angegebene Zeitbereich für die Statistik verwendet

FALSE: Der angegebene Zeitbereich wird nicht verwendet

HorizontalAlignment (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die horizontale Ausrichtung des Texts innerhalb des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HorizontalAlignment**[=HorizontalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- CheckBox
- ComboBox
- DateTimeField
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- RoundButton
- Switch
- SymbolicIOField
- TextField

HorizontalAlignment

Optional. Ein Wert oder eine Konstante, welche die horizontale Ausrichtung des Textes festlegt.

hmiAlignmentLeft (0): Der Text wird im Objekt linksbündig angeordnet.

hmiAlignmentCentered (1): Der Text wird im Objekt horizontal zentriert angeordnet.

hmiAlignmentRight (2): Der Text wird im Objekt rechtsbündig angeordnet.

HorizontalGridLines (RT Professional)

Beschreibung

Legt fest, ob horizontale Trennlinien angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HorizontalGridLines**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Horizontale Trennlinien werden angezeigt.

FALSE: Horizontale Trennlinien werden nicht angezeigt.

HorizontalScrollBarPosition (RT Professional)

Beschreibung

Legt die horizontale Verschiebung der Bildlaufleiste in einem Bildfenster mit Rollbalken fest.

Das Bild wird im Bildfenster mit der horizontalen bzw. vertikalen Verschiebung der Bildlaufleisten dargestellt. Wenn Sie das Bild ausgeschnitten darstellen wollen, in der die Bildlaufleisten sich am linken und oberen Rand des Bildes befinden, verwenden Sie für den Ursprung dieses Ausschnitts die Eigenschaften "OffsetLeft" und "OffsetTop".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HorizontalScrollBarPosition**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ScreenWindow

Int32

Optional. Ein Wert oder eine Konstante, welche die horizontale Verschiebung der Bildlaufleiste in einem Bildfenster mit Rollbalken festlegt.

Hotkey (RT Professional)

Beschreibung

Gibt beim Objekt Button die Funktionstaste für die Mausbedienung zurück.

Nur Lese-Zugriff.

Siehe auch

Button (Seite 1360)

HourNeedleHeight (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Länge des Stundenzeigers im Objekt "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.HourNeedleHeight[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Int32

Optional. Ein Wert oder eine Konstante, welche die Länge des Stundenzeigers festlegt.

Geben Sie die Länge des Stundenzeigers in Prozent an, bezogen auf den Radius des Zifferblattes.

HourNeedleWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Breite des Stundenzeigers im Objekt "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.HourNeedleWidth[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des Stundenzeigers festlegt.

Geben Sie die Breite in Prozent an, bezogen auf die doppelte Länge des Stundenzeigers.

3.5.5.7 Eigenschaften I-J (Panels, Comfort Panels, RT Advanced, RT Professional)

IconSpace (RT Professional)

Beschreibung

Legt den Abstand zwischen Symbol und Text in den Zellen der Tabelle fest. Der Wert ist wirksam, wenn Symbol und Text angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**IconSpace**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Wert, der den Abstand festlegt.

Index (RT Professional)

Beschreibung

Legt den Index des ausgewählten Textfeldes fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Index**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- CheckBox
- ComboBox
- ListBox
- OptionGroup

Int32

Optional. Ein Wert oder eine Konstante, welche den Index des ausgewählten Textfeldes festlegt.

InnerBackColorOff (RT Advanced)

Beschreibung

Legt die Farbe unter dem Schieber des Objekts "Switch" fest, wenn sich das Objekt im Zustand AUS befindet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.InnerBackColorOff[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den Zustand AUS festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

InnerBackColorOn (RT Advanced)

Beschreibung

Legt die Farbe unter dem Schieber des Objekts "Switch" fest, wenn sich das Objekt im Zustand EIN befindet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.InnerBackColorOn[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Switch

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den Zustand EIN festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

InputValue (RT Professional)

Beschreibung

Definiert den im EA-Feld vom Benutzer eingegebenen Wert. Der Wert wird beim Setzen der Eigenschaft nicht im EA-Feld angezeigt.

Wenn Sie wollen, dass der eingegebene Wert nach Bestätigung mit der Taste <Return> im EA-Feld angezeigt wird, projektieren Sie eine Direktverbindung zwischen den Eigenschaften "Eingabewert" und "Ausgabewert". Die Direktverbindung ist nur sinnvoll, wenn am Ausgabewert keine Variablenverbindung projiziert ist, der Benutzer aber trotzdem den eingegebenen Wert abfragen möchte, z. B. über einen Skript.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**InputValue**[=Object]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField
- SymbolicIOField

Object

Optional. Ein Wert oder eine Konstante, welche den Eingabewert festlegt.

Instance (RT Professional)

Beschreibung

Gibt eine Instanz des Alarm-Objekts zurück.

IntegerDigits (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Anzahl der Vorkommastellen (0 bis 20) fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**IntegerDigits**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Vorkommastellen (0 bis 20) festlegt.

Interval (RT Professional)

Beschreibung

Legt die Zeitintervalle fest, zum Aktualisieren der angezeigten Messwerte. Sie geben den Wert in Minuten an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Interval**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

Int32

Optional. Ein Wert oder eine Konstante, welche die Zeitintervalle festlegt, zum Aktualisieren der angezeigten Messwerte.

ItemBorderStyle (RT Professional)

Beschreibung

Legt die Linienart der Trennlinien in der Auswahlliste des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ItemBorderStyle**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicOField

Int32

hmiLineStyleNone (-1): Die Auswahlliste hat keine Trennlinien.

hmiLineStyleSolid (0): Die Auswahlliste hat durchgezogene Trennlinien.

hmiLineStyleDash (1): Die Auswahlliste hat gestrichelte Trennlinien.

hmiLineStyleDot (2): Die Auswahlliste hat gepunktete Trennlinien.

hmiLineStyleDashDot (3): Die Auswahlliste hat Strichpunktlinien als Trennlinien.

hmiLineStyleDashDotDot (4): Die Auswahlliste hat Strich-Punkt-Punkt-Linien als Trennlinien.

JumpToLimitsAfterMouseClicked (RT Professional)

Beschreibung

Legt fest, ob der Schieberegler auf den zugehörigen Endwert gestellt wird. Der Endwert ist der Minimal- oder Maximalwert. Um den Schieberegler auf den Endwert zu stellen, klickt der Anwender auf den Bereich außerhalb der aktuellen Reglereinstellung.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**JumpToLimitsAfterMouseClicked**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- WindowsSlider

BOOLEAN

Optional. TRUE, wenn der Schieberegler auf den zugehörigen Endwert gestellt wird.

3.5.5.8 Eigenschaften K-L (Panels, Comfort Panels, RT Advanced, RT Professional)

LabelColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Skalenbeschriftung im Objekt "Slider" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LabelColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Skalenbeschriftung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Language (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die aktuelle Runtime-Sprache fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Language**[= LONG]

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

LONG

Optional. Ein Wert oder eine Konstante, welche die Länderkennung enthält.

Bemerkungen

Die Runtime-Sprache geben Sie in VBS mit einer Länder-Kennung an, z.B. 1031 für Deutsch, 2057 für Englisch etc. Eine Übersicht aller Länder-Kennungen finden Sie in den Grundlagen von VBScript unter dem Thema "Gebietsschema-ID (LCID)-Diagramm".

Siehe auch

HMIRuntime (Seite 1313)

LargeTicksBold (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob die langen Teilstriche einer Skala fett angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LargeTicksBold**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn die langen Teilstriche einer Skala fett angezeigt werden.

LargeTicksSize (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Länge der langen Teilstriche einer Skala fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LargeTicksSize**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Int32

Optional. Ein Wert oder eine Konstante, welche die Länge der langen Teilstriche einer Skala festlegt.

LastConnectedObjectIndex (RT Professional)

Beschreibung

Legt die Indexnummer des Verbindungspunkts zum letzten, verbundene Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LastConnectedObjectIndex**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Connector".

Int

Int

Siehe auch

Connector (Seite 1379)

LastConnectedObjectName (RT Professional)

Beschreibung

Legt den Objektname des Objekts fest, das am unteren Verbinderpunkt angedockt ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LastConnectedObjectName**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Connector".

String

String

Siehe auch

Connector (Seite 1379)

LastError (RT Professional)

Beschreibung

Gibt einen Fehlercode über den Erfolg der letzten Operation zurück, z. B. Informationen zum Schreib- oder Lesevorgang einer Variablen:

Code in Hexadezimalschreibweise	Beschreibung
0x00000000	OK
0x80040001	Durchführungsfehler
0x80040002	Variablenfehler

Code in Hexadezimalschreibweise	Beschreibung
0x80040003	Server nicht verfügbar
0x80040004	Multi Tag Error; Fehler bei einer oder mehreren Variablen

Um eine Fehlerbeschreibung zu erhalten, führen Sie zuerst die Read-Methode aus.

Hinweis

Wenn der Fehler beim Zugriff über das TagSet-Objekt auftritt, werten Sie die LastError-Eigenschaft für jede Variable des TagSet-Objekts aus.

Um eine Aussage über die Qualität des gelieferten Werts zu erhalten, verwenden Sie die Eigenschaft "QualityCode". Um eine Beschreibung des Fehlers zu erhalten, verwenden Sie die Eigenschaft "ErrorDescription".

Zugriff in Runtime: Lesen

Syntax

Object.**LastError**

Object

Erforderlich. Ein Objekt vom Typ "Tag".

Beispiel

Das folgende Beispiel gibt den Fehlercode für die Variable "Tag1" aus:

```
'VBS77
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
MsgBox objTag.LastError
```

Das folgende Beispiel fügt der TagSet-Auflistung zwei Variablen hinzu und gibt die LastError-Eigenschaft als Trace aus:

```
'VBS178
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
HMIRuntime.Trace "LastError: " & group.LastError & vbNewLine
```

Auf die LastError-Eigenschaft einer in der Auflistung enthaltenen Variable greifen Sie wie folgt zu:

```
HMIRuntime.Trace "LastError: " & group("Motor1").LastError & vbNewLine
```

Siehe auch

Tag (Seite 1332)

TagSet (Auflistung) (Seite 1336)

Layer (RT Advanced, RT Professional)

Beschreibung

Gibt die Ebene als LONG im Bild zurück, in der sich ein Objekt befindet. Insgesamt stehen 32 Ebenen zur Verfügung, wobei Ebene "0" die unterste und Ebene "31" die oberste Ebene ist.

Innerhalb einer Ebene liegen die zuerst projizierten Objekte im Hintergrund.

Zugriff in Runtime: Lesen

Syntax

Object.**Layer**

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem".

Hinweis

Die Layer-Eigenschaft, gibt die Ebene aus, in der sich das Objekt befindet. Die Ebene "0" wird als Ebene "0" ausgegeben.

Beim Zugriff werden in VBS die Ebenen von 1 an gezählt. Sprechen Sie deshalb die Ebene "1" mit layers(2) an.

Beispiel

Das folgende Beispiel gibt für alle Objekte des Bildes "NewPDL1" Name und Ebene aus:

```
'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
```

Siehe auch

ScreenItem (Seite 1323)
HTMLBrowser (Seite 1416)
Group (Seite 1415)
ForeignControl (Seite 1393)
DateTimeField (Seite 1382)
ZoneQualityView (Seite 1543)
ZoneLabelView (Seite 1542)
WLANQualityView (Seite 1541)
ProtectedAreaName (Seite 1463)
RangeLabelView (Seite 1464)
RangeQualityView (Seite 1466)
ScreenWindow (Seite 1477)
ScriptDiagnostics (Seite 1479)
Slider (Seite 1481)
SmartClientView (Seite 1484)
StatusForce (Seite 1486)
SymbolLibrary (Seite 1496)
BatteryView (Seite 1359)
ChannelDiagnose (Seite 1364)
CheckBox (Seite 1366)
GraphicIOField (Seite 1409)
Button (Seite 1360)
Bar (Seite 1353)
WindowSlider (Seite 1538)
UserView (Seite 1536)
TubeTeeObject (Seite 1526)
TubePolyline (Seite 1524)
TubeDoubleTeeObject (Seite 1522)
TubeArcObject (Seite 1520)
Switch (Seite 1488)
Rectangle (Seite 1471)
MultiLineEdit (Seite 1429)
MediaPlayer (Seite 1427)
Line (Seite 1422)

Listbox (Seite 1424)
AlarmView (Seite 1350)
Circle (Seite 1369)
CircleSegment (Seite 1372)
CircularArc (Seite 1374)
Clock (Seite 1377)
Connector (Seite 1379)
DiskSpaceView (Seite 1384)
Ellipse (Seite 1386)
EllipseSegment (Seite 1388)
EllipticalArc (Seite 1391)
Gauge (Seite 1406)
GraphicView (Seite 1412)
Polygon (Seite 1456)
Polyline (Seite 1459)
RoundButton (Seite 1473)
SymbolicIOField (Seite 1491)
IOField (Seite 1418)
TextField (Seite 1505)
OptionGroup (Seite 1452)
RecipeView (Seite 1467)
TrendView (Seite 1516)
UserArchiveControl (Seite 1528)
SystemDiagnoseWindow (Seite 1501)
SystemDiagnoseView (Seite 1498)
TrendRulerControl (Seite 1508)
ProjectName (Seite 1461)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)
OnlineTrendControl (Seite 1443)
AlarmControl (Seite 1338)

LayerDeclutteringEnable (RT Professional)

Beschreibung

Gibt an, ob die Ebenen eines Bildes abhängig vom eingestellten Minimal- und Maximalzoom ein- oder ausgeblendet werden.

Zugriff in Runtime: Lesen

Syntax

Object.**LayerDeclutteringEnable**

Object

Erforderlich. Ein Objekt vom Typ "Screen".

Beispiel:

Das Beispiel gibt die LayerDecluttering-Eigenschaft des Bildes "NewPDL1" als Trace aus.

```
'VBS156  
Dim objScreen  
Set objScreen = HMIRuntime.Screens("NewPDL1")  
HMIRuntime.Trace "Enable: " & objScreen.LayerDeclutteringEnable & vbNewLine
```

Siehe auch

Screen (Seite 1320)

Layers (RT Professional)

Beschreibung

Gibt ein Objekt vom Typ "Layers" zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**Layers**

Object

Erforderlich. Ein Objekt vom Typ "Screen".

Siehe auch

Screen (Seite 1320)

Left (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Wert der X-Koordinate des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Left**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem". Diese Eigenschaft ist eine Standardeigenschaft des ScreenItem-Objekts und damit bei allen Ausprägungen verfügbar.

DOUBLE

Optional. Ein Wert oder eine Konstante, die den Wert der X-Koordinate in Pixel enthält (gemessen vom linken oberen Bildrand).

Bemerkungen

Die X-Koordinate bezieht sich auf die linke obere Ecke des objektumfassenden Rechtecks. In Runtime werden außerdem die Bildgrenzen überwacht. Wenn der zugewiesene Koordinatenwert die Displaygröße übersteigt, wird die benutzerdefinierte Funktion mit einer Fehlermeldung abgebrochen.

Genutzt bei folgenden Objekttypen

AUTOHOTSPOT
AUTOHOTSPOT
AUTOHOTSPOT

LeftOffset (RT Professional)

Beschreibung

Legt den Abstand des Bildes vom linken Rand des Bildfensters fest.

Das Bild wird aus dem Bildfenster ausgeschnitten dargestellt. Die Bildlaufleisten befinden sich am linken und oberen Rand des Bildes. Wenn Sie das Bild im Bildfenster mit der horizontalen und vertikalen Verschiebung der Bildlaufleisten darstellen wollen, verwenden Sie für die Verschiebung die Eigenschaften "HorizontalScrollBarPosition" und "VerticalScrollBarPosition".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LeftOffset**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Screenwindow

Int32

Optional. Ein Wert oder eine Konstante, welche den Abstand des Bildes vom linken Rand des Bildfensters festlegt.

Limit4LowerLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den unteren Grenzwert für "Reserve4" fest.

Die Eigenschaft "Limit4LowerLimitEnabled" muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve4" überwacht werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Limit4LowerLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den unteren Grenzwert für "Reserve4" festlegt.

Bemerkungen

Die Eigenschaft "Limit4LowerLimitRelative" legt fest, ob das Objekt prozentual oder absolut ausgewertet wird.

Limit4LowerLimitColor (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "Reserve4" fest.

Die Eigenschaft "Limit4LowerLimitEnabled" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Limit4LowerLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den unteren Grenzwert "Reserve4" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Limit4LowerLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert "Reserve4" überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Limit4LowerLimitEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der untere Grenzwert "Reserve4" überwacht wird.

Bemerkungen

Folgende Werte werden über die Eigenschaften "Limit4LowerLimit", "Limit4LowerLimitColor" und "Limit4LowerLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

Limit4LowerLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert "Reserve4" prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit4LowerLimitRelative[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional.

TRUE: Der untere Grenzwert "Reserve4" wird prozentual ausgewertet.

FALSE: Der untere Grenzwert "Reserve4" wird absolut ausgewertet.

Limit4UpperLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den oberen Grenzwert für "Reserve4" fest.

Die Eigenschaft "Limit4UpperLimitEnabled" muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve4" überwacht werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit4UpperLimit[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den oberen Grenzwert für "Reserve4" festlegt.

Bemerkungen

Die Eigenschaft "Limit4UpperLimitRelative" legt fest, ob das Objekt prozentual oder absolut ausgewertet wird.

Limit4UpperLimitColor (RT Professional)

Beschreibung

Legt die Farbe für den oberen Grenzwert "Reserve4" fest.

Die Eigenschaft "Limit4UpperLimitEnabled" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit4UpperLimitColor[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den oberen Grenzwert "Reserve4" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Limit4UpperLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der obere Grenzwert "Reserve4" überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit4UpperLimitEnabled[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der obere Grenzwert "Reserve4" überwacht wird.

Bemerkungen

Folgende Werte werden über die Eigenschaften "Limit4UpperLimit", "Limit4UpperLimitColor" und "Limit4UpperLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

Limit4UpperLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der obere Grenzwert "Reserve4" prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit4UpperLimitRelative[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional.

TRUE: Der untere Grenzwert "Reserve4" wird prozentual ausgewertet.

FALSE: Der untere Grenzwert "Reserve4" wird absolut ausgewertet.

Limit5LowerLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den unteren Grenzwert für "Reserve5" fest.

Die Eigenschaft "Limit5LowerLimitEnabled" muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve5" überwacht werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Limit5LowerLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den unteren Grenzwert für "Reserve5" festlegt.

Bemerkungen

Die Eigenschaft "Limit5LowerLimitRelative" legt fest, ob das Objekt prozentual oder absolut ausgewertet wird.

Limit5LowerLimitColor (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "Reserve5" fest.

Die Eigenschaft "Limit5LowerLimitEnabled" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Limit5LowerLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den unteren Grenzwert "Reserve5" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Limit5LowerLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert "Reserve5" überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit5LowerLimitEnabled[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der untere Grenzwert "Reserve5" überwacht wird.

Bemerkungen

Folgende Werte werden über die Eigenschaften "Limit5LowerLimit", "Limit5LowerLimitColor" und "Limit5LowerLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

Limit5LowerLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert "Reserve5" prozentual oder absolut ausgewertet wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit5LowerLimitRelative[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional.

TRUE: Der untere Grenzwert "Reserve5" wird prozentual ausgewertet.

FALSE: Der untere Grenzwert "Reserve5" wird absolut ausgewertet.

Limit5UpperLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den oberen Grenzwert für "Reserve5" fest.

Die Eigenschaft "Limit5UpperLimitEnabled" muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve5" überwacht werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit5UpperLimit[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den oberen Grenzwert für "Reserve5" festlegt.

Bemerkungen

Die Eigenschaft "TypeLimitHigh5" legt fest, ob das Objekt prozentual oder absolut ausgewertet wird.

Limit5UpperLimitColor (RT Professional)

Beschreibung

Legt die Farbe für den oberen Grenzwert "Reserve5" fest.

Die Eigenschaft "Limit5UpperLimitEnabled" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit5UpperLimitColor[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den oberen Grenzwert "Reserve5" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Limit5UpperLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der obere Grenzwert "Reserve5" überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.Limit5UpperLimitEnabled[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der obere Grenzwert "Reserve5" überwacht wird.

Limit5UpperLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der obere Grenzwert "Reserve5" prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Limit5UpperLimitRelative**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional.

TRUE: Der obere Grenzwert "Reserve5" wird prozentual ausgewertet.

FALSE: Der obere Grenzwert "Reserve5" wird absolut ausgewertet.

LineColor (RT Professional)

Beschreibung

Legt die Farbe für die Fenstertrennlinien fest. Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LineColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl

- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für die Fenstertrennlinien festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

LineEndShapeStyle (RT Professional)

Beschreibung

Legt die Gestalt der Linienenden fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LineEndShapeStyle**[=LineEndShapeStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- CircularArc
- ComboBox
- Ellipse
- EllipseSegment
- EllipticalArc
- GraphicIOField
- GraphicView
- IOField
- Line

- ListBox
- MultiLineEdit
- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- SymbolicIOField
- TextField
- WindowsSlider
- Switch

Bei folgenden Ausprägungen haben Sie keinen Zugriff in Runtime:

- Switch
- TubeArcObect

LineEndShapeStyle

Optional. Ein Wert oder eine Konstante, welche die Gestalt der Linienenden festlegt.

LineWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Linienstärke des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LineWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- CircularArc
- EllipticalArc
- FunctionTrendControl
- Line
- OnlineTableControl
- OnlineTrendControl
- Polyline

- TrendRulerControl
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- UserArchiveControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Polygon

Int32

Optional. Ein Wert oder eine Konstante, welche die Linienstärke in Pixel festlegt.

LoadDataImmediately (RT Professional)

Beschreibung

Legt fest, ob bei einem Bildaufschlag die Variablenwerte für den darzustellenden Zeitbereich aus den Archiven geladen werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LoadDataImmediately**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl

BOOLEAN

Optional. TRUE, wenn bei einem Bildaufschlag die Variablenwerte für den darzustellenden Zeitbereich aus den Archiven geladen werden.

LocalCursor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

LockSquaredExtent (RT Professional)

Beschreibung

Legt fest, ob sich die Größe der Uhr mit der Maus einstellen lässt.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LockSquaredExtent**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock
- Gauge

BOOLEAN

Optional. TRUE, wenn sich die Größe der Uhr durch Ziehen mit der Maus an den Markierungspunkten in jedem gewünschten Seitenverhältnis einstellen lässt.

Logging (RT Professional)

Beschreibung

Gibt ein Objekt vom Typ "Logging" zurück.
Zugriff in Runtime: Lesen

Syntax

Object.**Logging**

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

Siehe auch

HMIRuntime (Seite 1313)

LogOperation (RT Professional)

Beschreibung

Legt fest, ob nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LogOperation**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- CheckBox
- ComboBox
- IOField
- ListBox
- OptionGroup
- SymbolicIOField
- WindowsSlider

BOOLEAN

Optional. TRUE, wenn nach einer Bedienung dieses Objekts eine Meldung an das Meldesystem ausgegeben wird.

LongTermArchiveConsistency (RT Professional)

Beschreibung

Legt fest, wie Meldungen in der Langzeitarchivliste angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LongTermArchiveConsistency**[= BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

Wert	Erklärung
TRUE	Die 1000 jüngsten Meldungen werden auf dem Client von allen Servern oder redundantem Serverpaar in der Langzeitarchivliste angezeigt
FALSE	1000 Meldungen werden in der Langzeitarchivliste auf dem Einzelplatz, auf dem Server oder auf dem Client pro Server bzw. pro redundantem Serverpaar angezeigt.

LowerLimit (RT Professional)

Beschreibung

Legt den unteren Grenzwert für Eingabewerte fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LowerLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den unteren Grenzwert für Eingabewerte festlegt.

LowLimitColor (RT Professional)

Beschreibung

Legt die Farbe der unteren oder linken Schaltfläche zum Scrollen in einer Bildlaufleiste fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LowLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- WindowsSlider

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der unteren oder linken Schaltfläche zum Scrollen in einer Bildlaufleiste festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

3.5.5.9 Eigenschaften M-N (Panels, Comfort Panels, RT Advanced, RT Professional)

MachineName (RT Advanced)

Beschreibung

Legt die Netzwerkennung des zu überwachenden Geräts fest.

Geben Sie als Netzwerkennung den Namen oder den Port des Geräts ein.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MachineName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SmartClientView

STRING

Optional. Ein Wert oder eine Konstante, welche die Netzwerkennung enthält.

MarginToBorder (RT Professional)

Beschreibung

Legt die Breite des 3D-Rahmens in Pixel fest. Der Wert für die Breite ist abhängig von der Größe des Objekts.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MarginToBorder**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- WindowsSlider

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des 3D-Rahmens in Pixel festlegt.

MaximumValue (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Maximalwert der Skala im angegebenen Objekt fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MaximumValue**[=DOUBLE | Int32 | SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Gauge
- Slider
- WindowsSlider

DOUBLE | Int32 | SINGLE

Optional. Ein Wert oder eine Konstante, welche den Maximalwert festlegt. Der Datentyp ist abhängig von der Ausprägung:

- DOUBLE: Bar
- Int32: Slider, WindowsSlider
- SINGLE: Gauge

MenuToolBarConfig (RT Professional)

Beschreibung

Lädt die angegebene Konfigurationsdatei mit benutzerdefiniertem Menü und Symbolleisten oder gibt den Namen der Konfigurationsdatei zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MenuToolBarConfig**[=HmiObjectHandle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Screenwindow

HmiObjectHandle

Optional. Die Konfigurationsdatei mit benutzerdefiniertem Menü und Symbolleisten.

MessageBlockAlignment (RT Professional)

Beschreibung

Legt fest, wie die Inhalte des ausgewählten Meldeblocks in der Tabelle ausgerichtet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockAlignment** [=HorizontalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "AlarmControl", "TrendRulerControl", "OnlineTableControl", "UserArchiveControl".

HorizontalAlignment

Wert	Beschreibung	Erklärung
0	links	Die Inhalte des ausgewählten Meldeblocks werden linksbündig dargestellt.
1	zentriert	Die Inhalte des ausgewählten Meldeblocks werden zentriert dargestellt.
2	rechts	Die Inhalte des ausgewählten Meldeblocks werden rechtsbündig dargestellt.

Siehe auch

AlarmControl (Seite 1338)

MessageBlockAutoPrecisions (RT Professional)

Beschreibung

Legt fest, ob die Anzahl der Nachkommastellen automatisch bestimmt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockAutoPrecisions**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Die Anzahl der Nachkommastellen wird automatisch festgelegt.

FALSE: Der Wert im Feld "Dezimalstellen" ist wirksam.

MessageBlockCaption (RT Professional)

Beschreibung

Legt für den ausgewählten Meldeblock die Bezeichnung der Spaltenüberschrift in der Meldeanzeige fest. Die Option "Projekteinstellungen übernehmen" muss deaktiviert sein, um die Bezeichnungen ändern zu können. Die eingegebene Bezeichnung ist in allen Runtime-Sprachen wirksam.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockCaption**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt für den ausgewählten Meldeblock die Bezeichnung der Spaltenüberschrift in der Meldeanzeige fest.

MessageBlockCount (RT Professional)

Beschreibung

Legt die Anzahl der vorhandenen Meldblöcke fest, die für die Meldeliste und Hitliste zur Verfügung stehen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Legt die Anzahl der vorhandenen Meldblöcke fest, die für die Meldeliste und Hitliste zur Verfügung stehen.

MessageBlockDateFormat (RT Professional)

Beschreibung

Legt fest, welches Datumsformat zur Anzeige der Meldungen verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockDateFormat**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Wert	Erklärung
Automatisch	Das Datumsformat wird automatisch bestimmt.
dd.MM.yy	Tag.Monat.Jahr, z.B. 24.12.10.
dd.MM.yyyy	Tag.Monat.Jahr, z.B. 24.12.2010.
dd/MM/yy	Tag/Monat/Jahr, z.B. 24/12/10.
dd/MM/yyyy	Tag/Monat/Jahr, z.B. 24/12/2010.

MessageBlockExponentialFormat (RT Professional)

Beschreibung

Legt fest, ob die Werte des ausgewählten Meldeblocks in Exponentialdarstellung angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockExponentialFormat**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Die Werte werden in Exponentialdarstellung angezeigt.

FALSE: Die Werte werden in Dezimaldarstellung angezeigt.

MessageBlockFlashOn (RT Professional)

Beschreibung

Legt fest, ob der Inhalt des gewählten Meldeblocks beim Erscheinen einer Meldung in Runtime blinkt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockFlashOn**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Der Inhalt des Meldeblocks blinkt.

FALSE: Der Inhalt des Meldeblocks blinkt nicht.

MessageBlockHideText (RT Professional)

Beschreibung

Legt fest, ob der Inhalt des ausgewählten Meldeblocks als Text angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockHideText**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Der Inhalt wird nicht als Text angezeigt. Die Option ist nicht aktiviert.

FALSE: Der Inhalt wird als Text angezeigt. Die Option ist aktiviert.

MessageBlockHideTitleText (RT Professional)

Beschreibung

Legt fest, ob die Überschrift des ausgewählten Meldeblocks als Text angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockHideTitleText**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Die Überschrift wird nicht als Text angezeigt. Die Option ist nicht aktiviert.

FALSE: Die Überschrift wird als Text angezeigt. Die Option ist aktiviert.

MessageBlockId (RT Professional)

Beschreibung

Legt die Zuordnung von Ident-Nummer und Meldeblock in der Meldeanzeige fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockId**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Legt die Zuordnung von Ident-Nummer und Meldeblock in der Meldeanzeige fest.

MessageBlockIndex (RT Professional)

Beschreibung

Referenziert einen vorhandenen Meldeblock. Unter Verwendung des Attributs können Sie einem bestimmten Meldeblock die Werte anderer Attribute zuweisen. Gültige Werte für "MessageBlockIndex" liegen zwischen 0 und "MessageBlockCount" minus 1.

Das Attribut "MessageBlockIndex" ist über das Attribut **MessageBlockRepos** dynamisierbar.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Referenziert einen vorhandenen Meldeblock. Gültige Werte für "MessageBlockIndex" liegen zwischen 0 und "MessageBlockCount" minus 1.

MessageBlockLeadingZeros (RT Professional)

Beschreibung

Legt fest, ob das Format des ausgewählten Meldeblocks mit führenden Nullen angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockLeadingZeros**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

TRUE: Führende Nullen werden angezeigt. Die Anzahl der führenden Nullen legen Sie in den Eigenschaften fest.

FALSE: Führende Nullen werden nicht angezeigt.

MessageBlockLength (RT Professional)

Beschreibung

Legt für den Inhalt des gewählten Meldeblocks die Länge in Zeichen fest. Ein Wert kann nur eingegeben werden, wenn das Feld "Projekteinstellungen übernehmen" nicht aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockLength**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Ein Wert, der die Länge des Meldeblocks festlegt.

MessageBlockName (RT Professional)

Beschreibung

Legt den Namen für den ausgewählten Meldeblock fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

STRING

Optional. Legt den Namen für den ausgewählten Meldeblock fest.

MessageBlockPrecisions (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen der Werte des ausgewählten Meldeblocks fest. Sie können den Wert nur eingeben, wenn die Option "Automatisch" deaktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockPrecisions**[=Int16]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int16

Optional. Legt die Anzahl der Nachkommastellen fest.

MessageBlockSelected (RT Professional)

Beschreibung

Vorhandene Meldeblöcke sind Blöcke, die zur Verwendung im Control in Runtime für die Meldeliste oder Hitliste zur Verfügung stehen.

Auf der Registerkarte "Meldeblöcke" aktivieren Sie die Blöcke der vorhandenen Meldeblöcke, die Sie im Control benötigen. Auf den Registerkarten "Hitliste" und "Meldeliste" projizieren Sie die Hitliste und Meldeliste aus den vorhandenen Blöcken.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockSelected**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

Optional. Auf der Registerkarte "Meldeblöcke" aktivieren Sie die Blöcke der vorhandenen Meldeblöcke, die Sie im Control benötigen. Auf den Registerkarten "Hitliste" und "Meldeliste" projizieren Sie die Hitliste und Meldeliste aus den vorhandenen Blöcken.

MessageBlockShowDate (RT Professional)

Beschreibung

Legt fest, ob im Meldeblock "Uhrzeit" neben der Uhrzeit auch das Datum angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockShowDate**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Das Datum und die Uhrzeit werden angezeigt.

FALSE: Die Uhrzeit wird angezeigt.

MessageBlockShowIcon (RT Professional)

Beschreibung

Legt fest, ob der Inhalt des ausgewählten Meldeblocks als Symbol angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockShowIcon**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Der Inhalt wird als Symbol angezeigt.

FALSE: Der Inhalt wird nicht als Symbol angezeigt.

MessageBlockShowTitleIcon (RT Professional)

Beschreibung

Legt fest, ob die Überschrift des ausgewählten Meldeblocks als Text angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockShowTitleIcon**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Die Überschrift wird als Symbol angezeigt.

FALSE: Die Überschrift wird nicht als Symbol angezeigt.

MessageBlockTextId (RT Professional)

Beschreibung

Legt die Bezeichnung für den gewählten Meldeblock mit Hilfe einer Text-Identnummer fest, die der Text-Library entnommen wurde. Beim Wechsel der Runtime-Sprache wird dann die Bezeichnung automatisch angepasst.

Um eine Text-Identnummer einzugeben, müssen Sie die Option "Projekteinstellungen übernehmen" deaktivieren.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockTextId**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Int32

Optional. Legt die Bezeichnung für den gewählten Meldeblock mit Hilfe einer Text-Identnummer fest.

MessageBlockTimeFormat (RT Professional)

Beschreibung

Legt fest, welches Zeitformat bzw. Zeitdauerformat zur Anzeige der Meldungen verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageBlockTimeFormat**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Folgende Zeitformate stehen zur Verfügung:

Wert	Erklärung
Automatisch	Das Zeitformat wird automatisch bestimmt.
HH:mm:ss	Stunden:Minuten:Sekunden, z.B. 15:35:44
HH:mm:ss.ms	Stunden:Minuten:Sekunden.Millisekunden, z.B. 15:35:44.240.

Wert	Erklärung
hh:mm:ss tt	Stunden:Minuten:Sekunden AM/PM, z.B. 03:35:44 PM.
hh:mm:ss.ms tt	Stunden:Minuten:Sekunden.Millisekunden AM/PM, z.B. 03:35:44.240 PM.

Folgende Zeitdauerformate stehen zur Verfügung:

Wert	Erklärung
Automatisch	Das Zeitdauerformat wird automatisch bestimmt.
d H:mm:ss	Tag Stunden:Minuten:Sekunden, z.B. 1 2:03:55.
H:mm:ss.	Stunden:Minuten:Sekunden, z.B. 26:03:55.
m:ss	Minuten:Sekunden, Beispiel: 1563:55.
s	Sekunden, z.B. 93835.

MessageBlockType (RT Professional)

Beschreibung

Gibt die Anzahl der vorhandenen Meldblöcke an, die für die Meldeliste und Hitliste zur Verfügung stehen.

Zugriff in Runtime: Lesen

Syntax

Object.**MessageBlockType**[=AlarmBlockType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

AlarmBlockType

Wert	Beschreibung	Erklärung
0	Systemblock	Der Meldeblock gehört zu den Systemblöcken
1	Textblock	Der Meldeblock gehört zu den Anwendertextblöcken.
2	Prozesswertblock	Der Meldeblock gehört zu den Prozesswertblöcken.
3	Hitlistenblock	Der Meldeblock gehört zu den Meldeblöcken der Hitliste.

MessageColumnAdd (RT Professional)

Beschreibung

Übernimmt den markierten Meldeblock aus der Liste der vorhandenen Meldeblöcke in die Liste der gewählten Meldeblöcke.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Übernimmt den markierten Meldeblock aus der Liste der vorhandenen Meldeblöcke in die Liste der gewählten Meldeblöcke

MessageColumnCount (RT Professional)

Beschreibung

Legt die Anzahl der gewählten Meldeblöcke fest, die in der Meldeliste in Runtime angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Legt die Anzahl der gewählten Meldeblöcke fest, die in der Meldeliste in Runtime angezeigt werden.

MessageColumnIndex (RT Professional)

Beschreibung

Referenziert einen für die Meldeliste gewählten Meldeblock. Unter Verwendung des Attributs können Sie einem bestimmten Meldeblock der Meldeliste die Werte anderer Attribute zuweisen.

Gültige Werte für "MessageColumnIndex" liegen zwischen 0 und "MessageColumnCount" minus 1. Das Attribut "MessageColumnIndex" ist über das Attribut **MessageColumnRepos** dynamisierbar.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Referenziert einen für die Meldeliste gewählten Meldeblock. Gültige Werte für "MessageColumnIndex" liegen zwischen 0 und "MessageColumnCount" minus 1.

MessageColumnName (RT Professional)

Beschreibung

Legt den Namen des Meldeblocks der Meldeliste fest, der mit der Eigenschaft "MessageColumnIndex" referenziert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt den Namen des Meldeblocks der Meldeliste fest, der mit der Eigenschaft "MessageColumnIndex" referenziert wird.

MessageColumnRemove (RT Professional)

Beschreibung

Entfernt den markierten Meldeblock aus der Liste der gewählten Meldeblöcke und fügt ihn in die Liste der vorhandenen Meldeblöcke ein.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Entfernt den markierten Meldeblock aus der Liste der gewählten Meldeblöcke und fügt ihn in die Liste der vorhandenen Meldeblöcke ein.

MessageColumnRepos (RT Professional)

Beschreibung

Legt die Reihenfolge der Meldeblöcke fest. "Auf" und "Ab" bewegen den ausgewählten Meldeblock in der Liste nach oben oder unten. Dadurch wird in Runtime der Meldeblock in der Meldeanzeige weiter vorne oder hinten platziert.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnRepos**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Ein Wert, der die Reihenfolge der Meldeblöcke in der Liste festlegt.

MessageColumnSort (RT Professional)

Beschreibung

Legt fest, wie der im "MessageColumnIndex" referenzierte Meldeblock sortiert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnSort**[=SortMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

SortMode

0: Keine Sortierung.

1: Aufsteigende Sortierung vom kleinsten zum größten Wert

2: Absteigende Sortierung vom größten zum kleinsten Wert

MessageColumnSortIndex (RT Professional)

Beschreibung

Legt die Sortierreihenfolge des im "MessageColumnIndex" referenzierten Meldeblocks fest. Wenn Sie den Wert auf "0" setzen, wird das Sortierkriterium in "MessageColumnSort" entfernt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnSortIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Legt die Sortierreihenfolge des im "MessageColumnIndex" referenzierten Meldeblocks fest.

MessageColumnVisible (RT Professional)

Beschreibung

Legt fest, ob Meldeblöcke in der Meldeanzeige angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageColumnVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

Optional.

TRUE: Meldeblöcke werden in der Meldeanzeige angezeigt.

FALSE: Meldeblöcke werden nicht in der Meldeanzeige angezeigt.

MessageListType (RT Professional)

Beschreibung

Legt fest, aus welcher Meldeliste Inhalte beim Bildaufschlag angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MessageListType**[=AlarmListType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

AlarmListType

Wert	Beschreibung	Erklärung
0	Meldeliste	Bei einem Bildaufschlag werden die aktuell anstehenden Meldungen dargestellt.
1	Kurzzeitarchivliste	Bei einem Bildaufschlag wird eine Kurzzeitarchivliste mit den archivierten Meldungen dargestellt. Bei neu eingehenden Meldungen wird die Anzeige sofort aktualisiert.
2	Langzeitarchivliste	Bei einem Bildaufschlag wird eine Langzeitarchivliste mit den archivierten Meldungen dargestellt.
3	Sperrliste	Bei einem Bildaufschlag werden nur die aktuell gesperrten Meldungen dargestellt.
4	Hitliste	Bei einem Bildaufschlag werden die projizierten statistischen Informationen dargestellt.
5	Liste auszublender Meldungen	Bei einem Bildaufschlag werden die auszublender Meldungen dargestellt.

MinimumValue (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Minimalwert der Skala im angegebenen Objekt fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MinimumValue**[=DOUBLE | Int32 | SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Gauge
- Slider
- WindowsSlider

DOUBLE | Int32 | SINGLE

Optional. Ein Wert oder eine Konstante, welche den Minimalwert festlegt. Der Datentyp ist abhängig von der Ausprägung:

- DOUBLE: Bar
- Int32: Slider, WindowsSlider
- SINGLE: Gauge

MinuteNeedleHeight (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Länge des Minutenzeigers im Objekt "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MinuteNeedleHeight**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Int32

Optional. Ein Wert oder eine Konstante, welche die Länge des Minutenzeigers festlegt.

Geben Sie die Länge des Minutenzeigers in Prozent an, bezogen auf den Radius des Zifferblattes.

MinuteNeedleWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Breite des Minutenzeigers im Objekt "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MinuteNeedleWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des Minutenzeigers festlegt.

Geben Sie die Breite in Prozent an, bezogen auf die doppelte Länge des Minutenzeigers.

Mode (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Feldtyp des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**Mode**[=IOFieldType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- IOField
- RoundButton
- SymbolicIOField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- DateTimeField
- GraphicIOField
- Switch

IOFieldType

hmiIOFieldInput (1): Eingabefeld

hmiIOFieldOutput (0): Ausgabefeld

hmiIOFieldInOut (2): Eingabe- und Ausgabefeld

Moveable (RT Professional)

Beschreibung

Legt fest, ob das Fenster in Runtime bewegt werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Moveable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn das Fenster in Runtime bewegt werden kann.

Siehe auch

AlarmControl (Seite 1338)

UserArchiveControl (Seite 1528)

TrendRulerControl (Seite 1508)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

OnlineTrendControl (Seite 1443)

MsgFilterSQL (RT Professional)

Beschreibung

Legt für die benutzerdefinierte Selektion der Meldungen ein oder mehrere SQL-Statements fest. Mehrere benutzerdefinierte Selektionen werden mit "OR" verbunden. Wenn Sie eine feste Selektion "DefaultMsgFilterSQL" projektiert haben, werden die SQL-Statements von "DefaultMsgFilterSQL" und "MsgFilterSQL" mit "AND" verbunden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**MsgFilterSQL**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Ein Wert oder eine Konstante, welche die SQL-Statements für benutzerdefinierte Selektion der Meldungen festlegt.

Name (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt den Objektnamen als STRING zurück. Der zurückgegebene Wert ist abhängig vom verwendeten Objekt.

Zugriff in Runtime: Lesen

Syntax

Object.**Name**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- BatteryView
- Clock
- Ellipse
- Gauge
- Slider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- AlarmControl
- AlarmView
- ApplicationWindow
- Bar
- Button
- ChannelDiagnose
- CheckBox
- Circle
- CircleSegment
- CircularArc
- ComboBox
- DateTimeField
- DiscSpaceView
- EllipseSegment
- EllipticalArc

- FunctionTrendControl
- GraphicIOField
- GraphicView
- HTML-Browser
- IOField
- Line
- ListBox
- MediaPlayer
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- PLCCodeViewer
- Polygon
- Polyline
- ProtectedAreaNameView
- RangeLabelView
- RangeQualityView
- RecipeView
- Rectangle
- RoundButton
- S7GraphOverview
- Screenwindow
- SmartClientView
- StatusForce
- Switch
- SymbolLibrary
- SymbolicIOField
- SysDiagControl
- TextField
- TrendRulerControl
- TrendView
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject

- Tubepolyline
- UserArchiveControl
- UserView
- WlanQualityView
- WindowsSlider
- ZoneLabelView
- ZoneQualityView

Bemerkungen

Abhängig vom angegebenen Objekt werden folgende Objektnamen zurückgeben:

- Tag: Name der Variable ohne Server- und Variablenpräfix.
- Project: Name des aktuellen Runtime-Projekts.
- Dataltem: Name des Dataltem-Objekts.
- Layer: Name der Ebene.
- FunctionTrendControl : Name der durch die Eigenschaft "Index" referenzierte Kurve.

Hinweis

In der Auflistung "Tags" können Sie eine Variable über die Eigenschaft "Name" ansprechen. Variablennamen sind in WinCC nach folgendem Schema aufgebaut:

<Variablenpräfix><Name der Variable>

Wenn Sie nur den Variablennamen angeben, wird das Variablenpräfix aus dem Kontext des Bildfensters übernommen.

Beispiel

Das folgende Beispiel gibt den Namen des aktuellen Runtime-Projekts als Trace aus:

```
'VBS160  
HMIRuntime.Trace "Name: " & HMIRuntime.ActiveProject.Name & vbNewLine
```

NeedleBorderColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Linienfarbe der Zeiger im Objekt "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**NeedleBorderColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Color

Optional. Ein Wert oder eine Konstante, welche die Linienfarbe des Zeigers festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

NeedleColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Zeigerfarbe im Objekt "Clock" fest.

Die Eigenschaft "NeedleFillStyle" müssen Sie als "Transparent" projiziert haben, damit die Zeigerfarbe dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**NeedleColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Color

Optional. Ein Wert oder eine Konstante, welche die Zeigerfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

NeedleFillStyle (RT Professional)

Beschreibung

Legt fest, ob die Zeiger gefüllt oder transparent dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**NeedleFillStyle**[=THmiFillStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

THmiFillStyle

hmiFillStyleTransparent (65536): Die Zeiger werden ausgefüllt in Zeigerfüllfarbe und mit Rand in Vordergrundfarbe dargestellt.

hmiFillStyleSolid (0): Die Zeiger erscheinen transparent und werden durch einen Rand in Vordergrundfarbe dargestellt.

NormalRangeColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Normalbereichs auf der Skala des Objekts "Gauge" fest.

Die Eigenschaft "NormalRangeVisible" muss den Wert TRUE haben, damit der Normalbereich angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**NormalRangeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Normalbereichs festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

NormalRangeVisible (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Normalbereich in der Skala des Objekts "Gauge" angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**NormalRangeVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

BOOLEAN

Optional. TRUE, wenn der Normalbereich in der Skala angezeigt wird.

Bemerkungen

Die Farbe des Normalbereichs legen Sie mit der Eigenschaft "NormalRangeColor" fest.

3.5.5.10 Eigenschaften O-P (Panels, Comfort Panels, RT Advanced, RT Professional)

Object (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Wenn Sie ein nicht von WinCC bereitgestelltes Control verwenden, kann es vorkommen, dass die vom Control mitgebrachten Eigenschaften namensgleich sind mit den allgemeinen ScreenItem-Eigenschaften. In diesem Fall haben die ScreenItem-Eigenschaften Vorrang. Auf die "verdeckten" Eigenschaften eines Fremdanbieter-Controls kann über die zusätzliche Eigenschaft "object" zugegriffen werden.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.Object

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- AlarmView
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- RecipeView
- SmartClientView
- StatusForce
- SymbolLibrary
- UserArchiveControl

Anwendungsbeispiel

Sprechen Sie die Eigenschaften eines Fremdanbieter-Controls in folgender Form an:

Control.object.type

Wenn Sie nur die Form Control.type verwenden, werden bei Namensgleichheit die Eigenschaften des ScreenItem-Objekts verwendet.

ObjectSizeDeclutteringEnable (RT Professional)

Beschreibung

Gibt an, ob nur Objekte innerhalb eines definierten Größenbereichs angezeigt werden.

Zugriff in Runtime: Lesen

Syntax

Object.ObjectSizeDeclutteringEnable

Object

Erforderlich. Ein Objekt vom Typ "Screen".

Beispiel

Das Beispiel gibt die Decluttering-Eigenschaften des Bildes "NewPDL1" als Trace aus.

```
'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
```

Siehe auch

Screen (Seite 1320)

ObjectSizeDeclutteringMax (RT Professional)

Beschreibung

Gibt den oberen Größenbereich für die Anzeigeunterdrückung von Objekten des angegebenen Bildes als LONG zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**ObjectSizeDeclutteringMax**

Object

Erforderlich. Ein Objekt vom Typ "Screen".

Beispiel

Das Beispiel gibt die Decluttering-Eigenschaften des Bildes "NewPDL1" als Trace aus.

```
'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
```

Siehe auch

Screen (Seite 1320)

ObjectSizeDeclutteringMin (RT Professional)

Beschreibung

Gibt den unteren Größenbereich für die Anzeigeunterdrückung von Objekten des angegebenen Bildes als LONG zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**ObjectSizeDeclutteringMin**

Object

Erforderlich. Ein Objekt vom Typ "Screen".

Beispiel

Das Beispiel gibt die Decluttering-Eigenschaften des Bildes "NewPDL1" als Trace aus.

```
'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
```

Siehe auch

Screen (Seite 1320)

OCXState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

Online (RT Professional)

Beschreibung

Legt das Starten und Stoppen der Aktualisierung fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Online**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl
- FunctionTrendControl
- OnlineTableControl

BOOLEAN

Optional.

TRUE: Die aktualisierte Darstellung wird angehalten. Die Werte werden zwischengespeichert und bei erneuter Aktivierung der Schaltfläche nachgetragen.

FALSE: Die aktualisierte Darstellung wird fortgesetzt.

OperationSteps (RT Professional)

Beschreibung

Legt fest, um wie viele Schritte der Schieber der Bildlaufleiste mit einem Mausklick verschoben wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperationSteps**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- WindowSlider

Int32

Optional. Ein Wert oder eine Konstante, welche festlegt, um wie viele Schritte der Schieber der Bildlaufleiste mit einem Mausklick verschoben wird.

OperatorMessageld (RT Professional)

Beschreibung

Legt die Zuordnung von Ident-Nummer und Auslöseereignis in der Meldeanzeige fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageId**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Wert	Beschreibung	Erklärung
0	Lock	Auslöseereignis "Sperrern"
1	Unlock	Auslöseereignis "Freigeben"
2	Hide	Auslöseereignis "Ausblenden"
3	Unhide	Auslöseereignis "Einblenden"
4	Quit	Auslöseereignis "Quittieren"

OperatorMessageIndex (RT Professional)

Beschreibung

Referenziert das Meldeereignis für eine Bedienmeldung. Bei Verwendung der Eigenschaft können Sie einer bestimmten Bedienmeldung die Werte anderer Eigenschaften zuweisen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Wert	Erklärung
0	Meldeereignis "Sperrern"
1	Meldeereignis "Freigeben"
2	Meldeereignis "Ausblenden"
3	Meldeereignis "Einblenden"
4	Meldeereignis "Quittieren"

OperatorMessageName (RT Professional)

Beschreibung

Legt bei Meldeereignissen für Bedienmeldungen den Namen fest, der mit dem Ereignis "OperatorMessageIndex" referenziert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Wert	Erklärung
Lock	Das Meldeereignis "Sperrern".
Unlock	Das Meldeereignis "Freigeben".
Hide	Das Meldeereignis "Ausblenden".
Unhide	Das Meldeereignis "Einblenden".
Quit	Das Meldeereignis "Quittieren".

OperatorMessageNumber (RT Professional)

Beschreibung

Legt eine Meldenummer für die Bedienmeldung des ausgewählten Meldeereignisses fest, wenn *nicht* die Bedienmeldung von WinCC verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageNumber**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Legt eine Meldenummer für die Bedienmeldung des ausgewählten Meldeereignisses fest.

OperatorMessageSelected (RT Professional)

Beschreibung

Aktiviert die Meldeereignisse in der Liste, bei denen Bedienmeldungen ausgelöst werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSelected**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

Optional. Aktiviert die Meldeereignisse in der Liste, bei denen Bedienmeldungen ausgelöst werden.

OperatorMessageSource1 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 1" der hier projektierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z. B. "Motor gestört", soll im "Prozesswertblock 1" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "1" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource1**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 1" der hier projektierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource2 (RT Professional)

Beschreibung

Legt einen einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 2" der hier projektorierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z. B. "Motor gestört", soll im "Prozesswertblock 2" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "2" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource2**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 2" der hier projektorierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource3 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 3" der hier projektorierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z. B. "Motor gestört", soll im "Prozesswertblock 3" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "3" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource3**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 3" der hier projizierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource4 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 4" der hier projizierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 4" der gesperrten Meldung, z. B. "Motor gestört", soll im "Prozesswertblock 4" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "4" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource4**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 4" der hier projizierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource5 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 5" der hier projizierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z.B. "Motor gestört", soll im "Prozesswertblock 5" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "5" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource5**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 5" der hier projektierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource6 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 6" der hier projektierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z.B. "Motor gestört", soll im "Prozesswertblock 6" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "6" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource6**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 6" der hier projektierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource7 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 7" der hier projektierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z.B. "Motor gestört", soll im "Prozesswertblock 7" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "7" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.OperatorMessageSource7[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 7" der hier projektierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource8 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 8" der hier projektierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z.B. "Motor gestört", soll im "Prozesswertblock 8" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "8" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.OperatorMessageSource8[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 8" der hier projektierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource9 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 9" der hier projektierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z.B. "Motor gestört", soll im

"Prozesswertblock 9" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "9" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource9**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 9" der hier projektierten Bedienmeldung hinzugefügt wird.

OperatorMessageSource10 (RT Professional)

Beschreibung

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 10" der hier projektierten Bedienmeldung hinzugefügt wird.

Zum Beispiel wollen Sie beim Sperren einer Meldung eine Bedienmeldung erzeugen. Der Inhalt von "Anwendertextblock 1" der gesperrten Meldung, z.B. "Motor gestört", soll im "Prozesswertblock 10" der Bedienmeldung angezeigt werden. Wählen Sie dazu unter Prozesswert "10" als Meldeblock der bedienten Meldung "Anwendertextblock 1" aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**OperatorMessageSource10**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Legt einen Meldeblock der bedienten Meldung fest, der zum "Prozesswertblock 10" der hier projektierten Bedienmeldung hinzugefügt wird.

OperatorMessageSourceType1 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType1[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType2 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType2[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType3 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType1[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType4 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType4[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType5 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType5[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType6 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType6[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType7 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.**OperatorMessageType7**[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType8 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.**OperatorMessageType8**[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType9 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType9[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

OperatorMessageSourceType10 (RT Professional)

Beschreibung

Legt fest, in welchem Format der Inhalt der Quelle übergeben wird.

Syntax

Object.OperatorMessageType10[=Type]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Type

Wert	Beschreibung	Erklärung
0	Text	Der Inhalt der Quelle wird als Text übergeben.
1	Wert	Der Inhalt der Quelle wird als Wert übergeben.

Siehe auch

AlarmControl (Seite 1338)

PageMode (RT Professional)

Beschreibung

Legt fest, ob das Blättern in der Langzeitarchivliste möglich ist. Damit können Sie alle Meldungen des Umlaufarchivs seitenweise in der Langzeitarchivliste anzeigen. Über die Eigenschaft "PageModeMessageNumber" bestimmen Sie die Anzahl der Meldungen, die pro Seite angezeigt werden.

Wenn das Blättern aktiviert ist, können Sie die Schaltflächen zum Blättern in der Symbolleiste verwenden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PageMode**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Das Blättern in der Langzeitarchivliste ist möglich.

FALSE: Das Blättern in der Langzeitarchivliste ist nicht möglich.

PageModeMessageNumber (RT Professional)

Beschreibung

Legt die Anzahl der Meldungen fest, die beim Blättern in der Langzeitarchivliste pro Seite angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PageModeMessageNumber**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Meldungen pro Seite festlegt.

Path (RT Professional)

Beschreibung

Gibt den Pfad des aktuellen Projekts ohne den Dateinamen als STRING zurück. Auf einen WinCC-Client ohne eigenes Projekt wird der Pfad im UNC-Format zurückgeliefert, sonst der lokale Pfad.

Zugriff in Runtime: Lesen

Syntax

Object.**Path**

Object

Erforderlich. Ein Objekt vom Typ "Project".

Beispiel

Das folgende Beispiel gibt den Projektpfad als Trace aus:

```
'VBS161  
HMIRuntime.Trace "Path: " & HMIRuntime.ActiveProject.Path & vbNewLine
```

Siehe auch

Project (Seite 1319)

Password (RT Advanced)

Beschreibung

Legt das Passwort für den Aufbau der Fernüberwachung fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Password**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SmartClientView

STRING

Optional. Ein Wert oder eine Konstante, die das Passwort für den Aufbau der Fernüberwachung enthält.

PicAlignment (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Ausrichtung innerhalb des angegebenen Objekts fest.

Picture (RT Professional)

Beschreibung

Legt das Bild fest, das im Grafik-Objekt in Runtime angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Picture**[=Image]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- GraphicView
- Clock

Image

Optional. Ein Wert oder eine Konstante, die das Bild festlegt, das im Grafik-Objekt in Runtime angezeigt wird.

Bemerkungen

Das Bild (*.BMP oder *.DIB) muss sich im Ordner "GraCS" des aktuellen Projekts befinden, damit Sie es einbinden können.

PictureAlignment (RT Professional)

Beschreibung

Legt die Darstellungsart des Hintergrundbildes im Prozessbild fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PictureAlignment**[=PictureAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton

PictureAlignment

Optional. Ein Wert oder eine Konstante, welche die Darstellungsart des Hintergrundbildes im Prozessbild festlegt.

PictureDeactivated (RT Professional)

Beschreibung

Legt die Grafik fest, das im Zustand "Deaktiviert" angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

`Object.PictureDeactivated[=Image]`

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Roundbutton

Image

Optional. Ein Wert oder eine Konstante, welche die Grafik festlegt, die im Zustand "Deaktiviert" angezeigt wird.

Bemerkungen

Die Grafik muss sich im Ordner "GraCS" des aktuellen Projekts befinden, um es einzubinden.

PictureOff (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Grafik fest, die im Zustand "Aus" angezeigt wird.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**PictureOff**[=HmiObjectHandle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- GraphicIOField
- RoundButton

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Switch

HmiObjectHandle

Optional. Ein Wert oder eine Konstante, welche die Grafik festlegt, die im Zustand "Aus" angezeigt wird.

Bemerkungen

Die Grafik (*.BMP oder *.DIB) muss sich im Ordner "GraCS" des aktuellen Projekts befinden, damit Sie es einbinden können.

PictureOn (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt das Bild fest, das im Zustand "Ein" angezeigt wird.

Zugriff in Runtime:

- RT Advanced: Kein Zugriff
- RT Professional: Lesen und Schreiben

Syntax

Object.**PictureOn**[=HmiObjectHandle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- GraphicIOField
- RoundButton

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Switch

HmiObjectHandle

Optional. Ein Wert oder eine Konstante, die das Bild festlegt, das im Zustand "Ein" angezeigt wird.

Bemerkungen

Das Bild (*.BMP oder *.DIB) muss sich im Ordner "GraCS" des aktuellen Projekts befinden, damit Sie es einbinden können.

PointerColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Zeigerfarbe des Objekts "Gauge" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PointerColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Zeigerfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

PointsCount (RT Professional)

Beschreibung

Legt die Anzahl der Eckpunkte der Polylinie oder des Vielecks fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PointsCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Polygon
- Polyline
- Tubepolyline

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Line

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Eckpunkte der Polylinie festlegt.

Precision (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen (0 bis 20) fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Precision**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen (0 bis 20) festlegt.

Pressed (RT Professional)

Beschreibung

Legt fest, ob das angegebene Objekt gedrückt dargestellt ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Pressed**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton

BOOLEAN

Optional. TRUE, wenn das angegebene Objekt gedrückt dargestellt ist.

ProcessValue (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Voreinstellung für den anzuzeigenden Wert fest.

In Runtime wird dieser Wert verwendet, wenn beim Start des Bildes die zugehörige Variable nicht angeschlossen oder nicht aktualisiert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ProcessValue**[=DOUBLE | Int32 | Object | SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Gauge
- GraphicIOField
- IOField
- OptionGroup
- Slider
- SymbolicIOField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Button
- CheckBox
- DateTimeField
- Switch
- SymbolLibrary

DOUBLE | Int32 | SINGLE

Optional. Ein Wert oder eine Konstante, welche den voreingestellten Wert enthält. Der Datentyp ist abhängig von der Ausprägung:

- DOUBLE: Bar
- Int32: GraphicIOField, OptionGroup, Slider, SymbolicIOField, WindowsSlider
- Object: IOField
- SINGLE: Gauge

Bemerkungen

Wenn Sie der Eigenschaft "ProcessValue" SmartTags zuweisen wollen, dann müssen Sie die Zuweisung wie folgt formulieren:

'Beispiele für die Zuweisung von SmartTags

```
'Example 1
IOField.ProcessValue = SmartTags("TagName").Value
'Example 2
HmiRuntime.Screens("Screen_1").ScreenItems("IOField_1").ProcessValue =
SmartTags("Tag_1").Value
```

ProgID (RT Professional)

Beschreibung

Bei WinCC-fremden Controls wird die versionsunabhängige ProgID als Typ zurückgeliefert.

Siehe auch

ForeignControl (Seite 1393)

3.5.5.11 Eigenschaften Q-R (Panels, Comfort Panels, RT Advanced, RT Professional)

QualityCode (RT Professional)

Beschreibung

Gibt das Maß für die Qualität eines Variablenwerts nach dem Lesen der Variablen als SHORT zurück. Nach dem Schreiben einer Variable ist der Wert ungültig.

Zugriff in Runtime: Lesen

Syntax

Object.**QualityCode**

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

Beispiel

Das folgende Beispiel gibt die Qualität des gelesenen Wertes aus, wenn beim Lesen kein Fehler aufgetreten ist:

```
'VBS83
Dim objTag
Dim lngLastErr
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
lngLastErr = objTag.LastError
If 0 = lngLastErr Then
MsgBox objTag.QualityCode
End If
```

Siehe auch

Tag (Seite 1332)

Radius (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Radius des angegebenen Objekts "Circle" fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Radius**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Circle
- CircleSegment
- CircularArc
- RoundButton

Int32

Optional. Ein Wert oder eine Konstante, welche den Radius in Pixel festlegt.

RadiusHeight (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Nebenachse des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RadiusHeight**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Ellipse
- EllipseSegment
- EllipticalArc
- TubeArcObject

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Circle

Int32

Optional. Ein Wert oder eine Konstante, welche die Nebenachse in Pixel festlegt.

RadiusWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Hauptachse des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RadiusWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Ellipse
- EllipseSegment
- EllipticalArc
- TubeArcObject

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Circle

Int32

Optional. Ein Wert oder eine Konstante, welche die Hauptachse in Pixel festlegt.

RecipeName (RT Advanced)

Beschreibung

Gibt den Namen der Rezeptur zurück, die gerade in der "Rezepturanzeige" angezeigt wird.
Zugriff in Runtime: Lesen

Syntax

Object.**RecipeName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- RecipeView

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der Rezeptur zurückgibt.

RecipeNumber (RT Advanced)

Beschreibung

Gibt die Nummer der Rezeptur zurück, die gerade in der "Rezepturanzeige" angezeigt wird.
Zugriff in Runtime: Lesen

Syntax

Object.**RecipeNumber**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- RecipeView

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer der Rezeptur zurückgibt.

RecordName (RT Advanced)

Beschreibung

Gibt den Namen des Rezepturdatensatzes zurück, der gerade in der "Rezepturanzeige" angezeigt wird.

Zugriff in Runtime: Lesen

Syntax

Object.**RecordName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- RecipeView

STRING

Optional. Ein Wert oder eine Konstante, die den Namen des Rezepturdatensatzes zurückgibt.

RecordNumber (RT Advanced)

Beschreibung

Gibt die Nummer des Rezepturdatensatzes zurück, der gerade in der "Rezepturanzeige" angezeigt wird.

Zugriff in Runtime: Lesen

Syntax

Object.**RecordNumber**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- RecipeView

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer des Rezepturdatensatzes zurückgibt.

RelativeFillLevel (RT Professional)

Beschreibung

Legt den Prozentsatz fest, zu dem das Objekt gefüllt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RelativeFillLevel**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- Circle
- CircleSegment
- Ellipse
- EllipseSegment
- GraphicView
- OptionGroup
- Polygon
- Rectangle
- RoundButton
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox

Int32

Optional. Ein Wert oder eine Konstante, welche den Prozentsatz festlegt, zu dem das Objekt gefüllt wird.

Rotation (RT Professional)

Beschreibung

Legt den Drehwinkel des angegebenen Objekts in Grad fest. Der Drehwinkel wird entgegen dem Uhrzeigersinn gemessen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Rotation**[=SymbolLibraryRotation]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolLibrary

SymbolLibraryRotation

hmiSymbolLibraryRotationNone (0): Das Objekt dreht sich um 0 Grad.

hmiSymbolLibraryRotation90Degree (90): Das Objekt dreht sich um 90 Grad.

hmiSymbolLibraryRotation180Degree (180): Das Objekt dreht sich um 180 Grad.

hmiSymbolLibraryRotation270Degree (270): Das Objekt dreht sich um 270 Grad.

RotationAngle (RT Professional)

Beschreibung

Legt den Rotationswinkel in Grad fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RotationAngle**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Line
- Polygon
- Polyline
- TextField
- TubeTeeObject

Int32

Optional. Ein Wert oder eine Konstante, welche den Rotationswinkel in Grad festlegt.

Bemerkungen

Das Objekt dreht sich in Runtime im Uhrzeigersinn um den Referenzpunkt.

RotationCenterLeft (RT Professional)

Beschreibung

Legt die X-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RotationCenterLeft**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Line
- Polygon
- Polyline
- TextField

Int32

Optional. Ein Wert oder eine Konstante, welche die X-Koordinate des Drehpunkts festlegt, um den sich das Objekt in Runtime dreht.

Bemerkungen

Der Wert der X-Koordinate ist relativ zur Objektbreite. Geben Sie den Wert ausgehend von der linken Kante des objektumfassenden Rechteckes in Prozent an.

RotationCenterTop (RT Professional)

Beschreibung

Legt die Y-Koordinate des Drehpunkts fest, um den sich das Objekt in Runtime dreht.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RotationCenterTop**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Line
- Polygon
- Polyline
- TextField

Int32

Optional. Ein Wert oder eine Konstante, welche die Y-Koordinate des Drehpunkts festlegt, um den sich das Objekt in Runtime dreht.

Bemerkungen

Der Wert der Y-Koordinate ist relativ zur Objektbreite. Geben Sie den Wert ausgehend von der oberen Kante des objektumfassenden Rechteckes in Prozent an.

RoundCornerHeight (RT Professional)

Beschreibung

Legt den Eckradius fest. Geben Sie den Wert prozentual zur halben Höhe des Objektes ein.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RoundCornerHeight**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Rectangle

Int32

Optional. Ein Wert oder eine Konstante, welche den Eckradius festlegt.

RoundCornerWidth (RT Professional)

Beschreibung

Legt den Eckradius fest. Geben Sie den Wert prozentual zur halben Breite des Objektes ein.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RoundCornerWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Rectangle

Int32

Optional. Ein Wert oder eine Konstante, welche den Eckradius festlegt.

RowScrollbar (RT Professional)

Beschreibung

Legt fest, ob Zeilen-Rollbalken angezeigt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RowScrollbar**[=ScrollbarVisibility]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

ScrollbarVisibility

Wert	Beschreibung	Erklärung
0	nein	Die Zeilen-Rollbalken werden nicht angezeigt.
1	bei Bedarf	Die Zeilen-Rollbalken werden angezeigt, wenn der Platzbedarf des Controls in horizontaler Richtung größer ist als der zur Verfügung stehende Anzeigebereich.
2	immer	Die Zeilen-Rollbalken werden immer angezeigt.

RowTitleAlignment (RT Professional)

Beschreibung

Legt fest, wie die Zeilenüberschriften ausgerichtet werden.
Folgende Einstellungen stehen zur Auswahl:

Wert	Beschreibung	Erklärung
0	links	Die Zeilenüberschriften werden linksbündig angezeigt.
1	zentriert	Die Zeilenüberschriften werden zentriert angezeigt.
2	rechts	Die Zeilenüberschriften werden rechtsbündig angezeigt.

Das Attribut ist mit dem Namen **RowTitleAlignment** dynamisierbar.

Siehe auch

AlarmControl (Seite 1338)

RTPersistence (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RTPersistence**[=RTPersistence]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- SysDiagControl
- UserView

RTPersistence

Wert	Beschreibung	Erklärung
0	verwerfen	Die aktuellen Online-Projektierungen werden beim nächsten Bildwechsel verworfen.
1	beibehalten	Die aktuellen Online-Projektierungen werden beim nächsten Bildwechsel beibehalten.
2	zurücksetzen	Alle jemals vorgenommenen Online-Projektierungen gehen verloren. Das Bild wird auf den im Konfigurationssystem vorhandenen Inhalt gesetzt.

RTPersistenceType (RT Professional)

Beschreibung

Legt fest, wie Online-Projektierungen von WinCC beibehalten werden.

Syntax

Object.**RTPersistenceType**[=RTPersistenceType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- SysDiagControl
- UserView

RTPersistenceType

Wert	Beschreibung	Erklärung
0	nicht beibehalten	Online-Projektierungen werden nicht beibehalten. Sie gehen beim nächsten Bildwechsel verloren.
1	während Runtime beibehalten	Online-Projektierungen werden während Runtime beibehalten. Sie gehen beim Beenden verloren.
2	permanent beibehalten	Online-Projektierungen werden permanent beibehalten. Sie sind auch nach einem Neustart verfügbar.

RulerColor (RT Advanced)

Beschreibung

Legt die Farbe für die Skaleneinteilung (Hilfslinie) der Achsbeschriftung im Objekt "OnlineTrendControl" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RulerColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TrendView

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Skaleneinteilung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

3.5.5.12 Eigenschaften S (Panels, Comfort Panels, RT Advanced, RT Professional)

ScaleColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Skala des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScaleColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TrendView
- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Skala festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

ScreenItem-Objekt der Ausprägung "Bar": Damit die Farbe angezeigt wird, muss die Eigenschaft "ShowScale" mit TRUE projiziert sein.

ScaleGradation (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Abstand zwischen zwei großen Teilstrichen der Skala fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScaleGradation**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den Abstand zwischen zwei großen Teilstrichen der Skala festlegt.

ScaleLabelColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Beschriftung der Skalenteilung des Objekts "Gauge" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScaleLabelColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Beschriftungsfarbe der Skalenteilung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

ScaleNumerator (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ScalePosition (RT Professional)

Beschreibung

Legt die Position der Skala des angegebenen Objekts fest. Die Eigenschaft "ShowScale" muss auf TRUE gesetzt sein, damit die Skala angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScalePosition**[=ScalePosition]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Slider

ScalePosition

hmiScalePositionLeftUp (0): Für einen vertikalen Balken wird die Skala oben angezeigt. Für einen horizontalen Balken wird die Skala links angezeigt.

hmiScalePositionRightDown (1): Für einen vertikalen Balken wird die Skala unten angezeigt. Für einen horizontalen Balken wird die Skala rechts angezeigt.

ScaleTickColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Skalenteilung des Objekts "Gauge" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScaleTickColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Skalenteilung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

ScaleTickLabelPosition (RT Professional)

Beschreibung

Legt den Durchmesser des gedachten Kreises fest, auf dem sich die Beschriftung der Skalenteilung befindet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScaleTickLabelPosition**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den Durchmesser des gedachten Kreises festlegt, auf dem sich die Beschriftung der Skalenteilung befindet.

Wertebereich 0 bis 1

0: Die Beschriftung befindet sich in der Mitte der Skalenscheibe.

1: Der Durchmesser des gedachten Kreises für die Beschriftung ist der kleinere Wert der Geometrieigenschaften "Width" und "Height". Ein Teil der Beschriftung kann dadurch außerhalb der Objektbegrenzung liegen und ist damit unsichtbar.

ScaleTickLength (RT Professional)

Beschreibung

Legt die Länge der Hauptstriche der Skalenteilung fest. Der Wert bezieht sich auf die Hälfte des kleineren Werts der Geometrieigenschaften "Width" und "Height".

Die Länge der Teilstriche für die Feinteilung ist $0,5 \cdot \text{Skalenbreite}$.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScaleTickLength**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die Länge der Hauptstriche der Skalenteilung festlegt.

Wertebereich 0 bis Skalenendwert.

0: Es ist keine Skalenteilung vorhanden. Auch die Aufteilung der Skala in Bereiche ist nicht sichtbar.

Skalenendwert: Die Skalenteilung reicht vom Mittelpunkt der Skalenscheibe bis zum Wert, der durch Skalenendwert festgelegt ist.

ScaleTickPosition (RT Professional)

Beschreibung

Legt den Durchmesser des gedachten Kreises fest, auf dem sich die Skalenteilung befindet.

Die Hauptstriche der Skalenteilung liegen dabei mit ihrem nach außen gerichteten Ende auf diesem Kreis.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScaleTickPosition**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den Durchmesser des gedachten Kreises festlegt, auf dem sich die Skalenteilung befindet.

Wertebereich 0 bis 1

0: Die Skalenteilung befindet sich in der Mitte der Skalenscheibe.

1: Der Durchmesser des gedachten Kreises für die Skalenteilung ist der kleinere Wert der Geometrieigenschaften "Width" und "Height".

ScaleDenominator (Panels, Comfort Panels, RT Advanced)

Beschreibung

Legt den Zähler für die Skalierung auf dem Client fest.

Syntax

Object.ScaleDenominator=[Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "SmartClientView".

Int

Optional. Ein Wert oder eine Konstante, die den Wert festlegt.

Siehe auch

SmartClientView (Seite 1484)

Scaling (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ScalingType (RT Professional)

Beschreibung

Legt die Art der Balkenskalierung fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ScalingType[=BarScalingType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BarScalingType

Optional. Ein Wert oder eine Konstante, welche die Art der Balkenskalierung festlegt.

- hmiBarScalingLinear (0): Linear
Die große Teilstriche sind gleichmäßig über die Skala verteilt. Der Abstand zwischen den großen Teilstrichen entspricht dem Wert des Attributs "Achsenabschnitt".
- hmiBarScalingLogarithmic (1): Logarithmisch
Die Verteilung der großen Teilstriche auf der Skala folgt einer logarithmischen Funktion. Die Darstellung niedriger Werte wird sehr stark betont.
- hmiBarScalingNegativeLogarithmic (2): Negativ logarithmisch
Die Verteilung der großen Teilstriche auf der Skala folgt einer negativen logarithmischen Funktion. Die Darstellung hoher Werte wird sehr stark betont.
- hmiBarScalingAutomatic (3): Automatisch
Die großen Teilstriche sind gleichmäßig über die Skala verteilt. Der Abstand zwischen den großen Teilstrichen wird automatisch festgelegt.
- hmiBarScalingTangent (4): Tangente
Die Verteilung der großen Teilstriche auf der Skala betont die Darstellung der niedrigen und der hohen Werte.
- hmiBarScalingQuadratic (5): Quadratisch
Die Verteilung der großen Teilstriche auf der Skala folgt einer quadratischen Funktion. Die Darstellung hoher Werte wird betont.
- hmiBarScalingCubic (6): Kubisch
Die Verteilung der großen Teilstriche auf der Skala folgt einer kubischen Funktion. Dadurch wird die Darstellung großer Werte betont.

Bemerkungen

Damit die Farbe angezeigt wird, muss die Eigenschaft "ShowScale" den Wert TRUE haben.

Siehe auch

Bar (Seite 1353)

ScreenItems (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt die ScreenItems-Auflistung zurück.

Zugriff in Runtime: Lesen

Syntax

Object.ScreenItems

Object

Erforderlich. Ein Objekt vom Typ "ScreenItems".

ScreenName (RT Professional)

Beschreibung

Legt das Bild fest, das im Bildfenster in Runtime angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ScreenName**[=HmiObjectHandle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ScreenWindow

HmiObjectHandle

Optional. Ein Wert oder eine Konstante, welche das Bild festlegt, das im Bildfenster in Runtime angezeigt wird.

Screens (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt die Screens-Auflistung zurück. Die Screens-Auflistung enthält zwei Elemente: Das erste Element mit dem Index 0 stellt das Permanentfenster dar. Das zweite Element mit dem Index 1 stellt das Grundbild dar. Alternativ können Sie die beiden Elemente auch über ihre Namen ansprechen. Verwenden Sie "Overview" für das Permanentfenster und "Base" für das Grundbild.

Hinweis

Das Meldefenster und der Meldeindikator sind nicht in der Screens-Auflistung enthalten, auch wenn sie in Runtime den Fokus haben.

Zugriff in Runtime: Lesen

Syntax

Object.Screens

Object

Erforderlich. Ein Objekt vom Typ "Screens".

SecondNeedleHeight (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Länge des Sekundenzeigers im Objekt "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SecondNeedleHeight**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Int32

Optional. Ein Wert oder eine Konstante, welche die Länge des Sekundenzeigers festlegt.

Geben Sie die Länge des Sekundenzeigers in Prozent an, bezogen auf den Radius des Zifferblattes.

SecondNeedleWidth (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Breite des Sekundenzeigers im Objekt "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SecondNeedleWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Int32

Optional. Ein Wert oder eine Konstante, der die Breite des Sekundenzeigers festlegt. Geben Sie die Breite in Prozent an, bezogen auf die doppelte Länge des Sekundenzeigers.

SegmentColoring (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Art des Farbumschlags fest, mit der im Objekt "Bar" das Überschreiten von Grenzwerten angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SegmentColoring**[=THmiBarColorType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

THmiBarColorType

Optional. Ein Wert oder eine Konstante, welche die Art des Farbumschlags festlegt. Wertebereich von 0 bis 1.

hmiBarColorEntire (0): Farbumschlag erfolgt für den gesamten Balken.

hmiBarColorSegmented (1): Farbumschlag erfolgt segmentweise.

SelectedText (RT Professional)

Beschreibung

Zeigt den mit dem Attribut "Selektiertes Feld" (SelIndex) festgelegten Text, der im Kombinationsfeld oder Listenfeld hervorgehoben dargestellt wird.

Siehe auch

Listbox (Seite 1424)

SelectBackColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe des ausgewählten Texteintrags des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicOField

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe des ausgewählten Texteintrags des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectedCellColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe der markierten Zelle fest. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectedCellColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Hintegrundfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectedCellForeColor (RT Professional)

Beschreibung

Legt die Schriftfarbe der markierten Zelle fest. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectedCellForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Schriftfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectedIndex (RT Professional)

Beschreibung

Legt den Index fest, dessen zugehöriger Text im Kombinationsfeld oder Listenfeld hervorgehoben angezeigt wird.

Der Maximalwert ist die Anzahl der Zeilen (NumberLines) des Objekts.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectedIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ComboBox
- ListBox

Int32

Optional. Ein Wert oder eine Konstante, welche den Index des Texts festlegt, der hervorgehoben wird.

SelectedRowColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe der markierten Zeile fest. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectedRowColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Hintegrundfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectedRowForeColor (RT Professional)

Beschreibung

Legt die Schriftfarbe der markierten Zeile fest. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectedRowForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Schriftfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectedTitleColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe der markierten Tabellenüberschrift fest. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Die Einstellung ist in Runtime nur wirksam, wenn die Option "Markierungsfarbe" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectedTitleColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrund festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectedTitleForeColor (RT Professional)

Beschreibung

Legt die Schriftfarbe der markierten Tabellenüberschrift fest. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Die Einstellung ist in Runtime nur wirksam, wenn die Option "Markierungsfarbe" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectedTitleForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Schriftfarbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Siehe auch

UserArchiveControl (Seite 1528)

TrendRulerControl (Seite 1508)

OnlineTableControl (Seite 1434)

AlarmControl (Seite 1338)

SelectForeColor (RT Professional)

Beschreibung

Legt die Farbe des ausgewählten Texteintrags des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicIOField

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des ausgewählten Texteintrags des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectionBackColor (RT Advanced)

Beschreibung

Legt die Hintergrundfarbe der selektierten Zellen fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmView
- RecipeView
- StatusForce
- UserView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- TrendView

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der selektierten Zeile festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectionColoring (RT Professional)

Beschreibung

Legt fest, ob Markierungsfarben für Zelle oder Zeile verwendet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionColoring**[=GridSelectionColoring]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

GridSelectionColoring

Wert	Beschreibung	Erklärung
0	Keine	Keine Markierungsfarben für Zelle und Zeile.
1	Zelle	Markierungsfarbe für Zelle.
2	Zeile	Markierungsfarbe für Zeile.
3	Zelle und Zeile	Markierungsfarben für Zelle und Zeile.

SelectionForeColor (RT Advanced)

Beschreibung

Legt die Vordergrundfarbe der selektierten Zellen fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmView
- RecipeView
- StatusForce
- UserView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- TrendView

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der selektierten Zeile festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectionRect (RT Professional)

Beschreibung

Legt fest, ob ein Auswahlrahmen für die markierten Zellen bzw. Zeilen verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionRect**[=GridSelectionBorder]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

GridSelectionBorder

Wert	Beschreibung	Erklärung
0	Kein	Für markierte Zellen und Zeilen wird kein Auswahlrahmen verwendet.
1	Zelle	Für die markierte Zelle wird ein Auswahlrahmen verwendet.
2	Zeile	Für die markierte Zeile wird ein Auswahlrahmen verwendet.

SelectionRectColor (RT Professional)

Beschreibung

Legt die Farbe des Auswahlrechtecks im Meldfenster fest, wenn der SelectionType "1" ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionRectColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SelectionRectWidth (RT Professional)

Beschreibung

Legt die Linienstärke des Auswahlrechtecks im Meldefenster fest, wenn der SelectionType "1" ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionRectWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Linienstärke festlegt.

SelectionMode (RT Professional)

Beschreibung

Legt fest, wie viele Zeilen Sie markieren können.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionMode**[=GridSelectionMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

GridSelectionMode

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Zeilen festlegt, die Sie markieren können.

Folgende Einstellungen stehen zur Verfügung:

Wert	Beschreibung	Erklärung
0	Keine	Keine Zeile wird markiert.
1	Einzelselektion	Eine Zeile kann markiert werden.
2	Mehrfachselektion	Mehrere Zeilen können markiert werden.

SeparatorBackColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe der durchbrochenen Trennlinien in der Auswahlliste des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SeparatorBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicOField

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der durchbrochenen Trennlinien in der Auswahlliste des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SeparatorColor (RT Professional)

Beschreibung

Legt die Farbe der Trennlinien in der Auswahlliste des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SeparatorColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- SymbolicIOField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- S7GraphOverview

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Trennlinien in der Auswahlliste des angegebenen Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

SeparatorCornerStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime

SeparatorStyle (RT Professional)

Beschreibung

Legt die Linienart der Trennlinien in der Auswahlliste des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SeparatorStyle**[=LineStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicIOField

LineStyle

hmiLineStyleNone (-1): Die Auswahlliste hat keine Trennlinien.

hmiLineStyleSolid (0): Die Auswahlliste hat durchgezogene Trennlinien.

hmiLineStyleDash (1): Die Auswahlliste hat gestrichelte Trennlinien.

hmiLineStyleDot (2): Die Auswahlliste hat gepunktete Trennlinien.

hmiLineStyleDashDot (3): Die Auswahlliste hat Strichpunktlinien als Trennlinien.

hmiLineStyleDashDotDot (4): Die Auswahlliste hat Strich-Punkt-Punkt-Linien als Trennlinien.

SeparatorWidth (RT Professional)

Beschreibung

Legt die Breite der Trennlinien in der Auswahlliste des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SeparatorWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicIOField

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite der Trennlinien in der Auswahlliste des angegebenen Objekts festlegt.

SeparatorLineEndShapeStyle (RT Professional)

Beschreibung

Legt die die Form der Linienenden für das Objekt vom Typ "ScreenItem" mit der Ausprägung "SymbolicIOField" fest.

Siehe auch

SymbolicIOField (Seite 1491)

ServerScale (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ServerNames (RT Professional)

Beschreibung

Legt die Server eines verteilten Systems fest, von denen die Meldeanzeige Daten bezieht. Die Angabe erfolgt in der Form: NameServer1;NameServer2;NameServer3.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ServerNames**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Server eines verteilten Systems festlegt, von denen das Meldefenster Daten bezieht.

ServerPrefix (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime

Shared (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

ShiftDecimalPoint (RT Professional)

Beschreibung

Legt fest, dass das Feld "Dezimalkomma verschieben" nur gelesen werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShiftDecimalPoint**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "IOField".

BOOLEAN

Optional. TRUE, wenn das Feld "Dezimalkomma verschieben" nur gelesen werden kann.

Siehe auch

IOField (Seite 1418)

ShowAlarmsFromDate (RT Advanced)

Beschreibung

Legt fest, dass nur diese Meldeereignisse angezeigt werden, die in dieser Variablen gespeichert sind.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowAlarmsFromDate**[=HmiObjectHandle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmView

HmiObjectHandle

Optional. Ein Wert oder eine Konstante, welche festlegt, dass nur diese Meldeereignisse angezeigt werden, die in dieser Variablen gespeichert sind.

ShowBadTagState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob das Objekt bei einem schlechten QualityCode oder Variablenstatus ausgegraut dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowBadTagState**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- IOField
- OptionGroup
- SymbolicIOField
- WindowsSlider

Bei folgenden Ausprägungen haben Sie keinen Zugriff in Runtime:

- CheckBox

BOOLEAN

TRUE	Das Objekt wird bei einem schlechten QualityCode oder Variablenstatus ausgegraut dargestellt bzw. die Einstellungen zur Rasterfarbe werden verwendet.
FALSE	Das Objekt wird bei einem schlechten QualityCode oder Variablenstatus nicht ausgegraut dargestellt bzw. die Einstellungen zur Rasterfarbe werden nicht verwendet.

ShowBar (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der angezeigte Prozesswert im Objekt "Slider" zusätzlich durch einen ausgefüllten Balken angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowBar**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

BOOLEAN

Optional. TRUE, wenn der Prozesswert zusätzlich durch einen ausgefüllten Balken angezeigt wird.

ShowCaption (RT Professional)

Beschreibung

Legt fest, ob die Titelzeile ein- oder ausgeblendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowCaption**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- ApplicationWindow
- Screenwindow

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Switch

BOOLEAN

Optional. TRUE, wenn die Titelzeile eingeblendet wird.

ShowDecimalPoint (RT Professional)

Beschreibung

Legt fest, ob die Skala mit Dezimalzahlen (Dezimalkomma und eine Dezimalstelle) oder mit ganzen Zahlen beschriftet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowDecimalPoint**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

BOOLEAN

Optional. TRUE, wenn die Skala mit Dezimalzahlen (Dezimalkomma und eine Dezimalstelle) beschriftet wird.

ShowFillLevel (RT Professional)

Beschreibung

Legt fest, ob das angegebene Objekt gefüllt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowFillLevel**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Button
- Circle
- CircleSegment
- Ellipse
- EllipseSegment
- GraphicView
- OptionGroup
- Polygon
- Rectangle

- RoundButton
- TextField
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox

BOOLEAN

Optional. TRUE, wenn das angegebene Objekt gefüllt wird.

ShowFocusRectangle (RT Professional)

Beschreibung

Legt fest, ob in Runtime die Schaltfläche einen Selektionsrahmen erhält, wenn sie aktiviert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowFocusRectangle**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

BOOLEAN

Optional. TRUE, wenn in Runtime die Schaltfläche einem Selektionsrahmen erhält, wenn sie aktiviert wird.

ShowLargeTicksOnly (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob nur die großen Skalenstriche dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowLargeTicksOnly**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn nur die großen Skalenstriche dargestellt werden.

ShowLimitMarkers (RT Professional)

Beschreibung

Legt fest, ob die Grenzwerte als Skalenwert angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowLimitMarkers**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Slider

BOOLEAN

Optional. TRUE, wenn die Grenzwerte als Skalenwert angezeigt werden.

ShowPeakValuePointer (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob für das angegebene Objekt ein Schleppezeiger verwendet wird.

Der Schleppezeiger zeigt in Runtime den maximalen Zeigerausschlag an, solange das Prozessbild geladen ist. Wenn Sie das Prozessbild neu laden, wird der Schleppezeiger wieder zurückgesetzt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowPeakValuePointer**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

BOOLEAN

Optional. TRUE, wenn der Schleppezeiger verwendet wird.

ShowPosition (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Wert der aktuellen Reglerposition zusätzlich numerisch angezeigt wird. Der Wert wird dann unterhalb des Reglers angezeigt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowPosition**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

BOOLEAN

Optional. TRUE, wenn der Wert zusätzlich numerisch angezeigt wird.

RowTitles (RT Professional)

Beschreibung

Legt fest, ob die Meldeanzeige eine Spalte mit einer fortlaufenden Nummerierung der anstehenden Meldungen enthält.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RowTitles**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Meldeanzeige eine Spalte mit einer fortlaufenden Nummerierung der anstehenden Meldungen enthält.

Siehe auch

AlarmControl (Seite 1338)

ShowRuler (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob für die Achsbeschriftung des Objekts "OnlineTrendControl" eine Skaleneinteilung (Hilfslinie) eingeblendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowRuler**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl
- TrendView

BOOLEAN

Optional. TRUE, wenn die Skaleneinteilung eingeblendet wird.

ShowRulerInAxis (RT Professional)

Beschreibung

Legt fest, ob die Lineale auch in den Zeitachsen angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowRulerInAxis**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Lineale werden auch in den Zeitachsen angezeigt.

FALSE: Die Lineale werden nicht in den Zeitachsen angezeigt.

ShowScale (RT Professional)

Beschreibung

Legt fest, ob die Werte zusätzlich in einer Skala angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowScale**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Slider

BOOLEAN

Optional. TRUE, wenn die Werte zusätzlich in einer Skala angezeigt werden.

ShowScrollbars (RT Professional)

Beschreibung

Legt fest, ob die Rollbalken angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowScrollbars**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- FunctionTrendControl
- OnlineTrendControl
- Screenwindow

ShowScrollbars

Optional. TRUE, wenn Rollbalken angezeigt werden.

ShowSortButton (RT Professional)

Beschreibung

Legt fest, ob die Sortiertaste über dem vertikalen Rollbalken angezeigt wird. Über die Sortiertaste sortieren Sie mit einem Mausklick die ausgewählte Spalte in der projektierten Sortierreihenfolge. Wenn in der Tabelle kein vertikaler Rollbalken vorhanden ist, wird die Sortiertaste nicht angezeigt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowSortButton**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Sortiertaste wird angezeigt. Sie können die ausgewählte Spalte sortieren.

FALSE: Die Sortiertaste wird nicht angezeigt.

ShowSortIcon (RT Professional)

Beschreibung

Legt fest, ob das Sortiersymbol angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowSortIcon**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Das Sortiersymbol wird angezeigt.

FALSE: Das Sortiersymbol wird nicht angezeigt.

ShowSortIndex (RT Professional)

Beschreibung

Legt fest, ob ein Sortierindex angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowSortIndex**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Ein Sortierindex wird angezeigt.

FALSE: Ein Sortierindex wird nicht angezeigt.

ShowStatusBar (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob die Statusleiste angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowStatusBar**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- HTML-Browser

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- MediaPlayer

BOOLEAN

Optional. TRUE, wenn die Statusleiste angezeigt wird.

ShowTableGridlines (RT Advanced)

Beschreibung

Legt fest, ob in der Tabelle des angegebenen Objekts Gitterlinien angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowTableGridlines**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- StatusForce

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- TrendView
- UserView

BOOLEAN

Optional. TRUE, wenn Gitterlinien in der Tabelle angezeigt werden.

ShowThumb (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Schieberegler des Objekts "Slider" angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowThumb**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

BOOLEAN

Optional. TRUE, wenn der Schieberegler angezeigt wird.

ShowTickLabels (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob in der Skala die Beschriftung angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowTickLabels**[= BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Slider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Bar

BOOLEAN

Optional. TRUE, wenn die Beschriftung angezeigt wird.

Bemerkungen

Die Schrittweite der Messwerte wird in Abhängigkeit vom angegebenen Messbereich und der Größe des Objekts automatisch festgelegt.

ShowTicks (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob die Teilstriche in der Skala des angegebenen Objekts angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowTicks**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock
- Slider

BOOLEAN

Optional. TRUE, wenn die Teilstriche angezeigt werden.

ShowTitle (RT Professional)

Beschreibung

Legt die Darstellung der Fensterüberschrift des Controls fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowTitle**[=WindowHeaderStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

WindowHeaderStyle

Wert	Bezeichnung	Erläuterung
0	nein	Keine Fensterüberschrift.
1	normal	Die Fensterüberschrift besteht aus einem WinCC Symbol und Text. Der Text wird im Feld "Text" eingegeben.
2	schmal	Die Fensterüberschrift besteht nur aus Text. Der Text wird im Feld "Text" eingegeben.

ShowToolBar (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob eine Symbolleiste angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowToolBar**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- HTMLBrowser

BOOLEAN

Optional. TRUE, wenn eine Symbolleiste angezeigt wird.

ShowTrendIcon (RT Professional)

Beschreibung

Legt fest, ob unterhalb der Wertachsen ein Symbol eingeblendet wird. Das Symbol zeigt, welche Kurve im Vordergrund angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowTrendIcon**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Das Sybmol wird angezeigt.

FALSE: Das Sybmol wird nicht angezeigt.

ShowTrendIndicator (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob die Tendenz (steigend oder fallend) des zu überwachenden Messwertes mit einem kleinen Pfeil angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowTrendIndicator**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn die Tendenz (steigend oder fallend) des zu überwachenden Messwertes mit einem kleinen Pfeil angezeigt wird.

Sizeable (RT Professional)

Beschreibung

TRUE, wenn in Runtime die Größe des Objektes verändert werden kann. BOOLEAN Schreib-Lese-Zugriff.

Bei Applikationsfenster und Bildfenster: Nur Lese-Zugriff

Siehe auch

TrendRulerControl (Seite 1508)

OnlineTableControl (Seite 1434)

OnlineTrendControl (Seite 1443)

AlarmControl (Seite 1338)

ControlDesignMode (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

In diesem Auswahlfeld wird der Stil des Controls festgelegt.

Zugriff in Runtime: Schreiben

Syntax

Object.**ControlDesignMode**[=RTControlModes]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "MessageView", "OnlineTableControl", "OnlineTrendControl", "FunctionTrendControl", "RecipeView", "RulerControl".

RTControlModes

Folgende Einstellungen stehen zur Verfügung:

Wert	Bezeichnung	Erläuterung
	Projekteinstellung	Der Stil entspricht den Projekteinstellungen im WinCC Explorer.
0	Einfach	"Klassischer" WinCC Stil
1	Standard	Neuer WinCC V7 Stil

SmartTags (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt die SmartTags-Auflistung zurück.

Zugriff in Runtime: Lesen

Syntax

Object.SmartTags

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

SortByTimeDirection (RT Advanced)

Beschreibung

Legt fest, ob die zuletzt eingegangene Meldung im Objekt "AlarmControl" oben angezeigt wird (aufsteigende Sortierreihenfolge).

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SortByTimeDirection**[=SortByTimeDirection]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItems" mit folgender Ausprägung:

- AlarmView

SortByTimeDirection

Optional. TRUE, wenn die zuletzt eingegangene Meldung oben angezeigt wird.

SortByTimeEnabled (RT Advanced)

Beschreibung

Legt fest, ob die Sortierung der Meldungen nach der Uhrzeit im Objekt "AlarmView" geändert werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SortByTimeEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItems" mit folgender Ausprägung:

- AlarmView

BOOLEAN

Optional. TRUE, wenn die Sortierung vom Bediener am Gerät geändert werden kann.

SortSequence (RT Professional)

Beschreibung

Legt fest, wie die Sortierreihenfolge durch Mausklick verändert werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SortSequence**[=GridSortSequence]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

GridSortSequence

Wert	Beschreibung	Erklärung
0	auf/ab/keine	Per Mausklick können Sie zwischen aufsteigender, absteigender und keiner Sortierung umschalten.
1	auf/ab	Per Mausklick können Sie zwischen aufsteigender und absteigender Sortierung umschalten.

StartAngle (RT Professional)

Beschreibung

Legt den Winkel fest, um den der Anfangspunkt des angegebenen Objekts von der Nulllage (0°) abweicht.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StartAngle**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- CircleSegment
- CircularArc
- EllipseSegment
- EllipticalArc
- TubeArcObject

Int32

Optional. Ein Wert oder eine Konstante, welche den Winkel festlegt, um den der Anfangspunkt des angegebenen Objekts von der Nulllage (0°) abweicht.

StartStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, wie der Linienanfang des angegebenen Objekts dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StartStyle**[=LineEndStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Line
- Polyline

LineEndStyle

Optional. Ein Wert oder eine Konstante, die den Linienanfang festlegt. Wertebereich von 0 bis 6.

hmiLineEndStyleNone (0): Die Linie besitzt kein Anfangssymbol.

hmiLineEndStyleArrow (1): Die Linie beginnt mit einer Pfeilspitze.

hmiLineEndStyleFilledArrow (2): Die Linie beginnt mit einer ausgefüllten Pfeilspitze.

hmiLineEndStyleFilledArrowReversed (3): Die Linie beginnt mit einer umgekehrten Pfeilspitze.

hmiLineEndStyleLine (4): Die Linie beginnt mit einer senkrechten Linie.

hmiLineEndStyleCircle (5): Die Linie beginnt mit einem Kreis.

hmiLineEndStyleFilledCircle (6): Die Linie beginnt mit einem ausgefüllten Kreis.

StartTop (RT Professional)

Beschreibung

Kein Zugriff in Runtime.

StartValue (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den absoluten Wert für den Nullpunkt der Skalenanzeige fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StartValue**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den absoluten Wert für den Nullpunkt der Skalenanzeige festlegt.

State (RT Professional)

Beschreibung

Gibt den Status einer Meldung aus.

Die folgende Tabelle zeigt die möglichen Status einer Meldung:

State	Status der Meldung
1	Gekommen
2	Gegangen
5	Gekommen und Kommentar
6	Gegangen und Kommentar

StatusbarBackColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe für die Statuszeile fest. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog. Damit die Einstellung wirksam wird, müssen Sie die Option "Anzeigen" aktivieren.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe für die Statuszeile festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

StatusbarElementAdd (RT Professional)

Beschreibung

Legt ein neues, benutzerdefiniertes Element der Statuszeile an. Den von WinCC vergebenen Namen können Sie im Feld "Objektnamen" ändern.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche ein neues, benutzerdefiniertes Element der Statuszeile festlegt.

StatusbarElementAutoSize (RT Professional)

Beschreibung

Legt fest, ob die Breite des ausgewählten Elements der Statuszeile automatisch eingestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementAutoSize**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Breite des ausgewählten Elements wird automatisch eingestellt

FALSE: Die Breite des ausgewählten Elements wird nicht automatisch eingestellt.

Siehe auch

UserArchiveControl (Seite 1528)
TrendRulerControl (Seite 1508)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)
OnlineTrendControl (Seite 1443)
AlarmControl (Seite 1338)

StatusbarElementCount (RT Professional)

Beschreibung

Legt die Anzahl der projektierbaren Elemente der Statuszeile fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der projektierbaren Elemente der Statuszeile festlegt.

StatusbarElementIconId (RT Professional)

Beschreibung

Legt die Zuordnung von Ident-Nummer und Symbol eines Elementes der Statuszeile fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementIconId**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Zuordnung von Ident-Nummer und Symbol eines Elementes der Statuszeile festlegt.

StatusbarElementID (RT Professional)

Beschreibung

Legt die Ident-Nummer für das gewählte Element der Statuszeile fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementID**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Ident-Nummer für das gewählte Element der Statuszeile festlegt.

StatusbarElementIndex (RT Professional)

Beschreibung

Legt die Referenz eines Elements der Statuszeile fest.

Gültige Werte für "StatusbarElementIndex" liegen zwischen 0 und "StatusbarElementCount" minus 1.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Referenz eines Elements der Statuszeile festlegt.

StatusbarElementName (RT Professional)

Beschreibung

Legt den Objektnamen für das ausgewählte Element der Statuszeile fest.

Den Objektnamen für benutzerdefinierte Elemente der Statuszeile können Sie ändern. Die Eigenschaft "StatusbarElementName" für benutzerdefinierte Elemente ist über die Eigenschaft "StatusbarElementRename" dynamisierbar.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Objektnamen für das ausgewählte Element der Statuszeile festlegt.

StatusbarElementRemove (RT Professional)

Beschreibung

Entfernt das ausgewählte benutzerdefinierte Element der Statuszeile.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen des zu entfernenden benutzerdefinierten Elements der Statuszeile festlegt.

StatusbarElementRename (RT Professional)

Beschreibung

Ändert den Namen des benutzerdefinierten Elements der Statuszeile, das über die Eigenschaft "StatusbarElementIndex" referenziert wird.

Für benutzerdefinierte Elemente ist die Eigenschaft mit dem Namen "StatusbarElementRename" dynamisierbar. Mit "StatusbarElementRename" dynamisieren Sie auch die Eigenschaft "StatusbarElementName".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementRename**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den neuen Objektnamen für das ausgewählte Element der Statuszeile festlegt.

StatusbarElements (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die in der Statuszeile darzustellenden Elemente fest.

Kein Zugriff in Runtime.

StatusbarElementTooltipText (RT Professional)

Beschreibung

Legt den Text für den Tooltip des ausgewählten benutzerdefinierten Elements der Statuszeile fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementTooltipText**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Text für den Tooltip des ausgewählten benutzerdefinierten Elements der Statuszeile festlegt.

StatusbarElementUserDefined (RT Professional)

Beschreibung

Legt fest, ob das Element der Statuszeile vom Projektteur als neues benutzerdefiniertes Element hinzugefügt wurde.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementUserDefined**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Das Element der Statuszeile ist benutzerdefiniert.

FALSE: Das Element der Statuszeile ist vom System vorgegeben.

StatusbarElementVisible (RT Professional)

Beschreibung

Legt fest, ob das ausgewählte benutzerdefinierte Element der Statuszeile in Runtime angezeigt wird.

Aktivieren Sie in der Liste die Elemente der Statuszeile, die Sie in Runtime anzeigen wollen.

Klicken Sie auf einen Eintrag in der Liste, um die Eigenschaften anzupassen oder um die Position in der Statuszeile des Controls mit den Tasten "Auf" und "Ab" zu ändern.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Das ausgewählte benutzerdefinierte Element der Statuszeile wird angezeigt.

FALSE: Das ausgewählte benutzerdefinierte Element der Statuszeile wird nicht angezeigt.

StatusbarElementWidth (RT Professional)

Beschreibung

Gibt die Breite des ausgewählten benutzerdefinierten Elements der Statuszeile in Pixeln an.

Wenn die Option "Automatisch" nicht aktiviert ist, können Sie die Breite festlegen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarElementWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des ausgewählten benutzerdefinierten Elements der Statuszeile in Pixeln festlegt.

StatusbarFontColor (RT Professional)

Beschreibung

Legt die Schriftfarbe der Texte in der Statuszeile fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarFontColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Texte in der Statuszeile festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

StatusbarShowTooltips (RT Professional)

Beschreibung

Legt fest, ob in Runtime die Tooltips zu den Elementen der Statuszeile angezeigt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarShowTooltips**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Tooltips werden angezeigt.

FALSE: Die Tooltips werden nicht angezeigt.

StatusbarText (RT Professional)

Beschreibung

Legt den Standardtext der Statuszeile fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarText**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Standardtext der Statuszeile festlegt.

StatusbarUseBackColor (RT Professional)

Beschreibung

Legt fest, ob die Hintergrundfarbe der Statuszeile angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarUseBackColor**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Hintergrundfarbe der Statuszeile wird angezeigt.

FALSE: Die Hintergrundfarbe der Statuszeile wird nicht angezeigt.

StatusbarVisible (RT Professional)

Beschreibung

Legt fest, ob die Statuszeile des Controls angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Statuszeile wird angezeigt.

FALSE: Die Statuszeile wird nicht angezeigt.

Style (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Linienstil des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Style**[=LineStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- CircularArc
- EllipticalArc
- Line
- Polyline

LineStyle

Optional. Ein Wert oder eine Konstante, welche den Linienstil festlegt. Wertebereich von 0 bis 4.

hmiLineStyleSolid (0): durchgezogene Linie

hmiLineStyleDash (1): gestrichelte Linie

hmiLineStyleDot (2): gepunktete Linie

hmiLineStyleDashDot (3): Strich-Punkt-Linie

hmiLineStyleDashDotDot (4): Strich-Punkt-Punkt-Linie

Standardeinstellung: hmiLineStyleSolid

StyleSettings (RT Professional)

Beschreibung

Legt fest, in welchem Stil das Objekt dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StyleSettings**[=WinCCStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton
- WindowsSlider

WinCCStyle

Optional. Ein Wert oder eine Konstante, welche den Stil festlegt, in dem das Objekt dargestellt wird.

Benutzerdefiniert Stellt das Objekt entsprechend den eigenen Einstellungen dar.

Global Stellt das Objekt im global eingestellten Design dar.

Windows-Stil Stellt das Objekt im Windows-Stil dar.

SwapFirstWithLastConnection (RT Professional)

Beschreibung

Legt fest, ob der Text im Objekt horizontal dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SwapFirstWithLastConnection**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Connector".

BOOLEAN

Optional. TRUE, wenn der Text im Objekt horizontal dargestellt wird.

Siehe auch

Connector (Seite 1379)

3.5.5.13 Eigenschaften T (Panels, Comfort Panels, RT Advanced, RT Professional)

TableBackColor (RT Advanced)

Beschreibung

Legt die Hintergrundfarbe der Tabellenzellen des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmView
- RecipeView
- StatusForce
- TrendView
- UserView

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der Tabellenzellen festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TableColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe für die Zeilen fest.
Über die öffnen Sie den Farbauswahl-Dialog.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der Tabellenzeilen festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TableColor2 (RT Professional)

Beschreibung

Legt die zweite Hintergrundfarbe für die Zeilen fest.
Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Die Einstellung ist in Runtime nur wirksam, wenn die Option "Zeilenfarbe 2" aktiviert ist. Dann wird für den Zeilenhintergrund abwechselnd die "Zeilenfarbe 1" und die "Zeilenfarbe 2" verwendet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableColor2**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die zweite Hintergrundfarbe der Tabellenzeilen festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TableForeColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Textfarbe in den Tabellenzellen des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- AlarmView
- OnlineTableControl

- RecipeView
- StatusForce
- TrendRulerControl
- UserArchiveControl
- UserView

Color

Optional. Ein Wert oder eine Konstante, welche die Textfarbe in den Tabellenzellen festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TableForeColor2 (RT Professional)

Beschreibung

Legt die zweite Schriftfarbe fest.

Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Die Einstellung ist in Runtime nur wirksam, wenn die Option "Zeilenfarbe 2" aktiviert ist. Dann wird als Schriftfarbe abwechselnd die Schriftfarbe der "Zeilenfarbe 1" und der "Zeilenfarbe 2" verwendet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableForeColor2**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die zweite Schriftfarbe in den Tabellenzellen festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TableGridLineColor (RT Advanced)

Beschreibung

Legt die Farbe der Gitterlinien in der Tabelle des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableGridLineColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- RecipeView
- TrendView
- UserView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- StatusForce
- SysDiagControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Gitterlinien der Tabelle festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TableHeaderBackColor (RT Advanced)

Beschreibung

Legt die Hintergrundfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableHeaderBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmView
- RecipeView
- StatusForce
- TrendView
- UserView

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe in der Kopfzeile festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TableHeaderForeColor (RT Advanced)

Beschreibung

Legt die Textfarbe in der Kopfzeile der Tabelle des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableHeaderForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmView
- RecipeView
- StatusForce
- TrendView
- UserView

Color

Optional. Ein Wert oder eine Konstante, welche die Textfarbe in der Kopfzeile festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TagPrefix (RT Professional)

Beschreibung

Legt ein Präfix fest, das allen im Bild verwendeten Variablen vorangestellt wird.

Für ein Bildfenster kann ein Variablen-Präfix vergeben werden. Dieses Präfix wird allen im Bild verwendeten Variablen vorangestellt. So behält ein Bild, das in einem Bildfenster eingebunden ist, den Zugriff auf eigene Variablen, während ein anderes auf andere Variablen zugreift.

Die Änderung des Variablen-Präfix wird erst bei erneutem Laden des Bildes wirksam. Bei einem Bildwechsel geschieht dies automatisch; sonst erst, wenn der Bildname erneut zugewiesen wird.

Das Präfix ist frei definierbar, muss aber mit dem Namen der Strukturvariablen übereinstimmen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TagPrefix**[= STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Screenwindow

STRING

Optional. Ein Wert oder eine Konstante, welche das Variablen-Präfix festlegt.

Beispiel

Im Bildfenster soll das Bild "EinAusgabe" angezeigt werden. Das Bild "EinAusgabe" enthält drei EA-Felder die an eine Strukturvariable gebunden sind. Die Strukturvariable besteht aus den Elementen EA1, EA2, EA3; jeweils ein Element für jedes EA-Feld.

Im Projekt sind z. B. drei solche Strukturvariablen definiert, mit den Strukturnamen Struct1, Struct2 und Struct3.

Das Variablen-Präfix ist in diesem Fall der Strukturname mit anschließendem Punkt. Geben Sie als Variablen-Präfix z. B. "Struct2." an (der Punkt ist notwendig, um die Elemente der Strukturvariablen syntaktisch richtig als Strukturelemente anzusprechen), dann werden die EA-Felder im Bild "EinAusgabe" mit den Elementen der Strukturvariablen Struct2 verbunden:

Variablen-Präfix: "Struct2."

- Ausgabewert (erstes EA-Feld): EA1
- Ausgabewert (zweites EA-Feld): EA2
- Ausgabewert (drittes EA-Feld): EA3

Die aktuelle Variablenanbindung im Bildfenster wird damit

- Ausgabewert (erstes EA-Feld): Struct2.EA1
- Ausgabewert (zweites EA-Feld): Struct2.EA2
- Ausgabewert (drittes EA-Feld): Struct2.EA3

Tags (RT Professional)

Beschreibung

Gibt das Objekt vom Typ "Tags" zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**Tags**

Object

Erforderlich. Ein Objekt vom Typ "HMIRuntime".

Beispiel

Das folgende Beispiel greift auf die Variable "Tag1" zu:

```
'VBS86
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
```

Siehe auch

HMIRuntime (Seite 1313)

Template (RT Professional)

Beschreibung

Legt die Vorlage für die Anzeige des Fensterinhalts im Objekt "Applikationsfenster" fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Template**[=TemplateType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ApplicationWindow

TemplateType

Optional. Ein Wert oder eine Konstante, welche die Vorlage festlegt.

Abhängig vom Wert der Eigenschaft sind die folgenden Vorlagen möglich:

Fensterinhalt = Global Script

- "GSC-Diagnose"
Das Applikationsfenster wird von Applikationen des Global Script versorgt. Dargestellt werden die Ergebnisse des Diagnosesystems.
- "GSC-Runtime"
Das Applikationsfenster wird von Applikationen des Global Script versorgt. Dargestellt werden Analyseergebnisse zum Verhalten in Runtime.

Fensterinhalt = Print Jobs

- "All Jobs":
Das Applikationsfenster wird vom Protokolliersystem versorgt. Die verfügbaren Protokolle werden als Liste dargestellt.
- "All Jobs - Context Menu":
Das Applikationsfenster wird vom Protokolliersystem versorgt. Die verfügbaren Protokolle werden als Liste dargestellt. Ein Kontextmenü ermöglicht die Auswahl der Druckoptionen, die Darstellung einer Druckvorschau sowie den Ausdruck eines Protokolls.
- "Job Detail View":
Das Applikationsfenster wird vom Protokolliersystem versorgt. Die verfügbaren Protokolle werden in einem Auswahlménü dargestellt. Für das gewählte Protokoll werden Detailinformationen angezeigt.
- "Selected Jobs - Context Menu":
Das Applikationsfenster wird vom Protokolliersystem versorgt. Die verfügbaren Protokolle werden als Liste dargestellt. Diese Liste beinhaltet nur die Protokolle, für die Sie im Dialog "Druckauftrageigenschaften" die Option "Markierung für Druckauftragsliste" aktiviert haben. Ein Kontextmenü ermöglicht die Auswahl der Druckoptionen, die Darstellung einer Druckvorschau sowie den Ausdruck eines Protokolls.

Text (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Beschriftung für das Textfeld fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Text**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- TextField
- CheckBox
- ComboBox
- ListBox
- MultiLineEdit
- OptionGroup
- RoundButton

STRING

Optional. Ein Wert oder eine Konstante, welche die Beschriftung festlegt.

TextList (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt eine Auflistung zurück, welche die Zuordnungen zwischen dem Ausgabewert und dem tatsächlich auszugebenden Ausgabertext enthält.

Die Zuordnungen sind abhängig von der eingestellten Listenart. Die Listenart legen Sie mit der Eigenschaft "ListType" fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen

Syntax

Object.**TextList**[=HmiObjectHandle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SymbolicOField

Bei folgenden Ausprägungen haben Sie keinen Zugriff in Runtime:

- Button

HmiObjectHandle

Eine Auflistung, welche die Zuordnungen zwischen dem Ausgabewert und dem tatsächlich auszugebenden Ausgabertext enthält.

TextOff (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Text fest, der im Zustand "Aus" des angegebenen Objekts angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TextOff**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- Switch

Bei folgenden Ausprägungen haben Sie keinen Zugriff in Runtime:

- SymbolicOField

STRING

Optional. Ein Wert oder eine Konstante, welche die Beschriftung für den Zustand "Aus" festlegt.

Bemerkungen

Die Eigenschaft ist nur verfügbar, wenn das referenzierte Objekt "SymbolicOField", "Button" oder "Switch" vom Typ "Text" ist.

TextOn (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Text fest, der im Zustand "Ein" des angegebenen Objekts angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TextOn**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Switch

Bei folgenden Ausprägungen haben Sie keinen Zugriff in Runtime:

- Button
- SymbolicIOField

STRING

Optional. Ein Wert oder eine Konstante, welche die Beschriftung für den Zustand "Ein" festlegt.

Bemerkungen

Die Eigenschaft ist nur verfügbar, wenn das referenzierte Objekt "SymbolicIOField", "Button" oder "Switch" vom Typ "Text" ist.

TextOrientation (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Textrichtung des angegebenen Objekts fest.

Zugriff in Runtime:

- RT Advanced: kein Zugriff
- RT Professional: Lesen/Schreiben

Syntax

Object.**TextOrientation**[=TextOrientation]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- IOField
- OptionGroup
- RoundButton
- SymbolicIOField
- TextField
- WindowSlider

Bei folgenden Ausprägungen haben Sie keinen Zugriff in Runtime:

- CheckBox
- DateTimeField
- Switch

TextOrientation

hmiTextHorizontal (0): Der Text wird horizontal dargestellt.

hmiTextRotated90Degree (-1): Der Text wird vertikal dargestellt und links ausgerichtet.

hmiTextRotated270Degree (1): Der Text wird vertikal dargestellt und rechts ausgerichtet.

ThumbBackColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Hintergrundfarbe des Reglers im Objekt "Slider" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ThumbBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider
- WindowsSlider

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe des Reglers festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TicksColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe der Stundenmarken auf dem Zifferblatt des Objekts "Clock" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TicksColor[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Clock

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Stundenmarken festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TickStyle (RT Professional)

Beschreibung

Legt die Darstellung der Skala fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TickStyle[=SliderTickStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Slider

Bei folgenden Ausprägungen haben Sie keinen Zugriff in Runtime:

- Clock

SliderTickStyle

hmiSliderTickStyleNone (0): Das Objekt hat keine Skala.

hmiSliderTickStyleEffect1 (1): Die Skala besteht nur aus Haupteinteilungen. Die Skala ist schwarz auf weißem Grund.

hmiSliderTickStyleEffect2 (2): Die Skala besteht nur aus Haupteinteilungen. Die Skala ist weiß auf schwarzem Grund.

hmiSliderTickStyleNormal (3): Die Skala besteht aus einfachen Einteilungen.

Bemerkungen

Auf Grund der automatischen Skalierung ist es möglich, dass stellenweise zwei Teilstriche der Skala direkt nebeneinander liegen (scheinbar breiter Teilstrich). Durch geringfügiges Verlängern oder Verkürzen des Sliderobjekts kann dieser Effekt korrigiert werden.

Darüber hinaus können Sie auch die Darstellung der Skalierung völlig unterdrücken ("WithAxes").

TimeAxisBeginTime(i) (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Startzeitpunkt für die Darstellung der angegebenen Kurve fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "UseTimeRange(i)" und "ShareTimeAxis".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeAxisBeginTime**(i)[=DateTime]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- TrendView

DateTime

Optional. Ein Wert oder eine Konstante, die den Startzeitpunkt für die Darstellung der angegebenen Kurve festlegt.

TimeAxisEndTime (RT Professional)

Beschreibung

Legt den Endzeitpunkt für die Darstellung der angegebenen Kurve fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "Autorange", "UseTimeRange(i)" und "ShareTimeAxis".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeAxisEndTime**[=DateTime]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl

DateTime

Optional. Ein Wert oder eine Konstante, die den Endzeitpunkt für die Darstellung der angegebenen Kurve festlegt.

TimeAxisLabel(i) (RT Professional)

Beschreibung

Legt die Bezeichnung der Zeitachse fest. Ob die Angabe ausgewertet wird, ist abhängig von der Eigenschaft "ConfigureTimeAxis(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeAxisLabel**(i)[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Bezeichnung der Zeitachse festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

TimeAxisRange (RT Advanced)

Beschreibung

Kein Zugriff in Runtime.

TimeAxisTimeFormat(i) (RT Professional)

Beschreibung

Legt das Format der Angaben entlang der Zeitachse für die angegebene Kurve fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeAxisTimeFormat**(i)[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, die das Format für die Zeitachse festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

TimeBase (RT Professional)

Beschreibung

Legt die Zeitzone fest, auf deren Basis die Zeitwerte angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeBase**[=TimeBase]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- UserArchiveControl

TimeBase

hmiTimeBaseLocalTimezone (0): Ortzeit

hmiTimeBaseServerTimezone (1): Zeitzone des Servers

hmiTimeBaseUTC (2): UTC (Universal Time Coordinated)

hmiTimeBaseProjectSetting (3): Projekteinstellungen

Bemerkungen

Über den WinCC-Explorer bei der Eigenschaftsseite des Rechners kann der Uhrzeitmodus rechner-spezifisch festgelegt werden.

TimeColumnActualize (RT Professional)

Beschreibung

Legt fest, ob die Werte der ausgewählten Spalte aktualisiert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnActualize**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

BOOLEAN

TRUE: Die Zeitspalte wird aktualisiert.

FALSE: Die Zeitspalte wird nicht aktualisiert. Diese Einstellung ist sinnvoll, wenn eine Tabelle mit einer andere Tabelle verglichen wird.

TimeColumnAdd (RT Professional)

Beschreibung

Legt eine neue Zeitspalte an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

STRING

Optional. Ein Wert oder eine Konstante, die eine neue Zeitspalte festlegt.

TimeColumnAlignment (RT Professional)

Beschreibung

Legt fest, wie die ausgewählte Zeitspalte ausgerichtet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnAlignment** [=HorizontalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

HorizontalAlignment

Wert	Beschreibung	Erklärung
0	links	Die ausgewählte Zeitspalte wird links angezeigt.
1	zentriert	Die ausgewählte Zeitspalte wird zentriert angezeigt.
2	rechts	Die ausgewählte Zeitspalte wird rechts angezeigt.

Siehe auch

OnlineTableControl (Seite 1434)

TimeColumnAlignment(i) (RT Professional)

Beschreibung

Legt die Ausrichtung der Zeitspalte eines Spaltenpaars i fest. Der Parameter i gibt die Nummer des Spaltenpaars an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnAlignment(i)**[=AlignmentHorizontal]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "TableView".

AlignmentHorizontal

hmiAlignmentLeft (0): Der Text wird links ausgerichtet.

hmiAlignmentCentered (1): Der Text wird zentriert ausgerichtet.

hmiAlignmentRight (2): Der Text wird rechts ausgerichtet.

Siehe auch

OnlineTableControl (Seite 1434)

TimeColumnBackColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe der ausgewählten Zeitspalte fest.

Die Einstellung ist wirksam:

- Wenn die Option "in den Farben der Wertspalte" nicht aktiviert ist.
- Wenn auf der Registerkarte "Allgemeines" im Bereich "Spaltenfarbe verwenden" die Option "Schriftfarbe" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der ausgewählten Zeitspalte festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TimeColumnBeginTime (RT Professional)

Beschreibung

Legt den Anfangszeitpunkt des Zeitbereichs für die ausgewählte Zeitspalte fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnBeginTime**[=DateTime]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

DateTime

Optional. Ein Wert oder eine Konstante, welche den Anfangszeitpunkt für die ausgewählte Zeitspalte festlegt.

TimeColumnCaption (RT Professional)

Beschreibung

Legt die Bezeichnung der Zeitspalte fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnCaption**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Bezeichnung der Zeitspalte festlegt.

TimeColumnCount (RT Professional)

Beschreibung

Legt die Anzahl der projizierten Zeitspalten fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der projizierten Zeitspalten festlegt.

TimeColumnNameFormat (RT Professional)

Beschreibung

Legt fest, welches Datumsformat zur Darstellung der ausgewählten Zeitspalte verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnNameFormat**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

STRING

Folgende Datumsformate stehen zur Verfügung:

Wert	Erklärung
dd.MM.yy	Tag.Monat.Jahr, z. B. 24.12.13.
dd.yyyd.MM	Tag.Monat.Jahr, z. B. 24.12.2013.
dd/MM/yy	Tag/Monat/Jahr, z. B. 24/12/13
dd/MM/yyyy	Tag/Monat/Jahr, z. B. 24/12/2013.

TimeColumnEndTime (RT Professional)

Beschreibung

Legt den Endzeitpunkt des Zeitbereichs für die ausgewählte Zeitspalte fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnBeginTime**[=DateTime]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

DateTime

Optional. Ein Wert oder eine Konstante, die den Endzeitpunkt festlegt.

TimeColumnForeColor (RT Professional)

Beschreibung

Legt die Schriftfarbe der ausgewählten Zeitspalte fest.

Die Einstellung ist wirksam:

- Wenn die Option "in den Farben der Wertspalte" nicht aktiviert ist.
- Wenn auf der Registerkarte "Allgemeines" im Bereich "Spaltenfarbe verwenden" die Option "Schriftfarbe" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

OnlineTableControl

Color

Optional. Ein Wert oder eine Konstante, welche die Schriftfarbe der ausgewählten Zeitspalte festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TimeColumnHideText (RT Professional)

Beschreibung

Legt fest, ob der Inhalt der Zeitspalte als Text angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnHideText**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

BOOLEAN

TRUE: Der Inhalt wird nicht als Text angezeigt.

FALSE: Der Inhalt wird als Text angezeigt.

TimeColumnHideTitleText (RT Professional)

Beschreibung

Legt fest, ob die Überschrift der Zeitspalte als Text angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnHideTitleText**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

BOOLEAN

TRUE: Die Überschrift wird nicht als Text angezeigt.

FALSE: Die Überschrift wird als Text angezeigt.

TimeColumnIndex (RT Professional)

Beschreibung

Referenziert eine projektierte Zeitspalte. Unter Verwendung der Eigenschaft können Sie einer bestimmten Zeitspalte die Werte andere Eigenschaften zuweisen.

Gültige Werte für "TimeColumnIndex" liegen zwischen 0 und "TimeColumnCount" minus 1. Die Eigenschaft "TimeColumnCount" gibt die Anzahl der projektierten Zeitspalten an.

Syntax

Object.**TimeColumnIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

Int32

Optional. Ein Wert oder eine Konstante, die eine projektierte Zeitspanne referenziert.

TimeColumnLength (RT Professional)

Beschreibung

Legt die Breite für die ausgewählte Zeitspalte fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnLength**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- OnlineTableControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite für die ausgewählte Zeitspalte festlegt.

TimeColumnMeasurePoints (RT Professional)

Beschreibung

Legt die Anzahl der Messpunkte fest, die in der ausgewählten Zeitspalte angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnMeasurePoints**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Messpunkte festlegt.

TimeColumnName (RT Professional)

Beschreibung

Legt den Namen der ausgewählten Zeitspalte fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

STRING

Optional. Ein Wert oder eine Konstante, die den Namen der ausgewählten Zeitspalte festlegt.

TimeColumnRangeType (RT Professional)

Beschreibung

Legt den Zeitbereich fest, der für die ausgewählte Zeitspalte verwendet wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnRangeType**[=TimeRangeMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

TimeRangeMode

Wert	Beschreibung	Erklärung
0	Zeitbereich	Für die Zeitspalte werden Anfangszeitpunkt und Zeitbereich festgelegt.
1	Anfangs- bis Endzeitpunkt	Für die Zeitspalte werden Anfangs- und Endzeitpunkt festgelegt.
2	Anzahl der Messpunkte	Für die Zeitspalte werden Anfangszeitpunkt und Anzahl der Messpunkte festgelegt.

TimeColumnRemove (RT Professional)

Beschreibung

Entfernt die ausgewählte Zeitspalte aus der Liste.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

STRING

Optional. Entfernt die ausgewählte Zeitspalte aus der Liste.

TimeColumnRename (RT Professional)

Beschreibung

Ändert den Namen der Zeitspalte, die über die Eigenschaft "TimeColumnIndex" referenziert wird. Mit "TimeColumnRename" dynamisieren Sie auch die Eigenschaft "TimeColumnName".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnRename**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

STRING

Optional. Ändert den Namen der Zeitspalte, die über die Eigenschaft "TimeColumnIndex" referenziert wird.

TimeColumnRepos (RT Professional)

Beschreibung

Ändert die Reihenfolge der Zeitspalten mit den zugehörigen Wertspalten. "Auf" und "Ab" bewegen die ausgewählte Zeitspalte in der Liste nach oben oder unten. Dadurch wird die Zeitspalte mit den zugehörigen Wertspalten in der Tabelle weiter vorne oder hinten platziert.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnRepos**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

Int32

Optional. Wert oder Konstante, welche die Reihenfolge der Zeitspalten mit den zugehörigen Wertspalten ändert.

TimeColumnShowDate (RT Professional)

Beschreibung

Legt fest, ob die ausgewählte Zeitspalte mit Datum und Uhrzeit angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnShowDate**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

BOOLEAN

TRUE: Datum und Uhrzeit werden angezeigt. Das Datumsformat wird im Feld "Datumsformat" festgelegt.

FALSE: Das Datum wird nicht angezeigt. Nur die Uhrzeit wird angezeigt.

TimeColumnShowIcon (RT Professional)

Beschreibung

Legt fest, ob der Inhalt der Zeitspalte als Symbol angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnActualize**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- OnlineTableControl

BOOLEAN

TRUE: Der Inhalt wird als Symbol angezeigt.

FALSE: Der Inhalt wird nicht als Symbol angezeigt.

TimeColumnShowTitleIcon (RT Professional)

Beschreibung

Legt fest, ob die Überschrift als Symbol angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnShowTitleIcon**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

BOOLEAN

TRUE: Die Überschrift wird als Symbol angezeigt.

FALSE: Die Überschrift wird nicht als Symbol angezeigt.

TimeColumnSort (RT Professional)

Beschreibung

Legt fest, wie die im "TimeColumnIndex" referenzierte Zeitspalte sortiert wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnSort**[=SortMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

SortMode

Wert	Beschreibung	Erklärung
0	nein	Keine Sortierung
1	aufsteigend	Aufsteigende Sortierung vom kleinsten zum größten Wert.
2	absteigend	Absteigende Sortierung vom größten zum kleinsten Wert.

TimeColumnSortIndex (RT Professional)

Beschreibung

Legt die Sortierreihenfolge der im "TimeColumnIndex" referenzierten Zeitspalte fest. Wenn Sie den Wert auf "0" setzen, wird das Sortierkriterium in "TimeColumnSort" entfernt.

Syntax

Object.**TimeColumnSortIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

Int32

Optional. Wert oder Konstante, welche die Sortierreihenfolge der im "TimeColumnIndex" referenzierten Zeitspalte festlegt. Wenn Sie den Wert auf "0" setzen, wird das Sortierkriterium in "TimeColumnSort" entfernt.

TimeColumnTimeFormat (RT Professional)

Beschreibung

Legt fest, welches Zeitformat zur Darstellung der ausgewählten Zeitspalte verwendet wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnTimeFormat**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

STRING

Wert	Erklärung
Automatisch	Das Zeitformat wird automatisch bestimmt.
HH:mm:ss.ms	Stunden:Minuten:Sekunden, z.B. 15:35:44.240.
hh:mm:ss tt	Stunden:Minuten:Sekunden AM/PM, z.B. 03:35:44 PM.
hh:mm:ss.ms tt	Stunden:Minuten:Sekunden.Millisekunden AM/PM, z.B. 03:35:44.240 PM.

TimeColumnTimeRangeBase (RT Professional)

Beschreibung

Legt die Zeiteinheit zur Bestimmung des Zeitbereichs fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnTimeRangeBase**[=TagLoggingTimeUnit]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

TagLoggingTimeUnit

Wert	Beschreibung
500	500 ms
1000	1 Sekunde
60000	1 Minute

Wert	Beschreibung
3600000	1 Stunde
86400000	1 Tag

TimeColumnTimeRangeFactor (RT Professional)

Beschreibung

Legt den Faktor zur Bestimmung des Zeitbereichs fest. Nur ganzzahlige Faktoren sind zulässig.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnTimeRangeFactor**[=Int16]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

Int16

Optional. Ein Wert oder eine Konstante, die den Faktor zur Bestimmung des Zeitbereichs festlegt.

TimeColumnUseValueColumnColors (RT Professional)

Beschreibung

Legt fest, ob die ausgewählte Zeitspalte in den Farben der Wertspalte angezeigt wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnUseValueColumnColors**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

BOOLEAN

TRUE: Die ausgewählte Zeitspalte wird in den Farben der Wertspalte angezeigt. Die Einstellungen in den Feldern "Schriftfarbe" und "Hintergrundfarbe" sind unwirksam.

FALSE: Die ausgewählte Zeitspalte wird in den Farben angezeigt, die in den Feldern "Schriftfarbe" und "Hintergrundfarbe" festgelegt sind.

TimeColumnVisible (RT Professional)

Beschreibung

In der Liste werden die Zeitspalten aufgelistet, die Sie angelegt haben. Aktivieren Sie in der Liste die Zeitspalten, die Sie in der Tabelle anzeigen wollen.

Klicken Sie auf eine Zeitspalte in der Liste, um die Eigenschaften anzupassen und um den Zeitbereich der Zeitspalte festzulegen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

BOOLEAN

Optional. TRUE, wenn die Zeitspalte in der Tabelle angezeigt wird.

TimeStamp (RT Professional)

Beschreibung

Gibt den Zeitstempel des letzten Lesezugriffs einer Variablen in lokaler Zeit als DATE zurück.

Zugriff in Runtime: Lesen

Syntax

Object.**TimeStamp**

Object

Erforderlich. Ein Objekt vom Typ "Tag".

Bemerkungen

Um die TimeStamp-Eigenschaft als Klartext auszugeben, verwenden Sie die VBS-Funktion "FormatDateTime(Date[, NamedFormat])". Die Ausgabe ist abhängig von der Spracheinstellung. Um die Sprache einzustellen, verwenden Sie die VBS-Funktion "SetLocale()".

Wenn Sie den Zeitstempel z. B. getrennt nach Datum, Tag und Uhrzeit zurückgeben wollen, verwenden Sie den Parameter `NamedFormat` oder die VBS-Funktionen wie `Year`, `WeekDay`, `Day`, `Hour`, `Minute`, `Second`. Den Namen eines Wochentages erhalten Sie mit der VBS-Funktion `WeekdayName`.

Beispiele

Das folgende Beispiel gibt den Zeitstempel der Variablen "Tag11" mit Hilfe der Funktion "FormatDateTime" aus:

```
'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
```

Das folgende Beispiel gibt den Zeitstempel der Variablen "Tag1" aus:

```
'VBS88
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Read
MsgBox objTag.TimeStamp
```

Siehe auch

Tag (Seite 1332)

TitleCut (RT Professional)

Beschreibung

Legt fest, ob die Inhalte der Felder einer Titelleiste bei zu kleiner Spaltenbreite abgekürzt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TitleCut**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

Optional. TRUE, wenn die Spaltenüberschriften in der Titelleiste bei zu geringer Spaltenbreite abgekürzt werden.

TitleDarkShadowColor (RT Professional)

Beschreibung

Legt die Farbe für die dunkle Seite der Schattierung fest.

Über die Schaltfläche öffnen Sie den Farbauswahl. Die Einstellung ist nur wirksam, wenn die Option "Schattierungsfarbe" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TitleDarkShadowColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für die dunkle Seite der Schattierung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TitleForeColor (RT Professional)

Beschreibung

Legt die Schriftfarbe der Tabellenüberschrift für den gewählten Zustand an.

Über die Schaltfläche öffnen den Farbauswahl-Dialog.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TitleForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Schriftfarbe in der Titelzeile der Tabelle festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TitleGridLineColor (RT Professional)

Beschreibung

Legt die Farbe der Trennlinien in der Titelzeile der Tabelle fest.

Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TitleGridLineColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Trennlinien in der Titelzeile der Tabelle festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TitleLightShadowColor (RT Professional)

Beschreibung

Legt die Farbe für die helle Seite der Schattierung fest.

Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog. Die Einstellung ist nur wirksam, wenn die Option "Schattierungsfarbe" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TitleLightShadowColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für die helle Seite der Schattierung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBScript-Farbkonstanten wie vbRed oder vbGreen verwenden.

TitleSort (RT Professional)

Beschreibung

Legt fest, wie das Sortieren über Spaltenüberschrift ausgelöst wird. Um über Spaltenüberschrift zu sortieren, muss die Option "Auto Scrolling" deaktiviert sein.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TitleSort**[=GridSortTrigger]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

GridSortTrigger

Wert	Beschreibung	Erklärung
0	nein	Das Sortieren über Spaltenüberschrift ist nicht möglich.
1	mit Klick	Das Sortieren wird durch einen Klick auf die Spaltenüberschrift ausgelöst.
2	mit Doppelklick	Das Sortieren wird durch einen Doppelklick auf die Spaltenüberschrift ausgelöst.

TitleStyle (RT Professional)

Beschreibung

Legt fest, ob eine Schattierungsfarbe für die Tabellenüberschrift verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TitleStyle**[=GridColumnStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

GridColumnStyle

Wert	Beschreibung	Erklärung
0	Flach	Eine Schattierungsfarbe wird nicht verwendet. Flach wirkende Darstellung der Überschrift.
1	Button	Eine Schattierungsfarbe wird verwendet. Räumlich wirkende Darstellung der Überschrift.

Toggle (RT Professional)

Beschreibung

Legt fest, ob das angegebene Objekt in Runtime nach dem Betätigen einrastet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Toggle**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton

BOOLEAN

Optional. TRUE, wenn das angegebene Objekt in Runtime nach dem Betätigen einrastet.

Tolerance (RT Professional)

Beschreibung

Legt den Grenzwert für die Speicherplatzanzeige fest, ab dem eine Abweichung gemeldet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Tolerance**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

Int32

Optional. Ein Wert oder eine Konstante, die den Grenzwert für die Speicherplatzanzeige festlegt, ab dem eine Abweichung gemeldet wird.

ToleranceColor (RT Professional)

Beschreibung

Legt die Farben fest, in denen der Balken der Speicherplatzanzeige dargestellt wird, sobald der Toleranzbereich überschritten ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToleranceColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

Color

Optional. Ein Wert oder eine Konstante, welche die Farben festlegt, in denen der Balken der Speicherplatzanzeige dargestellt wird, sobald der Toleranzbereich überschritten ist.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

ToleranceLowerLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Untergrenze für Toleranz 1 fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ToleranceLowerLimit[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die Untergrenze für Toleranz 1 festlegt.

Bemerkungen

Folgende Werte werden über die Eigenschaften "ToleranceLowerLimit", "ToleranceLowerLimitColor" und "ToleranceLowerLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

ToleranceLowerLimitColor (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "ToleranceLowerLimit" fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToleranceLowerLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den unteren Grenzwert "ToleranceLowerLimit" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Die Eigenschaft "ToleranceLowerLimitEnabled" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ToleranceLowerLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Grenzwert "ToleranceLowerLimit" überwacht wird. Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "ToleranceLowerLimit", "ToleranceLowerLimitColor" und "ToleranceLowerLimitRelative" festgelegt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToleranceLowerLimitEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der Grenzwert "ToleranceLowerLimit" überwacht wird.

ToleranceLowerLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert "ToleranceLowerLimit" prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToleranceLowerLimitRelative**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional.

TRUE: Der untere Grenzwert "ToleranceLowerLimit" wird prozentual ausgewertet.

FALSE: Der untere Grenzwert "ToleranceLowerLimit" wird absolut ausgewertet.

ToleranceUpperLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Obergrenze für Toleranz 1 fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToleranceUpperLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die Obergrenze für Toleranz 1 festlegt.

Bemerkungen

Folgende Werte werden über die Eigenschaften "ToleranceUpperLimit", "ToleranceUpperLimitColor" und "ToleranceUpperLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

ToleranceUpperLimitColor (RT Professional)

Beschreibung

Legt die Farbe für den oberen Grenzwert "ToleranceUpperLimit" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToleranceUpperLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den oberen Grenzwert "ToleranceUpperLimit" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

ToleranceUpperLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Grenzwert "ToleranceUpperLimit" überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ToleranceUpperLimitEnabled[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der Grenzwert "ToleranceUpperLimit" überwacht wird.

Bemerkungen

Folgende Werte werden über die Eigenschaften "ToleranceUpperLimit", "ToleranceUpperLimitColor" und "ToleranceUpperLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

ToleranceUpperLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der obere Grenzwert "ToleranceUpperLimit" prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ToleranceUpperLimitRelative[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional.

TRUE: Der obere Grenzwert "ToleranceUpperLimit" wird prozentual ausgewertet.

FALSE: Der obere Grenzwert "ToleranceUpperLimit" wird absolut ausgewertet.

ToolBarAlignment (RT Professional)

Beschreibung

Legt die Position der Symbolleiste fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarAlignment**[=ToolBarPosition]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- SysDiagControl

ToolBarPosition

Optional. Ein Wert oder eine Konstante, welche die Position der Symbolleiste festlegt.

ToolBarBackColor (RT Professional)

Beschreibung

Hintergrundfarbe - ToolBarBackColor

Legt die Hintergrundfarbe der Symbolleiste fest. Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe.

Die projektierte Hintergrundfarbe wird nur angezeigt, wenn die Option "anzeigen" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe der Symbolleiste festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

ToolbarButtonActive (RT Professional)

Beschreibung

Legt fest, ob die mit der Taste verbundene Funktion in Runtime aktiv geschaltet ist. Ein Klick auf die Taste in Runtime löst die zugehörige Funktion aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolbarButtonActive**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die mit der Taste verbundene Funktion ist aktiv.

FALSE: Die mit der Taste verbundene Funktion ist nicht aktiv. Sie können eigene Funktion über lokale Skripte mit der Taste verbinden.

ToolBarButtonAdd (RT Professional)

Beschreibung

Legt eine neue benutzerdefinierte Tastenfunktion an.

Den von WinCC vergebenen Namen können Sie im Feld "Objektname" ändern.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der neuen Tastenfunktion festlegt.

ToolBarButtonBeginGroup (RT Professional)

Beschreibung

Legt fest, ob vor der ausgewählten Tastenfunktion ein Trennzeichen eingefügt wird. Mit den Trennzeichen können Sie die Schaltflächen der Tastenfunktionen gruppieren.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonBeginGroup**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Vor der ausgewählten Tastenfunktion ist das Trennzeichen eingefügt.

FALSE: Vor der ausgewählten Tastenfunktion ist kein Trennzeichen eingefügt

ToolBarButtonCount (RT Professional)

Beschreibung

Legt die Anzahl der in der Symbolleiste enthaltenen Schaltflächen fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Schaltflächen in der Symbolleiste festlegt.

ToolBarButtonEnabled (RT Professional)

Beschreibung

Legt fest, ob eine benutzerdefinierte Taste der Symbolleiste bedienbar ist.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

Optional. TRUE, wenn die ausgewählte benutzerdefinierte Taste der Symbolleiste bedienbar ist.

ToolBarButtonHotKey (RT Professional)

Beschreibung

Legt den Hotkey für die ausgewählte Tastenfunktion fest.

Um einen Hotkey anzulegen oder zu ändern, klicken Sie auf das Feld "Hotkey" und drücken Sie die gewünschte Taste oder Tastenkombination.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonHotKey**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl

- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche den Hotkey für die ausgewählte Tastenfunktion festlegt.

ToolbarButtonID (RT Professional)

Beschreibung

Legt eine eindeutige Ident-Nummer für die gewählte Tastenfunktion fest.

Die Ident-Nummer wird von WinCC vergeben und kann nicht geändert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolbarButtonID**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Ident-Nummer für die ausgewählte Tastenfunktion festlegt.

ToolbarButtonIndex (RT Professional)

Beschreibung

Referenziert eine Tastenfunktion.

Unter Verwendung des Attributs können Sie einer bestimmten Tastenfunktion die Werte anderer Attribute zuweisen.

Gültige Werte für "ToolbarButtonIndex" liegen zwischen 0 und "ToolbarButtonCount" minus 1. Das Attribut "ToolbarButtonCount" gibt die Anzahl der projektierbaren Tastenfunktionen an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer der ausgewählten Tastenfunktion festlegt.

ToolBarButtonLocked (RT Professional)

Beschreibung

Legt für eine benutzerdefinierte Schaltfläche der Symbolleiste fest, ob der eingerastete, gedrückte Zustand der Schaltfläche dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonLocked**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

Optional. TRUE, wenn für die ausgewählte benutzerdefinierte Schaltfläche der Symbolleiste der gedrückte Zustand dargestellt wird.

ToolBarButtonName (RT Professional)

Beschreibung

Legt den Namen der ausgewählten benutzerdefinierten Schaltfläche fest.

Den Namen einer benutzerdefinierten Schaltfläche können Sie ändern. Die Eigenschaft "ToolBarButtonName" für benutzerdefinierte Schaltflächen ist über die Eigenschaft "ToolBarButtonRename" dynamisierbar.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der ausgewählten benutzerdefinierten Schaltfläche festlegt.

ToolBarButtonAuthorization (RT Professional)

Beschreibung

Zeigt die Berechtigung für die ausgewählte Tastenfunktion an. Über die Auswahl Schaltfläche können Sie die Berechtigung ändern. Die Berechtigungen werden in der Benutzerverwaltung projiziert.

Siehe auch

UserArchiveControl (Seite 1528)

TrendRulerControl (Seite 1508)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

OnlineTrendControl (Seite 1443)

ToolBarButtonRemove (RT Professional)

Beschreibung

Entfernt die ausgewählte benutzerdefinierte Schaltfläche.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der zu entfernenden benutzerdefinierten Schaltfläche festlegt.

ToolBarButtonRename (RT Professional)

Beschreibung

Gibt den Namen der benutzerdefinierten Schaltfläche der Symbolleiste zurück, der über die Eigenschaft "ToolBarButtonID" referenziert wird.

Mit "ToolBarButtonRename" dynamisieren Sie auch die Eigenschaft "ToolBarButtonName".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonRename**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den neuen Namen der ausgewählten benutzerdefinierten Schaltfläche festlegt.

ToolBarButtonRepos (RT Professional)

Beschreibung

Ändert die Reihenfolge der benutzerdefinierten Tastenfunktionen in der Symbolleiste.

"Auf" und "Ab" bewegen die ausgewählte Tastenfunktion in der Liste nach oben oder unten. Dadurch wird die Tastenfunktion in der Symbolleiste des Controls weiter vorne oder hinten platziert.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarButtonRepos**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Position der ausgewählten benutzerdefinierten Schaltfläche in der Symbolleiste festlegt.

ToolBarButtonTooltipText (RT Professional)

Beschreibung

Legt den Text für den Tooltip der benutzerdefinierten Schaltfläche in der Symbolleiste fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ToolBarButtonTooltipText[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Text für den Tooltip der ausgewählten benutzerdefinierten Schaltfläche festlegt.

ToolBarButtonUserDefined (RT Professional)

Beschreibung

Legt fest, ob die Taste der Symbolleiste vom Projektteur als neue, benutzerdefinierte Taste hinzugefügt wurde.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ToolBarButtonUserDefined[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl

- TrendRulerControl
- UserArchiveControl

Boolean

TRUE: Die Taste der Symbolleiste ist benutzerdefiniert.

FALSE: Die Taste der Symbolleiste ist vom System vorgegeben.

ToolbarShowTooltips (RT Professional)

Beschreibung

Legt fest, ob in Runtime die Tooltips zu den Tastenfunktionen angezeigt werden. Die Eigenschaft ist mit dem Namen ToolbarShowTooltips dynamisierbar. Die Eigenschaft zum Festlegen des Tooltip-Textes ist "ToolbarButtonTooltipText".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolbarShowTooltips**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Tooltips werden angezeigt.

FALSE: Die Tooltips werden nicht angezeigt

ToolbarUseBackColor (RT Professional)

Beschreibung

Legt fest, ob die Hintergrundfarbe der Symbolleiste angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolbarUseBackColor**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Hintergrundfarbe der Symbolleiste wird angezeigt.

FALSE: Die Hintergrundfarbe der Symbolleiste wird nicht angezeigt.

ToolbarUseHotKeys (RT Professional)

Beschreibung

Legt fest, ob die Hotkeys für die Tastenfunktionen in Runtime aktiviert sind. Die Hotkeys für die Tastenfunktion fügen Sie im Feld "Hotkey" ein.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolbarUseHotKeys**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Hotkeys sind aktiviert.

FALSE: Die Hotkeys sind nicht aktiviert.

ToolBarVisible (RT Professional)

Beschreibung

Legt fest, ob die Symbolleiste des Controls angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolBarVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- FunctionTrendControl
- OnlineTableControl
- OnlineTrendControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Symbolleiste wird angezeigt.

FALSE: Die Symbolleiste wird nicht angezeigt.

ToolTipText (RT Advanced, RT Professional)

Beschreibung

Legt den Text des Tooltips fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ToolTipText**[=STRING]

ToolTipText

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- Bar
- Button
- CheckBox
- Circle

- CircleSegment
- CircularArc
- ComboBox
- Ellipse
- EllipseSegment
- EllipticalArc
- GraphicIOField
- GraphicView
- IOField
- Line
- ListBox
- MultiLineEdit
- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- StatusForce
- Switch
- SymbolicIOField
- TextField
- TubeArcObject

STRING

Optional. Ein Wert oder eine Konstante, welche den Text des Tooltips festlegt.

Top (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Wert der Y-Koordinate des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Top**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- AlarmView
- ApplicationWindow
- Bar
- BatteryView
- Button
- ChannelDiagnose
- CheckBox
- Circle
- CircleSegment
- CircularArc
- Clock
- ComboBox
- DateTimeField
- DiscSpaceView
- Ellipse
- EllipseSegment
- EllipticalArc
- FunctionTrendControl
- Gauge
- GraphicIOField
- GraphicView
- HTML-Browser
- IOField
- Line
- ListBox
- MediaPlayer
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- PLCCodeViewer
- Polygon

- Polyline
- ProtectedAreaView
- RangeLabelView
- RangeQualityView
- RecipeView
- Rectangle
- RoundButton
- Screenwindow
- Slider
- SmartClientView
- StatusForce
- Switch
- SymbolLibrary
- SymbolicIOField
- SysDiagControl
- TextField
- TrendRulerControl
- TrendView
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- UserArchiveControl
- UserView
- WlanQualityView
- WindowsSlider
- ZoneLabelView
- ZoneQualityView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- S7GraphOverview

Int32

Optional. Ein Wert oder eine Konstante, die den Wert der Y-Koordinate in Pixeln enthält (gemessen vom linken oberen Bildrand).

Bemerkungen

Die Y-Koordinate bezieht sich auf die linke obere Ecke des objektumfassenden Rechtecks. In Runtime werden außerdem die Bildgrenzen überwacht. Wenn der zugewiesene Koordinatenwert die Displaygröße übersteigt, wird die benutzerdefinierte Funktion mit einer Fehlermeldung abgebrochen.

TopOffset (RT Professional)

Beschreibung

Legt den Abstand des Bildes vom oberen Rand des Bildfensters fest.

Das Bild wird aus dem Bildfenster ausgeschnitten dargestellt. Die Bildlaufleisten befinden sich am linken und oberen Rand des Bildes. Wenn Sie das Bild im Bildfenster mit der horizontalen und vertikalen Verschiebung der Bildlaufleisten darstellen wollen, verwenden Sie für die Verschiebung die Eigenschaften "HorizontalScrollBarPosition" und "VerticalScrollBarPosition".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TopOffset**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Screenwindow

Int32

Optional. Ein Wert oder eine Konstante, welche den Abstand des Bildes vom oberen Rand des Bildfensters festlegt.

Total (RT Professional)

Beschreibung

Legt die Speicherkapazität fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Total**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiskSpaceView

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die Speicherkapazität festlegt.

Transparency (RT Professional)

Beschreibung

Legt die Transparenz des Objekts in Prozent fest.

Dabei bedeuten 0 = keine Transparenz und 100 = vollständige Transparenz (Unsichtbarkeit).

Die Texte und Felder der grafischen Objekte werden nur beim Wert "100" transparent dargestellt.

In Runtime funktioniert auch ein völlig transparentes und damit unsichtbares Objekt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Transparency**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- CircularArc
- Clock
- ComboBox
- Ellipse
- EllipseSegment
- EllipticalArc
- Gauge
- GraphicIOField
- GraphicView
- IOField
- Line
- ListBox
- MultiLineEdit

- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- Slider
- SymbolicIOField
- TextField
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- WindowsSlider

Int32

Optional. Ein Wert oder eine Konstante, welche die Transparenz des Objekts in Prozent festlegt.

TransparentColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, welche Farbe der zugeordneten Grafik (*.bmp, *.dib) des angegebenen Objekts auf "transparent" gesetzt wird.

Die Eigenschaft "UseTransparentColor" muss den Wert TRUE haben, damit die Farbe transparent dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TransparentColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- GraphicView
- GraphicIOField

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe festlegt, die transparent dargestellt wird.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TransparentColorDeactivatedPicture (RT Professional)

Beschreibung

Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Deaktiviert" auf "transparent" gesetzt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TransparentColorDeactivatedPicture[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- RoundButton

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des zugeordneten Bitmap-Objekts festlegt, die für den Zustand "Deaktiviert" auf "transparent" gesetzt wird.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Die Eigenschaft "PicDeactUseTransColor" muss den Wert TRUE haben, damit die Farbe auf "transparent" gesetzt wird.

TransparentColorPictureOff (RT Professional)

Beschreibung

Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Aus" auf "transparent" gesetzt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TransparentColorPictureOff[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des zugeordneten Bitmap-Objekts festlegt, die für den Zustand "Aus" auf "transparent" gesetzt wird.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TransparentColorPictureOn (RT Professional)

Beschreibung

Legt fest, welche Farbe des zugeordneten Bitmap-Objekts für den Zustand "Ein" auf "transparent" gesetzt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TransparentColorPictureOn[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des zugeordneten Bitmap-Objekts festlegt, die für den Zustand "Ein" auf "transparent" gesetzt wird.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Die Eigenschaft "PicDownUseTransColor" muss den Wert TRUE haben, damit die Farbe auf "transparent" gesetzt wird.

TrendActualize (RT Professional)

Beschreibung

Legt fest, ob die ausgewählte Kurve aktualisiert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendActualize**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die ausgewählte Kurve wird immer aktualisiert.

FALSE: Die ausgewählte Kurve wird nicht aktualisiert. Diese Einstellung ist sinnvoll, wenn eine archivierte Kurve mit einer aktuellen Kurve verglichen wird.

TrendAdd (RT Professional)

Beschreibung

Legt eine neue Kurve an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Der Name der neuen Kurve.

TrendBeginTime (RT Professional)

Beschreibung

Legt den Anfangszeitpunkt des Zeitbereichs für die Datenversorgung der ausgewählten Kurve fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendBeginTime**[=DateTime]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

DateTime

Optional. Legt den Anfangszeitpunkt für die Datenversorgung der ausgewählten Kurve fest.

TrendColor (RT Professional)

Beschreibung

Legt die Farbe des Objekts fest oder gibt sie zurück.

Die Trendanzeige stellt die Tendenz (steigend oder fallend) des zu überwachenden Messwerts mit einem kleinen Pfeil dar. Um die Trendanzeige zu aktivieren, muss die Eigenschaft "Trend" auf "TRUE" gesetzt sein.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Objekts festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendCount (RT Professional)

Beschreibung

Legt die Anzahl der projizierten Kurven fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TrendCount[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Kurven des angegebenen Objekts festlegt.

TrendEndTime (RT Professional)

Beschreibung

Legt den Endzeitpunkt des Zeitbereichs für die Datenanbindung der ausgewählten Kurve fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendEndTime**[=DateTime]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

DateTime

Optional. Legt den Endzeitpunkt für die Datenversorgung der ausgewählten Kurve fest.

TrendExtendedColorSet (RT Professional)

Beschreibung

Legt fest, ob Sie die Punktfarbe und die Füllfarbe projektieren können und ob die Farben in Runtime dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendExtendedColorSet**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Einstellungen in den Feldern "Punktfarbe" und "Füllfarbe" sind projektierbar und in Runtime wirksam.

FALSE: Die Einstellungen in den Feldern "Punktfarbe" und "Füllfarbe" sind nicht projektierbar und in Runtime nicht wirksam.

Siehe auch

FunctionTrendControl (Seite 1398)

OnlineTrendControl (Seite 1443)

TrendFill (RT Professional)

Beschreibung

Legt fest, ob die Fläche unterhalb der Kurve gefüllt dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendFill**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Fläche unterhalb der Kurve wird gefüllt dargestellt. Wenn die Option "Erweitert" nicht aktiviert ist, wird die Kurvenfarbe als Füllfarbe verwendet.

FALSE: Die Kurve wird nicht gefüllt dargestellt..

TrendFillColor (RT Professional)

Beschreibung

Legt die Füllfarbe der Kurve fest.

Die Füllfarbe wird verwendet, wenn die Option "Gefüllt" aktiviert ist. Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe.

Die Projektierung ist nur möglich, wenn die Option "Erweitert" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendFillColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Füllfarbe der ausgewählten Kurve festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendIndex (RT Professional)

Beschreibung

Referenziert eine projizierte Kurve. Unter Verwendung dieser Eigenschaft können Sie einer bestimmten Kurve die Werte anderer Eigenschaften zuweisen. Der Index muss immer gesetzt werden, bevor Sie die Eigenschaften einer Kurve in Runtime ändern.

Gültige Werte für "TrendIndex" liegen zwischen 0 und "TrendCount" minus 1. Die Eigenschaft "TrendCount" gibt die Anzahl der projizierten Kurven an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer der ausgewählten Kurve festlegt.

TrendIndicatorColor (RT Professional)

Beschreibung

Legt die Farbe der Trendanzeige fest. Die Trendanzeige stellt mit einem kleinen Pfeil die Tendenz (steigend oder fallend) des zu überwachenden Messwertes dar. Um die Trendanzeige zu aktivieren, muss die Eigenschaft "ShowTrendIndicator" den Wert "TRUE" haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendIndicatorColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Trendanzeige festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendLabel (RT Professional)

Beschreibung

Legt die Bezeichnung der ausgewählten Kurve fest. Die Bezeichnung wird in Runtime angezeigt, wenn das Attribut "UseTrendNameAsLabel" den Wert "FALSE" hat.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendLabel**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Bezeichnung der ausgewählten Kurve festlegt.

TrendLineStyle (RT Professional)

Beispiel

Legt fest, welche Linienart zur Darstellung der Kurve verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendLineStyle**[=LineStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

LineStyle

Wert	Beschreibung	Erklärung
0	durchgezogen	Die Kurve wird durchgezogen dargestellt.
1	gestrichelt	Die Kurve wird gestrichelt dargestellt.
2	Punkte	Die Kurve wird mit einer punktierten Linie dargestellt.
3	strichpunktiert	Die Kurve wird mit einer strichpunktierten Linie dargestellt.
4	Strich-Punkt-Punkt	Die Kurve wird mit einer Strich-Punkt-Punkt-Linie dargestellt.

TrendLineType (RT Professional)

Beschreibung

Legt fest, wie die Kurve dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendLineType**[=TrendLineTypeScada]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

FunctionTrendControl

OnlineTrendControl

TrendLineTypeScada

Wert	Beschreibung	Erklärung
0	keine	Nur die Punkte werden dargestellt.
1	Punkte linear verbinden	Eine Kurve mit den linear verbundenen Punkten wird dargestellt.
2	Treppenkurve	Eine Treppenkurve mit den verbundenen Punkten wird dargestellt.

TrendLineWidth (RT Professional)

Beschreibung

Legt die Linienstärke der ausgewählten Kurve in Pixeln fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendLineWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Linienstärke der ausgewählten Kurve in Pixeln festlegt.

TrendLowerLimit (RT Professional)

Beschreibung

Gibt den unteren Grenzwert für eine Variable an. Wenn die Variable den Wert von "TrendLowerLimit" unterschreitet, werden die Werte mit der in "TrendLowerLimitColor" eingestellten Farbe gekennzeichnet. Die Angabe ist wirksam, wenn die Eigenschaft "TrendLowerLimitColoring" den Wert "TRUE" hat.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendLowerLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den unteren Grenzwert für eine Variable festlegt.

TrendLowerLimitColor (RT Professional)

Beschreibung

Legt die Farbe fest, die Variablenwerte kennzeichnet, die unterhalb des Werts von "TrendLowerLimit" liegen. Die Einstellung ist wirksam, wenn das Attribut "TrendLowerLimitColoring" den Wert "TRUE" hat.

Syntax

Object.**TrendLowerLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Variablenwerte unterhalb des Werts von "TrendLowerLimit" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendLowerLimitColoring (RT Professional)

Beschreibung

Legt fest, ob das Attribut "TrendLowerLimitColor" verwendet wird, um die Variablenwerte zu kennzeichnen, die unterhalb des Werts von "TrendLowerLimit" liegen.

Syntax

Object.**TrendLowerLimitColoring**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Eigenschaft "TrendLowerLimitColor" ist wirksam.

FALSE: Die Eigenschaft "TrendLowerLimitColor" ist unwirksam.

TrendMeasurePoints (RT Professional)

Beschreibung

Legt die Anzahl der Messpunkte fest, die zur Darstellung der ausgewählten Kurve verwendet werden.

Legt die Anzahl der Wertepaare fest, wenn die Kurve über ein Anwenderarchiv versorgt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendMeasurePoints**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Messpunkte bzw. der Wertepaare der ausgewählten Kurve festlegt.

TrendName (RT Professional)

Beschreibung

Zeigt den Namen der ausgewählten Kurve an. Den Namen legen Sie auf der Registerkarte "Kurven" fest. Die Eigenschaft "TrendName" ist über die Eigenschaft "TrendRename" dynamisierbar

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der ausgewählten Kurve festlegt.

TrendPointColor (RT Professional)

Beschreibung

Gibt die Farbe der Punkte der ausgewählten Kurve an.

Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe.

Die Projektierung ist nur möglich, wenn die Option "Erweitert" aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendPointColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Punkte der ausgewählten Kurve festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendPointStyle (RT Professional)

Beschreibung

Legt fest, wie die Punkte auf der Kurve dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendPointStyle**[=PointStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

PointStyle

Wert	Beschreibung	Erklärung
0	kein	Die Punkte werden nicht dargestellt.
1	Punkte	Die Punkte der Kurve werden als 1 Pixel große Punkte dargestellt. Die Einstellung im Feld "Punktbreite" ist nicht wirksam.
2	Quadrate	Die Punkte werden als Quadrate dargestellt. Die Einstellung im Feld "Punktbreite" ist wirksam.
3	Kreise	Die Punkte werden als Kreise dargestellt. Die Einstellung im Feld "Punktbreite" ist wirksam.

TrendPointWidth (RT Professional)

Beschreibung

Legt die Punktbreite in Pixeln fest. Die Punktbreite können Sie nur für die Punktarten "Quadrate" und "Kreise" projektieren.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendPointWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Punktbreite der ausgewählten Kurve in Pixeln festlegt.

TrendProvider (RT Professional)

Beschreibung

Legt die Datenversorgung der ausgewählten Kurve fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendProvider**[=Provider]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Provider

Wert	Beschreibung	Erklärung
0	Keine	Keine projektierte Datenversorgung, um eine Anbindung in Runtime mit einer benutzerdefinierten Funktion herzustellen.
1	Archivvariablen	Datenversorgung mit Archivvariablen eines Prozesswertarchivs.
2	HMI-Variablen	Datenversorgung mit Variablenwerten von HMI-Variablen.
3	Rezepturdaten	Datenversorgung mit Spalten einer Rezeptur.

TrendRangeType (RT Professional)

Beschreibung

Legt den Zeitbereich fest, in dem die ausgewählte Kurve mit Daten versorgt wird.

Bei einer Datenversorgung über Anwenderarchive können Sie nur die Anzahl der Messpunkte festlegen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendRangeType**[=RangeType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

RangeType

Wert	Beschreibung	Erklärung
0	Zeitbereich	Für die Datenanbindung werden Anfangszeitpunkt und Zeitbereich festgelegt.
1	Anfangs- bis Endzeitpunkt	Für die Datenanbindung werden Anfangs- und Endzeitpunkt festgelegt.
2	Anzahl der Messpunkte	Für die Datenanbindung werden Anfangszeitpunkt und Anzahl der Messpunkte festgelegt.

TrendRemove (RT Professional)

Beschreibung

Entfernt die ausgewählte Kurven aus der Liste.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der zu entfernenden Kurve festlegt.

TrendRename (RT Professional)

Beschreibung

Ändert den Namen der Kurve, die über die Eigenschaft "TrendIndex" referenziert wird.
Mit "TrendRename" dynamisieren Sie auch die Eigenschaft "TrendName".
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendRename**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den neuen Namen der ausgewählten Kurve festlegt.

TrendRepos (RT Professional)

Beschreibung

Ändert die Reihenfolge der ausgewählten Kurve im Kurvenfenster.

"Auf" und "Ab" bewegen die ausgewählte Kurve in der Liste nach oben oder unten. Dadurch wird die Kurve in Runtime weiter im Vordergrund oder Hintergrund dargestellt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendRepos**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die neue Position der ausgewählten Kurve festlegt.

TrendSelectTagNameX (RT Professional)

Beschreibung

Öffnet den Dialog zur Auswahl des Variablennamens für die Datenversorgung der X-Achse in der f(x)-Kurvenanzeige. Die Eigenschaft können Programmierer nutzen, um z. B. über eine Schaltfläche den Benutzer einen Variablennamen auswählen zu lassen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendSelectTagNameX**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE:

FALSE:

TrendSelectTagNameY (RT Professional)

Beschreibung

Öffnet den Dialog zur Auswahl des Variablennamens für die Datenversorgung der Y-Achse in der f(x)-Kurvenanzeige. Die Eigenschaft können Programmierer nutzen, um z. B. über eine Schaltfläche den Benutzer einen Variablennamen auswählen zu lassen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendSelectTagNameY**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE:

FALSE:

TrendTagNameX (RT Professional)

Beschreibung

Legt den Namen der angebenen HMI-Variablen bzw. Spalte für die X-Achse fest. Über die Auswahl Schaltfläche wählen Sie eine HMI-Variable oder eine Spalte aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendTagNameX**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der HMI-Variablen bzw. Spalte für die X-Achse festlegt.

TrendTagNameY (RT Professional)

Beschreibung

Legt den Namen der angebotenen HMI-Variablen bzw. Spalte für die Y-Achse fest. Über die Auswahlfläche wählen Sie eine HMI-Variable oder eine Spalte aus.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendTagNameY**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der HMI-Variablen bzw. Spalte für die Y-Achse festlegt.

TrendTimeRangeBase (RT Professional)

Beschreibung

Legt die Zeiteinheit zur Bestimmung des Zeitbereichs fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendTimeRangeBase**[=TagLoggingTimeUnit]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

TagLoggingTimeUnit

Wert	Beschreibung
500	500 ms
1000	1 Sekunde
60000	1 Minute
3600000	1 Stunde
86400000	1 Tag

TrendTimeRangeFactor (RT Professional)

Beschreibung

Legt den Faktor zur Bestimmung des Zeitbereichs fest. Nur ganzzahlige Faktoren sind zulässig.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeRangeFactor**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche den Faktor zur Bestimmung des Zeitbereichs festlegt.

TrendTrendWindow (RT Professional)

Beschreibung

Legt fest, in welchem Kurvenfenster die ausgewählte Kurve dargestellt wird.

Die zur Verfügung stehenden Kurvenfenster legen Sie auf der Registerkarte "Kurvenfenster" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendTrendWindow**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen des Kurvenfensters für die ausgewählte Kurve festlegt.

TrendUncertainColor (RT Professional)

Beschreibung

Werte haben einen unsicheren Status, wenn der Anfangswert nach dem Aktivieren von Runtime unbekannt ist oder ein Ersatzwert verwendet wird.

Mit der Eigenschaft "TrendUncertainColor" legen Sie die Farbe fest, die für die Kennzeichnung dieser Werte verwendet wird. Ob die Angabe ausgewertet wird, hängt von der Eigenschaft "TrendUncertainColoring" ab.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendUncertainColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Werte mit unsicherem Status der ausgewählten Kurve festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendUncertainColoring (RT Professional)

Beschreibung

Werte haben einen unsicheren Status, wenn der Anfangswert nach dem Aktivieren von Runtime unbekannt ist oder ein Ersatzwert verwendet wird. Über "TrendUncertainColoring" legen Sie fest, ob derartige Werte, mit der in "TrendUncertainColor" eingestellten Farbe gekennzeichnet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendUncertainColoring**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Einstellung der Eigenschaft "TrendUncertainColor" ist wirksam.

FALSE: Die Einstellung der Eigenschaft "TrendUncertainColor" ist unwirksam.

TrendUpperLimit (RT Professional)

Beschreibung

Gibt den oberen Grenzwert für eine Variable an.

Wenn die Variable den Wert von "TrendUpperLimit" unterschreitet, werden die Werte mit der in "TrendUpperLimitColor" eingestellten Farbe dargestellt. Die Angabe ist wirksam, wenn das Attribut "TrendUpperLimitColoring" den Wert "TRUE" hat.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendUpperLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den oberen Grenzwert für die Werte der ausgewählten Kurve festlegt.

TrendUpperLimitColor (RT Professional)

Beschreibung

Legt die Farbe fest, die Variablenwerte kennzeichnet, die unterhalb des Werts von "TrendLowerLimit" liegen. Die Einstellung ist wirksam, wenn das Attribut "TrendUpperLimitColoring" den Wert "TRUE" hat.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TrendUpperLimitColor[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der für die Werte unterhalb des Grenzwerts der ausgewählten Kurve festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendUpperLimitColoring (RT Professional)

Beschreibung

Legt fest, ob der Auswahlrahmen mit der vom System festgelegten Farbe dargestellt wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TrendUpperLimitColoring[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Einstellung des Attributs "TrendUpperLimitColor" ist wirksam.

FALSE: Die Einstellung des Attributs "TrendUpperLimitColor" ist unwirksam.

TrendVisible (RT Professional)

Beschreibung

Legt fest, ob die ausgewählte Kurve angezeigt wird.

In der Liste werden die Kurven aufgelistet, die Sie angelegt haben.

Aktivieren Sie in der Liste die Kurven, die Sie in den Kurvenfenstern anzeigen wollen.

Klicken Sie auf eine Kurve in der Liste, um die Eigenschaften anzupassen und um Achsen und Kurvenfenster der Kurve zuzuordnen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die ausgewählte Kurve wird angezeigt.

FALSE: Die ausgewählte Kurve wird nicht angezeigt.

TrendWindowAdd (RT Professional)

Beschreibung

Legt eine neue Kurvenanzeige an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der neuen Kurvenanzeige festlegt.

TrendWindowCoarseGrid (RT Professional)

Beschreibung

Legt fest, ob die Gitternetzlinien für die Hauptskalierung angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowCoarseGrid**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Gitternetzlinien für die Hauptskalierung werden angezeigt.

FALSE: Die Gitternetzlinien für die Hauptskalierung werden nicht angezeigt.

TrendWindowCoarseGridColor (RT Professional)

Beschreibung

Gibt die Farbe der Gitternetzlinien für die Hauptskalierung an.

Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowCoarseGridColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Gitternetzlinien für die Hauptskalierung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendWindowFineGrid (RT Professional)

Beschreibung

Legt fest, ob die Gitternetzlinien für die Hilfsskalierung angezeigt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowFineGrid**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Gitternetzlinien für die Hilfsskalierung werden angezeigt.

FALSE: Die Gitternetzlinien für die Hilfsskalierung werden nicht angezeigt.

TrendWindowFineGridColor (RT Professional)

Beschreibung

Gibt die Farbe der Gitternetzlinien für die Hilfsskalierung an.
Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowFineGridColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der Gitternetzlinien für die Hilfsskalierung festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendWindowForegroundTrendGrid (RT Professional)

Beschreibung

Legt fest, ob nur die Gitternetzlinien für die Vordergrundkurve im ausgewählten Kurvenfenster angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowForegroundTrendGrid**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die Gitternetzlinien für die Vordergrundkurve werden im Kurvenfenster angezeigt.

FALSE: Die Gitternetzlinien für alle Kurven werden im Kurvenfenster angezeigt.

TrendWindowGridInTrendColor (RT Professional)

Beschreibung

Legt fest, ob die Gitternetzlinien für die Hauptskalierung in der Kurvenfarbe dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowGridInTrendColor**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die Gitternetzlinien werden in der Kurvenfarbe dargestellt.

FALSE: Die Gitternetzlinien werden mit der im Feld "Farbe" eingestellten Farbe dargestellt.

TrendWindowHorizontalGrid (RT Professional)

Beschreibung

Legt fest, ob die horizontalen Gitternetzlinien angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowHorizontalGrid**[=BOOLEAN]

Object

- FunctionTrendControl

BOOLEAN

TRUE: Die horizontalen Gitternetzlinien werden angezeigt.

FALSE: Die horizontalen Gitternetzlinien werden nicht angezeigt.

TrendWindowIndex (RT Professional)

Beschreibung

Referenziert ein projektiertes Kurvenfenster. Unter Verwendung der Eigenschaft können Sie einem bestimmten Kurvenfenster die Werte anderer Eigenschaften zuweisen.

Gültige Werte für "TrendWindowIndex" liegen zwischen 0 und "TrendWindowCount" minus 1. Die Eigenschaft "TrendWindowCount" gibt die Anzahl der projektierten Kurvenfenster an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowIndex**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer des ausgewählten Kurvenfensters festlegt.

TrendWindowName (RT Professional)

Beschreibung

Legt den Namen des ausgewählten Kurvenfensters fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen des ausgewählten Kurvenfensters festlegt.

TrendWindowRemove (RT Professional)

Beschreibung

Entfernt die ausgewählte Kurvenanzeige aus der Liste.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen des zu entfernenden Kurvenfensters festlegt.

TrendWindowRename (RT Professional)

Beschreibung

Ändert den Namen des Kurvenfensters, das über die Eigenschaft "TrendWindowIndex" referenziert wird.

Mit "TrendWindowRename" dynamisieren Sie auch die Eigenschaft "TrendWindowName".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TrendWindowRename[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den neuen Namen der ausgewählten Kurvenanzeige festlegt.

TrendWindowRepos (RT Professional)

Beschreibung

Ändert die Reihenfolge der Kurvenanzeige.

"Auf" und "Ab" bewegen die ausgewählte Kurve in der Liste nach oben oder unten. Die Reihenfolge in der Liste bestimmt die Position im Control. Die erste Kurvenanzeige wird an der untersten Position, die letzte Kurvenanzeige an der obersten Position dargestellt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindwRepos**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die neue Position der ausgewählten Kurvenanzeige festlegt.

TrendWindowRulerColor (RT Professional)

Beschreibung

Gibt die Farbe des Lineals an.

Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe. Die Farbe ist projektierbar und darstellbar, wenn für die Darstellung "1 - grafisch" eingestellt ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowRulerColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Lineals festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

TrendWindowRulerLayer (RT Professional)

Beschreibung

Legt die Darstellungsebene des Lineals in der Kurvenanzeige fest..
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowRulerLayer**[=RulerLayer]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

RulerLayer

Wert	Beschreibung	Erklärung
0	hinter Gitternetz	Das Lineal wird hinter das Gitternetz gelegt.
1	hinter Kurven	Das Lineal wird hinter die Kurven und vor das Gitternetz gelegt.
2	vor Kurven	Das Lineal wird vor die Kurven gelegt.

TrendWindowRulerStyle (RT Professional)

Beschreibung

Legt die Darstellung des Lineals fest..
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowRulerStyle**[=RulerStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

RulerStyle

TRUE: Das Lineal wird als eine einfache schwarze Linie dargestellt..

FALSE: Das Lineal wird in der projektierten "Farbe" und "Stärke" dargestellt.

TrendWindowRulerWidth (RT Professional)

Beschreibung

Legt die Stärke des Lineals in Pixeln fest.

Die Stärke ist projektierbar und darstellbar, wenn für die Darstellung "1 - grafisch" eingestellt ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowRulerWidth**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Stärke des Lineals in Pixeln festlegt.

TrendWindowSpacePortion (RT Professional)

Beschreibung

Legt den Anteil des ausgewählten Kurvenfensters bei der Darstellung im Control fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowSpacePortion**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche den Anteil des ausgewählten Kurvenfensters im Control in Prozent festlegt.

TrendWindowVerticalGrid (RT Professional)

Beschreibung

Legt fest, ob die vertikalen Gitternetzlinien angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowVerticalGrid**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Die vertikalen Gitternetzlinien werden angezeigt.

FALSE: Die vertikalen Gitternetzlinien werden nicht angezeigt.

TrendWindowVisible (RT Professional)

Beschreibung

Legt fest, ob das ausgewählte Kurvenfenster angezeigt wird.

In der Liste werden die Kurvenanzeigen aufgelistet, die Sie angelegt haben.

Aktivieren Sie in der Liste die Kurvenanzeigen, die Sie im Control anzeigen wollen.

Klicken Sie auf einen Eintrag in der Liste, um die Eigenschaften des Lineals und die Gitternetzlinien anzupassen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendWindowVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Das ausgewählte Kurvenfenster wird angezeigt.

FALSE: Das ausgewählte Kurvenfenster wird nicht angezeigt.

TrendXAxis (RT Professional)

Beschreibung

Legt fest, welche X-Achse für die ausgewählte Kurve verwendet wird.

Die zur Verfügung stehenden X-Achsen legen Sie auf der Registerkarte "X-Achsen" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendXAxis**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der für die ausgewählte Kurve verwendeten X-Achse festlegt.

TrendYAxis (RT Professional)

Beschreibung

Legt fest, welche Y-Achse für die ausgewählte Kurve verwendet wird.

Die zur Verfügung stehenden Y-Achsen legen Sie auf der Registerkarte "Y-Achsen" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TrendYAxis**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der für die ausgewählte Kurve verwendeten Y-Achse festlegt.

3.5.5.14 **Eigenschaften U-W (Panels, Comfort Panels, RT Advanced, RT Professional)**

Unit (RT Professional)

Beschreibung

Legt die Maßeinheit fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Unit**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- IOField

STRING

Optional. Ein Wert oder eine Konstante, welche die Maßeinheit festlegt.

UnitColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Textfarbe für die Bezeichnung der Maßeinheit im Objekt "Gauge" fest
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UnitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Textfarbe für die Maßeinheit festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

UnitText (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Text für die Maßeinheit des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UnitText**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

STRING

Optional. Ein Wert oder eine Konstante, die den Text für die Maßeinheit festlegt.

Bemerkungen

Geben Sie einen Text ein, um z.B. die physikalische Einheit des dargestellten Werts anzuzeigen. Standardmäßig ist hier kein Text voreingestellt.

UnitTop (RT Professional)

Beschreibung

Legt den Abstand der Maßeinheit zur Oberkante des angegebenen Objekts fest. Der Schriftzug ist nur längs des vertikalen Durchmessers der Skalenscheibe positionierbar. Der Wert der Eigenschaft ist bezogen auf die Höhe des angegebenen Objekts und wird gemessen von der Oberkante des angegebenen Objekts zur Unterkante des Schriftzugs.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UnitTop**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den Abstand der Maßeinheit zur Oberkante des angegebenen Objekts festlegt.

Wertebereich 0 bis 1

0: Die Unterkante des Schriftzugs liegt auf der oberen Begrenzung des angegebenen Objekts. Der Text ist nicht mehr sichtbar, da er außerhalb des angegebenen Objekts liegt.

1: Die Unterkante des Schriftzugs liegt auf der unteren Begrenzung des angegebenen Objekts.

UpperLimit (RT Professional)

Beschreibung

Legt den oberen Grenzwert für Eingabewerte fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UpperLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den oberen Grenzwert für Eingabewerte festlegt.

Used (RT Professional)

Beschreibung

Gibt die Größe des belegten Speicherplatzes zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Used**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die Größe des belegten Speicherplatzes zurück gibt.

UseDesignColorSchema (RT Professional)

Beschreibung

Legt fest, ob für dieses Objekt die im globalen Farbschema des aktuellen Designs definierten Farben verwendet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseDesignColorSchema**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar
- Button
- CheckBox
- Circle
- CircleSegment
- CircularArc
- Clock
- ComboBox
- Ellipse
- EllipseSegment
- EllipticalArc
- Gauge
- GraphicView
- IOField
- Line
- ListBox
- MultiLineEdit

- OptionGroup
- Polygon
- Polyline
- Rectangle
- RoundButton
- Slider
- SymbolicIOField
- TextField
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- WindowsSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- AlarmView
- DateTimeField
- RecipeView
- StatusForce
- Switch
- SysDiagControl
- TrendView
- UserView

BOOLEAN

Optional. TRUE, wenn das Objekt mit den Farben aus dem für diesen Objekttyp festgelegten globalen Farbschema dargestellt wird.

FALSE, wenn das Objekt mit den Farben entsprechend den Einstellungen im Objekt dargestellt wird.

UseDesignShadowSettings (RT Professional)

Beschreibung

Legt fest, ob das Objekt mit der im aktiven Design festgelegten Schattierung dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseDesignShadowSettings**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Circle
- CircleSegment
- CircularArc
- Ellipse
- EllipseSegment
- EllipticalArc
- GraphicView
- Line
- Polygon
- Polyline
- Rectangle
- TextField
- TubeArcObject
- TubeDoubleTeeObject
- Tubepolyline
- TubeTeeObject
- Bar
- Button
- CheckBox
- Clock
- ComboBox
- Gauge
- GraphicIOField
- IOField
- ListBox
- MultiLineEdit
- OptionGroup
- RoundButton
- Slider
- SymbolicIOField
- WindowsSlider

BOOLEAN

TRUE: Das Objekt wird mit der für diesen Objekttyp festgelegten globalen Schattierung dargestellt.

FALSE: Das Objekt wird ohne Schatten dargestellt.

UsedPercent (RT Professional)

Beschreibung

Legt die Messwerte für den belegten Speicherplatz in Prozent fest. Die Werte können in Runtime abgefragt werden. Die Werte können nicht vordefiniert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UsedPercent**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiscSpaceView

Int32

Optional. Ein Wert oder eine Konstante, welche die Messwerte für den belegten Speicherplatz in Prozent zurückgibt.

UseExponentialFormat (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob die Zahlen mit Exponenten (z. B. "1,00e+000") dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseExponentialFormat**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn die Zahlen mit Exponenten (z. B. "1,00e+000") dargestellt werden.

UseFlashTransparentColor (RT Professional)

Beschreibung

Legt fest, ob die Farbe des Bitmap-Objektes eines Blinkbildes auf "transparent" gesetzt wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseFlashTransparentColor**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "GraphicIOField".

BOOLEAN

Optional. TRUE, wenn die Farbe des Bitmap-Objektes eines Blinkbildes auf "transparent" gesetzt wird.

Siehe auch

GraphicIOField (Seite 1409)

UseMessageColor (RT Professional)

Beschreibung

Legt fest, ob die vereinbarten Farben der Meldeklassen angezeigt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseMessageColor** [=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl

BOOLEAN

TRUE: Die Farben werden angezeigt.

FALSE: Die Farbeinstellungen auf der Registerkarte "Darstellung" sind wirksam, die für den Tabelleninhalt angegeben sind.

UseSelectedTitleColor (RT Professional)

Beschreibung

Legt fest, ob eine Markierungsfarbe für die Tabellenüberschriften von markierten Tabellenzellen verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseSelectedTitleColor**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Eine Markierungsfarbe wird verwendet. Die Einstellungen zu "Hintergrund" und "Schrift" sind in Runtime wirksam.

FALSE: Eine Markierungsfarbe wird nicht verwendet. Die Einstellungen zu "Hintergrund" und "Schrift" sind in Runtime nicht wirksam.

UseTableColor2 (RT Professional)

Beschreibung

Legt fest, ob eine zweite Zeilenfarbe bei der Darstellung der Tabelle verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseTableColor2**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Die Einstellungen der "Zeilenfarbe 2" werden im Wechsel mit der "Zeilenfarbe 1" verwendet.

FALSE: Die Einstellungen der "Zeilenfarbe 1" werden für alle Zeilen verwendet.

UseTagLimitColors (RT Professional)

Beschreibung

Legt fest, dass die Farben verwendet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseTagLimitColors**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- IOField

BOOLEAN

Optional. TRUE, wenn die Farben verwendet werden.

UseTransparentColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob die mit der Eigenschaft "TransparentColor" festgelegte Farbe des angegebenen Objekts transparent dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseTransparentColor**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- GraphicView
- GraphicIOField

BOOLEAN

Optional. TRUE, wenn die festgelegte Farbe transparent dargestellt wird.

UseTransparentColorDeactivatedPicture (RT Professional)

Beschreibung

Legt fest, ob die mit der Eigenschaft "TransparentColorDeactivatedPicture" festgelegte Transparentfarbe für den Zustand "Deaktiviert" verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseTransparentColorDeactivatedPicture**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- RoundButton

BOOLEAN

Optional. TRUE, wenn die mit der Eigenschaft "TransparentColorDeactivatedPicture" festgelegte Transparentfarbe für den Zustand "Deaktiviert" verwendet wird.

UseTransparentColorPictureOff (RT Professional)

Beschreibung

Legt fest, ob die mit der Eigenschaft "TransparentColorPictureOff" festgelegte Transparentfarbe für den Zustand "Aus" verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseTransparentColorPictureOff**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton

BOOLEAN

Optional. TRUE, wenn die mit der Eigenschaft "TransparentColorPictureOff" festgelegte Transparentfarbe für den Zustand "Aus" verwendet wird.

UseTransparentColorPictureOn (RT Professional)

Beschreibung

Legt fest, ob die mit der Eigenschaft "TransparentColorPictureOn" festgelegte Transparentfarbe für den Zustand "Ein" verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.UseTransparentColorPictureOn[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton

BOOLEAN

Optional. TRUE, wenn die mit der Eigenschaft "TransparentColorPictureOn" festgelegte Transparentfarbe für den Zustand "Ein" verwendet wird.

UseTrendNameAsLabel (RT Professional)

Beschreibung

Legt fest, ob die Eigenschaft "Name" oder "Beschriftung" oder als Bezeichnung der Kurve in Runtime verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.UseTrendNameAsLabel[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl
- OnlineTrendControl

BOOLEAN

TRUE: Der unter "Eigenschaften > Eigenschaften > Kurven > Name" projektierte Name wird verwendet.

FALSE: Der unter "Eigenschaften > Eigenschaften > Kurven > Beschriftung" projektierte Name wird verwendet.

UserName (RT Professional)

Beschreibung

Gibt den Namen des Users zurück, der das Alarm-Objekt ausgelöst hat.

UV_ColumnWidth_AKZ (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_Descriptor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_InstallationDate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_LADDR (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_Name (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_OKZ (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_OperationState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_OrderID (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_ProfileID (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_Rack (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_Slot (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_SoftwareRevision (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_SpecificProfileData (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_State (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_Station (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_SubAddress (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_SubSlot (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_SubSystem (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ColumnWidth_Type (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_AKZ (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_Descriptor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_InstallationDate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_LADDR (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_Name (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_OKZ (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_OperationState (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_OrderID (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_ProfileID (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_Rack (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_Slot (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_SoftwareRevision (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_SpecificProfileData (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_State (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_Station (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_SubAddress (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_SubSlot (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_SubSystem (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

UV_ShowItem_Type (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Kein Zugriff in Runtime.

Value (RT Professional)

Beschreibung

Legt abhängig vom verwendeten Objekt einen Wert fest oder gibt ihn zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Value**[=VARIANT]

Object

Erforderlich. Ein Objekt vom Typ "Tag", "DataItem" oder "ScreenItem" mit der Ausprägung "Gauge".

VARIANT

Optional. Der Wert, der abhängig vom verwendeten Objekt festgelegt wird:

- `Tag.Value`: Gibt den Variablenwert beim letzten Lesezugriff zurück oder legt den zukünftigen Variablenwert fest. Um aus der Eigenschaft "Value" den Variablenwert auszulesen, verwenden Sie die Methode "Read". Mit der Methode "Write" weisen Sie der Eigenschaft "Value" einen neuen Variablenwert zu.
- `Dataset.Value`: Legt einen Wert fest oder gibt eine Kopie des Werts oder die Objektreferenz zurück. Achten Sie beim Zurückgeben von Objektreferenzen darauf, dass die Objektreferenz multithreadfähig sind.
- `ScreenItem("Gauge_1").Value`: Legt den Wert fest, auf den der Zeiger des Zeigerinstruments zeigt. Wertebereich innerhalb der durch die Eigenschaften "ValueMin" und "ValueMax" gesetzten Werte.

Beispiele

Das folgende Beispiel schreibt einen neuen Wert in die Variable "Tag1":

```
'VBS94
Dim objTag
Set objTag = HMIRuntime.Tags("Tag1")
objTag.Value = 50
objTag.Write
```

Das Beispiel zeigt, wie Sie einen Wert in eine Variablen-Auflistung aufnehmen und als Trace ausgeben. Anschließend wird der Wert geändert, erneut ausgegeben und dann entfernt. Sinnvollerweise geschieht dies in verschiedenen Aktionen:

```
'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")
```

Siehe auch

[Dataltem \(Seite 1308\)](#)

[Tag \(Seite 1332\)](#)

ValueAxisAutoRange(i) (RT Professional)

Beschreibung

Legt fest, ob der Wertebereich der Y-Achse automatisch oder über die Werte "ValueAxisBegin(i)" und "ValueAxisEnd(i)" ermittelt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueAxisAutoRange(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl

BOOLEAN

Optional. TRUE, wenn der Wertebereich der Y-Achse automatisch oder über die Werte "ValueAxisBegin(i)" und "ValueAxisEnd(i)" ermittelt wird.

ValueAxisLabel(i) (RT Professional)

Beschreibung

Legt die Bezeichnung der Wertachse fest. Ob die Angabe ausgewertet wird, ist abhängig von der Eigenschaft "ConfigureTimeAxis(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueAxisLabel(i)[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Bezeichnung der Wertachse festlegt.

ValueAxisScalingType(i) (RT Professional)

Beschreibung

Legt die Art der Skalierung der Achsen fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueAxisScalingType(i)[=AxisScalingType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTrendControl

AxisScalingType

hmiBarScalingLinear (0): Lineare Achsenskalierung

hmiBarScalingLogarithmic (1): Logarithmische Achsenskalierung

hmiBarScalingNegativeLogarithmic (2): Negativ-logarithmische Achsenskalierung

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

ValueColumnAlignment(i) (RT Professional)

Beschreibung

Legt die Ausrichtung der Variablenwerte dieses Spaltenpaars fest. Der Parameter i gibt die Nummer des Spaltenpaars an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueColumnAlignment(i)[=HorizontalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- OnlineTableControl

HorizontalAlignment

hmiAlignmentLeft (0): Der Text wird links ausgerichtet.

hmiAlignmentCentered (1): Der Text wird zentriert ausgerichtet.

hmiAlignmentRight (2): Der Text wird rechts ausgerichtet.

VerticalAlignment (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die vertikale Ausrichtung des Texts im angegebenen Objekt fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.VerticalAlignment[=VerticalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- DateTimeField
- IOField
- OptionGroup
- RoundButton
- Switch
- SymbolicIOField
- TextField

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- CheckBox

VerticalAlignment

Optional. Ein Wert, der die vertikale Ausrichtung festlegt.

hmiAlignmentTop (0): Der Text wird am oberen Rand dargestellt.

hmiAlignmentMiddle (1): Der Text wird in der Mitte dargestellt.

hmiAlignmentBottom (2): Der Text wird am unteren Rand dargestellt.

VerticalGridLines (RT Professional)

Beschreibung

Legt fest, ob vertikale Trennlinien angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**VerticalGridLines**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- OnlineTableControl
- TrendRulerControl
- UserArchiveControl

BOOLEAN

TRUE: Vertikale Trennlinien werden angezeigt.

FALSE: Vertikale Trennlinien werden nicht angezeigt.

VerticalScrollBarPosition (RT Professional)

Beschreibung

Legt die vertikale Verschiebung der Bildlaufleiste in einem Bildfenster mit Rollbalken fest oder gibt den Wert zurück.

Das Bild wird im Bildfenster mit der horizontalen bzw. vertikalen Verschiebung der Bildlaufleisten dargestellt. Wenn Sie das Bild ausgeschnitten darstellen wollen, in der die Bildlaufleisten sich am linken und oberen Rand des Bildes befinden, verwenden Sie für den Ursprung dieses Ausschnitts die Eigenschaften "OffsetLeft" und "OffsetTop".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**VerticalScrollBarPosition**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Screenwindow

Int32

Optional. Ein Wert oder eine Konstante, welche die Verschiebung der vertikalen Bildlaufleiste festlegt.

ViewOnly (RT Advanced)

Beschreibung

Legt fest, ob die Sm@rtClient-Anzeige zur Fernüberwachung oder zur Fernwartung eingesetzt wird.

Fernwartung bedeutet, dass Einstellungen des überwachten Geräts geändert werden können.

Fernüberwachung bedeutet, dass die Einstellungen des überwachten Geräts nicht geändert werden können.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ViewOnly**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- SmartClientView

BOOLEAN

Optional. TRUE, wenn die Sm@rtClient-Anzeige nur zur Fernüberwachung verwendet wird.

Visible (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob das angegebene Objekt sichtbar ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Visible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- AlarmView
- ApplicationWindow
- Bar
- BatteryView
- Button
- ChannelDiagnose
- CheckBox
- Circle
- CircleSegment
- CircularArc
- Clock
- ComboBox
- DateTimeField
- DiscSpaceView
- Ellipse
- EllipseSegment
- EllipticalArc
- FunctionTrendControl
- Gauge
- GraphicIOField
- HTML-Browser

- IOField
- Line
- ListBox
- MediaPlayer
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- PLCCodeViewer
- Polygon
- Polyline
- ProtectedAreaNameView
- RangeLabelView
- RangeQualityView
- RecipeView
- Rectangle
- RoundButton
- Screenwindow
- Slider
- SmartClientView
- StatusForce
- Switch
- SymbolLibrary
- SymbolicIOField
- SysDiagControl
- TextField
- TrendRulerControl
- TrendView
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- UserArchiveControl
- UserView
- WlanQualityView

- WindowsSlider
- ZoneLabelView
- ZoneQualityView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- S7GraphOverview

BOOLEAN

Optional. TRUE, wenn das Objekt sichtbar ist.

Warning (RT Professional)

Beschreibung

Legt fest, ab wieviel Prozent verbrauchtem Speicherplatz eine Warnung ausgegeben wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Warning**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiskSpaceView

Int32

Optional. Ein Wert oder eine Konstante, die den Prozentwert für den verbrauchtem Speicherplatz angibt, ab dem eine Warnung ausgegeben wird.

WarningColor (RT Professional)

Beschreibung

Legt die Farbe fest, in der der Balken der Speicherplatzanzeige dargestellt wird, sobald der Warnbereich erreicht ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- DiskSpaceView

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe festlegt, in der der Balken der Speicherplatzanzeige dargestellt wird, sobald der Warnbereich erreicht ist.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

WarningLowerLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den unteren Grenzwert "WarningLowerLimit" fest.

Die Eigenschaft "WarningLowerLimitEnable" muss den Wert "TRUE" haben, damit der Grenzwert überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningLowerLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den unteren Grenzwert "WarningLowerLimit" festlegt.

Bemerkungen

Folgende Werte werden über die Eigenschaften "WarningLowerLimitColor" und "WarningLowerLimitRelative" festgelegt:

- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

WarningLowerLimitColor (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "WarningLowerLimit" fest.

Die Eigenschaft "WarningLowerLimitEnable" muss den Wert "TRUE" haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningLowerLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den unteren Grenzwert "WarningLowerLimit" festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

Folgende Werte werden über die Eigenschaften "WarningUpperLimit", "WarningUpperLimitColor" und "WarningUpperLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

WarningLowerLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Grenzwert "WarningLowerLimit" überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningLowerLimitEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der Grenzwert "WarningLowerLimit" überwacht wird.

Bemerkungen

Folgende Werte werden über die Eigenschaften "WarningLowerLimit", "WarningLowerLimitColor" und "WarningLowerLimitRelative" festgelegt:

- Grenzwert
- Darstellung beim Erreichen des Grenzwerts
- Art der Auswertung

WarningLowerLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert "WarningLowerLimit" prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningLowerLimitRelative**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional.

TRUE: Der untere Grenzwert "WarningLowerLimit" wird prozentual ausgewertet.

FALSE: Der untere Grenzwert "WarningLowerLimit" wird absolut ausgewertet.

WarningRangeColor (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Farbe des Warnbereichs auf der Skala des Objekts "Gauge" fest.

Die Eigenschaft "WarningRangeVisible" muss den Wert TRUE haben, damit der Gefahrenbereich angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningRangeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Gefahrenbereichs festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

WarningRangeStart (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ab welchem Skalenwert der Warnbereich des Objekts "Gauge" beginnt.

Die Eigenschaft "WarningRangeVisible" muss den Wert TRUE haben, damit der Warnbereich angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningRangeStart**[=SINGLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

SINGLE

Optional. Ein Wert oder eine Konstante, welche den Skalenwert für den Beginn des Warnbereichs enthält.

Bemerkungen

Der Bereich erstreckt sich ab dem Wert "Warning" bis zum Wert "Danger". Wenn kein Bereich für "Danger" aktiviert ist, so geht der Bereich für "Warning" bis zum Ende der Skala.

WarningRangeVisible (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der Warnbereich in der Skala des Objekts "Gauge" angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningRangeVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Gauge

BOOLEAN

Optional. TRUE, wenn der Warnbereich in der Skala angezeigt wird.

Bemerkungen

Die Farbe des Warnbereichs legen Sie mit der Eigenschaft "WarningRangeColor" fest.

Den Beginn des Warnbereichs legen Sie mit der Eigenschaft "WarningRangeStart" fest.

WarningUpperLimit (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die obere Warngrenze fest.

Die Eigenschaft "WarningUpperLimitEnabled" muss den Wert TRUE haben, damit der Grenzwert überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningUpperLimit**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

DOUBLE

Optional. Ein Wert oder eine Konstante, welche den oberen Grenzwert festlegt.

Bemerkungen

"WarningUpperLimitColor" legt die Darstellung beim Erreichen des Grenzwerts fest.

"WarningUpperLimitRelative" legt die Art der Auswertung fest.

WarningUpperLimitColor (RT Professional)

Beschreibung

Legt die Farbe für die untere Warngrenze fest.

Die Eigenschaft "WarningUpperLimitEnabled" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningUpperLimitColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe für den oberen Grenzwert festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

WarningUpperLimitEnabled (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob der obere Grenzwert überwacht wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningUpperLimitEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der obere Grenzwert überwacht wird.

WarningUpperLimitRelative (RT Professional)

Beschreibung

Legt fest, ob der untere Grenzwert "WarningUpperLimit" prozentual oder absolut ausgewertet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WarningUpperLimitRelative**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn der untere Grenzwert "WarningUpperLimit" prozentual ausgewertet wird.

Width (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt die Breite des Objekts in Pixeln fest.

Zugriff in Runtime: Lesen und Schreiben.

Syntax

Object.**Width**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ Typ "ScreenItem" mit folgender Ausprägung:

- AlarmControl
- AlarmView
- ApplicationWindow
- Bar
- BatteryView
- Button
- ChannelDiagnose
- CheckBox
- Circle
- CircleSegment
- CircularArc
- Clock
- ComboBox
- DateTimeField
- DiscSpaceView
- Ellipse
- EllipseSegment
- EllipticalArc
- FunctionTrendControl
- Gauge
- GraphicIOField
- GraphicView
- HTML-Browser
- IOField
- Line
- ListBox
- MediaPlayer
- MultiLineEdit
- OnlineTableControl
- OnlineTrendControl
- OptionGroup
- PLCCodeViewer
- Polygon

- Polyline
- ProtectedAreaNameView
- RangeLabelView
- RangeQualityView
- RecipeView
- Rectangle
- RoundButton
- Screenwindow
- Slider
- SmartClientView
- StatusForce
- Switch
- SymbolLibrary
- SymbolicIOField
- SysDiagControl
- TextField
- TrendRulerControl
- TrendView
- TubeArcObject
- TubeDoubleTeeObject
- TubeTeeObject
- Tubepolyline
- UserArchiveControl
- UserView
- WlanQualityView
- WindowsSlider
- ZoneLabelView
- ZoneQualityView

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- S7GraphOverview

Int32

Optional. Ein Wert oder eine Konstante, welche die Breite des Objekts in Pixeln festlegt.

WindowCloseEnabled (RT Professional)

Beschreibung

Zeigt an, ob ein Fenster in Runtime geschlossen werden kann.

Zugriff in Runtime: Lesen

Syntax

Object.**WindowCloseEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ApplicationWindow
- ScreenWindow

BOOLEAN

Optional. TRUE, wenn das Fenster in Runtime geschlossen werden kann.

WindowMaximizeEnabled (RT Professional)

Beschreibung

Gibt zurück, ob das Objekt in Runtime maximiert werden kann.

Zugriff in Runtime: Lesen

Syntax

Object.**WindowMaximizeEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ApplicationWindow
- Screenwindow

BOOLEAN

TRUE: Das Objekt kann in Runtime maximiert werden.

FALSE: Das Objekt kann in Runtime nicht maximiert werden.

WindowMovingEnabled (RT Professional)

Beschreibung

Gibt zurück, ob das Objekt in Runtime verschoben werden kann.
Zugriff in Runtime: Lesen

Syntax

Object.**WindowMovingEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ApplicationWindow
- Screenwindow

BOOLEAN

TRUE: Das Objekt kann in Runtime verschoben werden.

FALSE: Das Objekt kann in Runtime nicht verschoben werden.

WindowOnTop (RT Professional)

Beschreibung

Gibt zurück, ob das Objekt in Runtime immer im Vordergrund bleibt.
Zugriff in Runtime: Lesen

Syntax

Object.**WindowOnTop**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ApplicationWindow
- Screenwindow

BOOLEAN

TRUE: Das Objekt bleibt in Runtime immer im Vordergrund.

FALSE: Das Objekt bleibt in Runtime nicht immer im Vordergrund.

WindowsContents (RT Professional)

Beschreibung

Gibt den Inhalt des Applikationsfenster wieder. Nur Lese-Zugriff.

Siehe auch

ScriptDiagnostics (Seite 1479)

WindowSizingEnabled (RT Professional)

Beschreibung

Zeigt an, ob in Runtime die Größe des angegebenen Objekts verändert werden kann.

Zugriff in Runtime: Lesen

Syntax

Object.**WindowSizingEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- ApplicationWindow
- ScreenWindow

BOOLEAN

Optional. TRUE, wenn in Runtime die Größe des angegebenen Objekts verändert werden kann.

WindowsStyle (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt fest, ob das Objekt im allgemeinen Windows-Stil dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**WindowsStyle**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- WindowSlider

Bei folgender Ausprägung haben Sie keinen Zugriff in Runtime:

- Switch

BOOLEAN

Optional. TRUE, wenn das Objekt im allgemeinen Windows-Stil dargestellt wird.

3.5.5.15 Eigenschaften X-Z (Panels, Comfort Panels, RT Advanced, RT Professional)

XAxisAdd (RT Professional)

Beschreibung

Legt eine neue X-Achse..

Syntax

Object.**XAxisAdd**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche???

XAxisAlignment (RT Professional)

Beschreibung

Legt fest, wie die ausgewählte Achse ausgerichtet wird.

Syntax

Object.**XAxisAlignment**[=AxisAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

AxisAlignment

Wert	Beschreibung	Erklärung
0	unten	Die ausgewählte X-Achse wird unter der Kurve angezeigt.
1	oben	Die ausgewählte X-Achse wird über der Kurve angezeigt.

XAxisAutoPrecisions (RT Professional)

Beschreibung

Legt fest, ob die Anzahl der Nachkommastellen automatisch festgelegt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisAutoPrecisions[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die Anzahl der Nachkommastellen wird automatisch festgelegt.

FALSE: Der im Inspektorfenster unter "Eigenschaften > X-Achse > Format > Dezimalstellen" projektierte Wert ist wirksam.

XAxisAutoRange(i) (RT Professional)

Beschreibung

Legt fest, ob der Wertebereich der X-Achse automatisch oder über die Attribute "XAxisBegin(i)" und "XAxisEnd(i)" ermittelt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisAutoRange(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

Optional. TRUE, wenn der Wertebereich der X-Achse automatisch oder über die Attribute "XAxisBegin(i)" und "XAxisEnd(i)" ermittelt wird.

XAxisBeginValue (RT Professional)

Beschreibung

Legt das untere Ende des Wertebereichs der ausgewählten X-Achse fest.

Um den Wert zu projektieren, deaktivieren Sie bei "Wertebereich" die Option "Automatisch".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisBeginValue**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die untere Grenze des Wertebereichs der ausgewählten X-Achse festlegt.

XAxisColor (RT Professional)

Beschreibung

Mit dem Attribut legen Sie die verwendete Farbe der gemeinsamen X-Achse fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Color

Ein Wert oder eine Konstante, welche die verwendete Farbe der gemeinsamen X-Achse festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

XAxisCount (RT Professional)

Beschreibung

Legt die Anzahl der X-Achsen fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisCount[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der X-Achsen festlegt.

XAxisEndValue (RT Professional)

Beschreibung

Legt das obere Ende des Wertebereichs der ausgewählten X-Achse fest.

Um den Wert zu projektieren, deaktivieren Sie bei "Wertebereich" die Option "Automatisch".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisEndValue[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die obere Grenze des Wertebereichs der ausgewählten X-Achse festlegt.

XAxisExponentialFormat (RT Professional)

Beschreibung

Legt fest, ob die Werte der gewählten Achse in Exponentialdarstellung angezeigt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisExponentialFormat[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die Werte werden in Exponentialdarstellung angezeigt.

FALSE: Die Werte werden nicht in Exponentialdarstellung angezeigt.

XAxisIndex (RT Professional)

Beschreibung

Referenziert eine projektierte X-Achse. Unter Verwendung der Eigenschaft können Sie einer bestimmten X-Achse die Werte anderer Eigenschaften zuweisen.

Gültige Werte für "XAxisIndex" liegen zwischen 0 und "XAxisCount" minus 1. Die Eigenschaft "ToolBarButtonCount" gibt die Anzahl der projektierten X-Achsen an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisIndex[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer der ausgewählten X-Achse festlegt.

XAxisInTrendColor (RT Professional)

Beschreibung

Legt fest, ob die ausgewählte Achse in der Kurvenfarbe angezeigt wird. Wenn mehrere Kurven im Kurvenfenster angezeigt werden, wird die Farbe der ersten Kurve verwendet. Die Reihenfolge der Kurven legen Sie in den Eigenschaften "Kurven" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisInTrendColor[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die ausgewählte Achse wird in der Kurvenfarbe angezeigt. Die Einstellung im Feld "Farbe" ist unwirksam.

FALSE: Die ausgewählte Achse wird in der Farbe angezeigt, die im Feld "Farbe" eingestellt ist.

XAxisLabel(i) (RT Professional)

Beschreibung

Legt abhängig vom Wert von "XAxisMode(i)" die Bezeichnung der X-Achse einer Kurve fest, die über "CurrentCurveIndex" referenziert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisLabel(i)[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Bezeichnung der X-Achse einer Kurve festlegt, die über "CurrentCurveIndex" referenziert ist.

XAxisName (RT Professional)

Beschreibung

Legt den Namen der ausgewählten Achse fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisName**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, die den Namen der ausgewählten Achse festlegt.

XAxisPrecisions (RT Professional)

Beschreibung

Legt fest, mit wie vielen Nachkommastellen die Werte der ausgewählten X-Achse angezeigt werden.

Um den Wert einzugeben, deaktivieren Sie bei "Format" die Option "Automatisch".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisPrecisions**[=Int16]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int16

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen festlegt.

XAxisRemove (RT Professional)

Beschreibung

Entfernt die ausgewählte Achse aus der Liste.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche die Achse festlegt, die aus der Liste entfernt werden soll.

XAxisRepos (RT Professional)

Beschreibung

Legt die Reihenfolge der Achsen fest. Die Reihenfolge in der Liste bestimmt in Runtime die Position der Achse in der Kurvenanzeige. Wenn die Ausrichtung gleich ist und die Achse in der Liste weiter oben steht, wird sie an einer kurvenferneren Position dargestellt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisRepos**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Reihenfolge der Achsen festlegt???

XAxisScalingType(i) (RT Professional)

Beschreibung

Legt die Art der Skalierung der X-Achse fest. Ob die Angabe ausgewertet wird, ist abhängig vom Wert des Attributs "XAxisMode(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisScalingType**(i)[=AxisScalingType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

AxisScalingType

hmiBarScalingLinear (0): Lineare Achsenskalierung.

hmiBarScalingLogarithmic (1): Logarithmische Achsenskalierung

hmiBarScalingNegativeLogarithmic (2): Negativ-logarithmische Achsenskalierung

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

XAxisTrendWindow (RT Professional)

Beschreibung

Legt fest, in welchem Kurvenfenster die ausgewählte Achse verwendet wird. Die zur Verfügung stehenden Kurvenfenster legen Sie in den Eigenschaften bei "Kurven" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisTrendWindow**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, die den Namen des Kurvenfensters angibt.

XAxisVisible (RT Professional)

Beschreibung

In der Liste werden die Achsen aufgelistet, die Sie angelegt haben.

Aktivieren Sie in der Liste die Achsen, die in den Kurvenanzeigen dargestellt werden.

Klicken Sie auf eine Achse in der Liste, um die Eigenschaften anzupassen und um die Achse einer Kurvenanzeige zuzuordnen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisVisible[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

Optional. Legt fest, ob die X-Achse sichtbar angezeigt wird.

YAxisAdd (RT Professional)

Beschreibung

Legt eine neue Y-Achse..

Syntax

Object.YAxisAdd[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, die ??? festlegt.

YAxisAlignment (RT Professional)

Beschreibung

Legt fest, wie die ausgewählte Achse ausgerichtet wird.

Syntax

Object.YAxisAlignment[=AxisAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

AxisAlignment

Wert	Beschreibung	Erklärung
0	links	Die ausgewählte Y-Achse wird links von der Kurve angezeigt.
1	rechts	Die ausgewählte Y-Achse wird rechts von der Kurve angezeigt.

YAxisAutoPrecisions (RT Professional)

Beschreibung

Legt fest, ob die Anzahl der Nachkommastellen automatisch festgelegt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisAutoPrecisions[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die Anzahl der Nachkommastellen wird automatisch festgelegt.

FALSE: Der im Inspektorfenster unter "Eigenschaften > Y-Achse > Format > Dezimalstellen" projektierte Wert ist wirksam.

YAxisAutoRange (RT Professional)

Beschreibung

Legt fest, ob der Wertebereich der ausgewählten Y-Achse automatisch ermittelt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisAutoRange[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

Wert	Erklärung
TRUE	Der Wertebereich wird automatisch ermittelt.
FALSE	Der Wertebereich wird bestimmt durch die projizierten Werte in den Feldern "von" und "bis".

YAxisBeginValue (RT Professional)

Beschreibung

Legt das untere Ende des Wertebereichs der ausgewählten Y-Achse fest.

Um den Wert zu projizieren, deaktivieren Sie bei "Wertebereich" die Option "Automatisch".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisBeginValue[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die untere Grenze des Wertebereichs der ausgewählten Y-Achse festlegt.

YAxisColor (RT Professional)

Beschreibung

Legt die Farbe der ausgewählten Achse fest. Über die Schaltfläche öffnen Sie den Dialog "Farbauswahl" zur Auswahl der Farbe. Die Einstellung ist nur wirksam, wenn das Feld "in Kurvenfarbe" nicht aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisColor[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der ausgewählten Achse festlegt.

Bemerkungen

Sie können die Funktion "RGB" verwenden, um die Farbe im RGB-Format (Rot, Grün, Blau) festzulegen. Dazu geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an (Wertebereich von 0 bis 255). Die Farbe "Rot" wird beispielsweise so dargestellt: RGB(255, 0, 0). Des Weiteren können Sie auch die VBS-Farbkonstanten wie vbRed oder vbGreen verwenden.

YAxisCount (RT Professional)

Beschreibung

Legt die Anzahl der projizierten Y-Achsen fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**YAxisCount**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Anzahl der projizierten Y-Achsen festlegt.

YAxisEndValue (RT Professional)

Beschreibung

Legt das obere Ende des Wertebereichs der ausgewählten Y-Achse fest.

Um den Wert zu projizieren, deaktivieren Sie bei "Wertebereich" die Option "Automatisch".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**YAxisEndValue**[=DOUBLE]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

DOUBLE

Optional. Ein Wert oder eine Konstante, welche die obere Grenze des Wertebereichs der ausgewählten Y-Achse festlegt.

YAxisExponentialFormat (RT Professional)

Beschreibung

Legt fest, ob die Werte der gewählten Achse in Exponentialdarstellung angezeigt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisExponentialFormat[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die Werte werden in Exponentialdarstellung angezeigt.

FALSE: Die Werte werden nicht in Exponentialdarstellung angezeigt.

YAxisIndex (RT Professional)

Beschreibung

Referenziert eine projektierte Y-Achse. Unter Verwendung der Eigenschaft können Sie einer bestimmten Y-Achse die Werte anderer Eigenschaften zuweisen.

Gültige Werte für "YAxisIndex" liegen zwischen 0 und "YAxisCount" minus 1. Die Eigenschaft "ToolBarButtonCount" gibt die Anzahl der projektierten Y-Achsen an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisIndex[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. Ein Wert oder eine Konstante, welche die Nummer der ausgewählten Y-Achse festlegt.

YAxisInTrendColor (RT Professional)

Beschreibung

Legt fest, ob die ausgewählte Achse in der Kurvenfarbe angezeigt wird. Wenn mehrere Kurven im Kurvenfenster angezeigt werden, wird die Farbe der ersten Kurve verwendet. Die Reihenfolge der Kurven legen Sie in den Eigenschaften "Kurven" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisInTrendColor[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

TRUE: Die ausgewählte Achse wird in der Kurvenfarbe angezeigt. Die Einstellung im Feld "Farbe" ist unwirksam.

FALSE: Die ausgewählte Achse wird in der Farbe angezeigt, die im Feld "Farbe" eingestellt ist.

YAxisLabel (RT Professional)

Beschreibung

Legt den Text fest, mit dem die ausgewählte Achse beschriftet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisLabel[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Beschriftungstext der ausgewählten Y-Achse festlegt.

YAxisName (RT Professional)

Beschreibung

Legt den Namen der ausgewählten Achse fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisName[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der ausgewählten Achse festlegt.

YAxisPrecisions (RT Professional)

Beschreibung

Legt fest, mit wie vielen Nachkommastellen die Werte der ausgewählten Y-Achse angezeigt werden.

Um den Wert einzugeben, deaktivieren Sie bei "Format" die Option "Automatisch".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisPrecisions[=Int16]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int16

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen festlegt.

YAxisRemove (RT Professional)

Beschreibung

Entfernt die ausgewählte Achse aus der Liste.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**YAxisRemove**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den Namen der ausgewählten Achse festlegt.

YAxisRename (RT Professional)

Beschreibung

Ändert den Namen der Y-Achse, die über die Eigenschaft "YAxisIndex" referenziert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**YAxisRename**[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, welche den neuen Namen der ausgewählten Y-Achse festlegt.

YAxisRepos (RT Professional)

Beschreibung

Ändert die Reihenfolge der Achsen. "Auf" und "Ab" bewegen die ausgewählte Achse in der Liste nach oben oder unten.

Die Reihenfolge in der Liste bestimmt in Runtime die Position der Achse in der Kurvenanzeige. Wenn die Ausrichtung gleich ist und die Achse in der Liste weiter oben steht, wird sie an einer kurvenferneren Position dargestellt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisRepos[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

Int32

Optional. #####.

YAxisScalingType (RT Professional)

Beschreibung

Legt fest, wie die Y-Achse skaliert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisScalingType[=AxisScalingType]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

AxisScalingType

Folgende Einstellungen stehen zur Verfügung:

Wert	Beschreibung
0	Linear
1	Logarithmisch
2	Negativ-logarithmisch

YAxisTrendWindow (RT Professional)

Beschreibung

Legt fest, in welchem Kurvenfenster die ausgewählte Achse verwendet wird. Die zur Verfügung stehenden Kurvenfenster legen Sie in den Eigenschaften bei "Kurven" fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YAxisTrendWindow[=STRING]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

STRING

Optional. Ein Wert oder eine Konstante, die den Namen des Kurvenfensters angibt.

YAxisVisible (RT Professional)

Beschreibung

In der Liste werden die Achsen aufgelistet, die Sie angelegt haben.

Aktivieren Sie in der Liste die Achsen, die in den Kurvenanzeigen dargestellt werden.

Klicken Sie auf eine Achse in der Liste, um die Eigenschaften anzupassen und um die Achse einer Kurvenanzeige zuzuordnen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**YAxisVisible**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- FunctionTrendControl

BOOLEAN

Optional. Legt fest, ob die ausgewählte Y-Achse sichtbar angezeigt wird.

ZeroPoint (RT Professional)

Beschreibung

Legt die Lage des Nullpunkts prozentual zur Balkenhöhe fest. Der Nullpunkt kann auch außerhalb des dargestellten Bereichs liegen.

Die Eigenschaft "ScalingType" muss auf "Auto" gesetzt sein.

Die Eigenschaft "ShowScale" muss auf "TRUE" gesetzt sein.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ZeroPoint**[=Int32]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

Int32

Optional. Ein Wert oder eine Konstante, welche die Lage des Nullpunkts prozentual zur Balkenhöhe festlegt.

Zoom/Zoomfactor (RT Professional)

Beschreibung

Setzt den Zoomfaktor eines Bildes bzw. Bildfensters oder liest ihn aus.

Wenn der angegebene Zoomfaktor kleiner ist als der Minimalwert, so wird automatisch der Zoomfaktor auf den Minimalwert gesetzt. Wenn der angegebene Zoomfaktor größer ist als der Maximalwert, so wird der Zoomfaktor auf den Maximalwert gesetzt.

Der Minimalwert des Zoomfaktors liegt bei 2 Prozent, der Maximalwert bei 800 Prozent.

Beim Screen-Objekt wird der Zoomfaktor als numerischer Wert und beim Bildfenster-Objekt in Prozent angegeben.

Zoom: Zoomfaktor des Bildes

Zoomfactor: Zoomfaktor des Bildfensters

Beispiel

Das folgende Beispiel verdoppelt den Zoomfaktor des aktuellen Bildes:

```
'VBS97  
HMIRuntime.ActiveScreen.Zoom = HMIRuntime.ActiveScreen.Zoom * 2
```

Siehe auch

ScreenWindow (Seite 1477)

1 (Panels, Comfort Panels, RT Advanced, RT Professional)

AlarmLow (RT Professional)

Beschreibung

Legt den unteren Grenzwert fest, bei dem Alarm ausgelöst wird oder gibt ihn zurück.

Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft "TypeAlarmLow" fest.

Zugriff in Runtime: Schreiben und Lesen

Syntax

Object.AlarmLow[=REAL]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

REAL

Wert für den unteren Grenzwert

Siehe auch

Bar (Seite 1353)

AlignmentTop (RT Professional)

Beschreibung

Legt die vertikale Ausrichtung des Texts fest oder gibt sie zurück. Wertebereich von 0 bis 2.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlignmentTop**[=VerticalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "TextField", "IOField", "SymbolicIOField", "Button", "RoundButton", "CheckBox", "OptionGroup".

VerticalAlignment

0: Ausrichtung links.

1: Ausrichtung zentriert

2: Ausrichtung rechts

Application (RT Professional)

Beschreibung

Gibt den Inhalt des Applikationsfenster wieder. Nur Lese-Zugriff.

Siehe auch

ScriptDiagnostics (Seite 1479)

AssumeOnExit (RT Professional)

Beschreibung

Legt fest, ob der eingegebene Text nach dem Verlassen des Eingabefeldes übernommen, z. B. mit der Taste <TAB> oder Mausklick.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AssumeOnExit**

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "IOField", "SymbolicIOField".

Siehe auch

IOField (Seite 1418)

SymbolicIOField (Seite 1491)

AssumeOnFull (RT Professional)

Beschreibung

Legt fest, ob das Eingabefeld nach vollständiger Eingabe (die vorgegebene Anzahl an Zeichen wurde eingegeben) automatisch verlassen und die Eingabe dabei übernommen wird..

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.AssumeOnFull[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "IOField", "SymbolicIOField".

BOOLEAN

TRUE: Wird übernommen.

FALSE: Wird nicht übernommen.

Siehe auch

IOField (Seite 1418)

SymbolicIOField (Seite 1491)

BackColorWidth (RT Professional)

Beschreibung

Legt die Breite des 3D-Rahmens in Pixel fest oder gibt sie zurück. Der Wert für die Breite ist abhängig von der Größe des Objektes.

BackColor3-Eigenschaft (RT Professional)

Beschreibung

Legt die Farbe des Balkenhintergrunds fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

BackFlashColorOff (RT Professional)

Beschreibung

Legt die Farbe des Objekthintergrundes für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

BackFlashColorOn (RT Professional)

Beschreibung

Legt die Farbe des Objekthintergrundes für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

BorderFlashColorOff (RT Professional)

Beschreibung

Legt die Linienfarbe des Objektes für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

BorderFlashColorOn (RT Professional)

Beschreibung

Legt die Linienfarbe des Objektes für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG-Schreib-Lese-Zugriff.

BottomConnectedConnectionPointIndex (RT Professional)

Beschreibung

Gibt die Indexnummer des unteren Verbinderpunktes an oder setzt sie.
Long Schreib-Lese-Zugriff

BottomConnectedObjectName (RT Professional)

Beschreibung

Gibt den Objektnamen des Objektes an, das am unteren Verbinderpunkt angedockt ist oder setzt ihn.
Long Schreib-Lese-Zugriff

BoxAlignment (RT Professional)

Beschreibung

TRUE, wenn die Felder rechtsbündig angeordnet sind. BOOLEAN Schreib-Lese-Zugriff.

BoxCount (RT Professional)

Beschreibung

Legt die Anzahl der Felder fest oder gibt sie zurück. Wertebereich von 0 bis 31.

BoxType (RT Professional)

Beschreibung

Legt den Feldtyp fest oder gibt ihn zurück. Wertebereich von 0 bis 2:

- 0: Ausgabe
- 1: Eingabe
- 2: E/A-Feld

Button1Width (RT Professional)

Beschreibung

Legt die Breite des Buttons 1 in Pixel fest oder gibt sie zurück.
Wenn die Eigenschaft SameSize auf TRUE gesetzt ist, erhalten alle Buttons dieselbe Breite.

Button2Width (RT Professional)

Beschreibung

Legt die Breite des Buttons 2 in Pixel fest oder gibt sie zurück.
Wenn die Eigenschaft SameSize auf TRUE gesetzt ist, erhalten alle Buttons dieselbe Breite.

Button3Width (RT Professional)

Beschreibung

Legt die Breite des Buttons 3 in Pixel fest oder gibt sie zurück.
Wenn die Eigenschaft SameSize auf TRUE gesetzt ist, erhalten alle Buttons dieselbe Breite.

Button4Width (RT Professional)

Beschreibung

Legt die Breite des Buttons 4 in Pixel fest oder gibt sie zurück.
Wenn die Eigenschaft SameSize auf TRUE gesetzt ist, erhalten alle Buttons dieselbe Breite.

ButtonColor (RT Professional)

Beschreibung

Legt die Farbe des Schiebers fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

ButtonCommand (RT Professional)

Beschreibung

Legt die Anzeige in der Meldeanzeige fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ButtonCommand**[=Long]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

Long

Optional. Ein Wert oder eine Konstante, welche die Anzeige in der Meldeanzeige festlegt.

0x00000001; 1; Meldeliste ·

0x00000002; 2; Kurzzeitarchivliste·

0x00000004; 4; Langzeitarchivliste·

0x00200000; 2097152; Sperrliste·

0x00000008; 8; Quittierung zentraler Melder·

0x00000010; 16; Einzelquittierung·

0x00000020; 32; Sammelquittierung·

0x00000040; 64; Autoscroll·

0x00000080; 128; Selektions-Dialog·

0x00000100; 256; Sperr-Dialog·

0x00000200; 512; Druck Meldeprotokoll·

0x00000800; 2048; Not-Quittierung·

0x00001000; 4096; Erste Meldung·

0x00002000; 8192; Letzte Meldung·

0x00004000; 16384; Nächste Meldung·

0x00008000; 32768; Vorherige Meldung·

0x00010000; 65536; Infotext-Dialog·

0x00020000; 131072; Kommentar-Dialog·

0x00040000; 262144; Loop in Alarm·

0x00100000; 1048576; Druck aktuelle Ansicht·

0x00400000; 4194304; Meldung sperren/freigeben·

0x00800000; 8388608; Sortier-Dialog·

0x01000000; 16777216; Zeitbasis-Dialog·

0x02000000; 33554432; Hitliste

Siehe auch

AlarmControl (Seite 1338)

CheckAlarmHigh (RT Professional)

Beschreibung

TRUE, wenn der Grenzwert "AlarmHigh" überwacht wird. BOOLEAN Schreib-Lese-Zugriff. Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "AlarmHigh", "ColorAlarmHigh" und "TypeAlarmHigh" festgelegt.

CheckAlarmLow (RT Professional)

Beschreibung

TRUE, wenn der Grenzwert "AlarmLow" überwacht wird. BOOLEAN Schreib-Lese-Zugriff. Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "AlarmLow", "ColorAlarmLow" und "TypeAlarmLow" festgelegt.

CheckLimitHigh4 (RT Professional)

Beschreibung

TRUE, wenn der obere Grenzwert "Reserve 4" überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff. Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "LimitHigh4", "ColorLimitHigh4" und "TypeLimitHigh4" festgelegt.

CheckLimitHigh5 (RT Professional)

Beschreibung

TRUE, wenn der obere Grenzwert "Reserve 5" überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff. Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "LimitHigh5", "ColorLimitHigh5" und "TypeLimitHigh5" festgelegt.

CheckLimitLow4 (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Reserve 4" überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff. Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "LimitLow4", "ColorLimitLow4" und "TypeLimitLow4" festgelegt.

CheckLimitLow5 (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Reserve 5" überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "LimitLow5", "ColorLimitLow5" und "TypeLimitLow5" festgelegt.

CheckToleranceHigh (RT Professional)

Beschreibung

TRUE, wenn der Grenzwert "ToleranceHigh" überwacht wird. BOOLEAN Schreib-Lese-Zugriff.
Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "ToleranceHigh", "ColorToleranceHigh" und "TypeToleranceHigh" festgelegt.

CheckToleranceLow (RT Professional)

Beschreibung

TRUE, wenn der Grenzwert "ToleranceLow" überwacht wird. BOOLEAN Schreib-Lese-Zugriff.
Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "ToleranceLow", "ColorToleranceLow" und "TypeToleranceLow" festgelegt.

CheckWarningHigh (RT Professional)

Beschreibung

TRUE, wenn der Grenzwert "WarningHigh" überwacht wird. BOOLEAN Schreib-Lese-Zugriff.
Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "WarningHigh", "ColorWarningHigh" und "TypeWarningHigh" festgelegt.

CheckWarningLow (RT Professional)

Beschreibung

TRUE, wenn der Grenzwert "WarningLow" überwacht wird. BOOLEAN Schreib-Lese-Zugriff.
Der Grenzwert, die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "WarningLow", "ColorWarningLow" und "TypeWarningLow" festgelegt.

ClearOnNew (RT Professional)

Beschreibung

TRUE, wenn der Feldeintrag gelöscht wird, sobald das E/A-Feld den Fokus erhält. BOOLEAN Schreib-Lese-Zugriff.

ColorAlarmHigh (RT Professional)

Beschreibung

Legt die Balkenfarbe für den Grenzwert "AlarmHigh" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckAlarmHigh" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorAlarmLow (RT Professional)

Beschreibung

Legt die Balkenfarbe für den Grenzwert "AlarmLow" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckAlarmLow" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorLimitHigh4 (RT Professional)

Beschreibung

Legt die Farbe für den oberen Grenzwert "Reserve 4" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckLimitHigh4" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorLimitHigh5 (RT Professional)

Beschreibung

Legt die Farbe für den oberen Grenzwert "Reserve 5" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckLimitHigh5" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorLimitLow4 (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "Reserve 4" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckLimitLow4" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorLimitLow5 (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "Reserve 5" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckLimitLow5" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorToleranceHigh (RT Professional)

Beschreibung

Legt die Farbe für den oberen Grenzwert "ToleranceHigh" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckToleranceHigh" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorToleranceLow (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "ToleranceLow" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckToleranceLow" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorWarningHigh (RT Professional)

Beschreibung

Legt die Farbe für den oberen Grenzwert "WarningHigh" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckWarningHigh" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

ColorWarningLow (RT Professional)

Beschreibung

Legt die Farbe für den unteren Grenzwert "WarningLow" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft "CheckWarningLow" muss den Wert TRUE haben, wenn sich die Balkenfarbe bei Erreichen des Grenzwertes ändern soll.

CountValueColumns (RT Professional)

Beschreibung

Gibt die Anzahl der projizierten Kurven oder Spaltenpaare (sichtbare und unsichtbare) des Fensters zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.CountValueColumns[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der projizierten Kurven oder Spaltenpaare (sichtbare und unsichtbare) des Fensters zurückgibt.

Siehe auch

OnlineTableControl (Seite 1434)

CurrentColumnIndex (RT Professional)

Beschreibung

Legt fest, welche Einstellungen einem bestimmten Spaltenpaar zugeordnet werden können. Die Eigenschaft wird von anderen Eigenschaften ausgewertet. Gültige Werte für den Index haben Werte von 0 bis -1. Die Eigenschaft "CountValueColumns" enthält die Anzahl der anzuzeigenden Spaltenpaare.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.CurrentColumnIndex[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

Int

Optional. Ein Wert oder eine Konstante, welche festlegt, welche Einstellungen einem bestimmten Spaltenpaar zugeordnet werden können.

Bemerkungen

Die Eigenschaft "NumItems" enthält die Anzahl der anzuzeigenden Kurven/Spaltenpaare.

Siehe auch

OnlineTableControl (Seite 1434)

DeleteEnable (RT Professional)

Beschreibung

Legt fest, ob die Rezepturanzeige in Runtime gelöscht werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DeleteEnable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn die Rezepturanzeige in Runtime gelöscht werden kann.

DrawAlwaysInsideFrame (RT Professional)

Beschreibung

Legt fest, ob die Rahmenlinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens oder symmetrisch zum Rahmen gezeichnet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DrawAlwaysInsideFrame**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Rectangle".

BOOLEAN

Option. TRUE, wenn die Umrandungslinie des angegebenen Objekts ab Linienstärke größer als 1 innerhalb des Rahmens gezeichnet wird.

Siehe auch

Rectangle (Seite 1471)

DrawStylishButton (RT Professional)

Beschreibung

Legt fest, ob die Schaltfläche im allgemeinen Windows-Stil dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DrawStylishButton**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Button".

BOOLEAN

Optional. TRUE, wenn die Schaltfläche im allgemeinen Windows-Stil dargestellt wird.

Siehe auch

Button (Seite 1360)

EditAtOnce (RT Professional)

Beschreibung

TRUE, wenn beim Anspringen des Feldes mit der Taste <Tab> die Eingabe sofort und ohne weitere Aktion erfolgen kann. BOOLEAN Schreib-Lese-Zugriff.

Exponent (RT Professional)

Beschreibung

TRUE, wenn die Zahlendarstellung mit Exponenten (z.B. "1,00e+000") erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

FillColor (RT Professional)

Beschreibung

Legt die Farbe des Füllmusters für das Objekt fest oder gibt sie zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.FillColor[=LONG]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "Line", "Polyline", "Ellipse", "Circle", "EllipseSegment", "CircleSegment", "EllipticalArc", "CircularArc", "Rectangle", "Polygon", "TextField", "IOField", "SymbolicIOField", "DateTimeField", "Button", "Roundbutton", "Switch", "Bar", "StatusForce", "Clock", "Gauge", "Slider", "Checkbox".

Ermittlung des Farbwertes

Die Farbe wird im RGB-Format (Rot, Grün, Blau) dargestellt. Geben Sie für jeden der drei RGB-Werte den entsprechenden Dezimalwert an.

Beispiel:

RGB(200, 150, 100)

Beispiel

Das folgende Beispiel setzt die Füllfarbe des Bildes "ScreenWindow1" auf Blau:

```
'VBS73
Dim objScreen
Set objScreen = HMIRuntime.Screens("ScreenWindow1")
objScreen.FillStyle = 131075
objScreen.FillColor = RGB(0, 0, 255)
```

Filling (RT Professional)

Beschreibung

TRUE, wenn das Objekt mit geschlossener Rahmenlinie gefüllt werden kann (also z.B. den Füllstand eines Tanks darstellt). BOOLEAN Schreib-Lese-Zugriff.
Den Füllstand des Objektes setzen Sie mit der Eigenschaft "FillingIndex".

FillingIndex (RT Professional)

Beschreibung

Legt den %-Wert (bezogen auf die Höhe des Objekts) fest, zu dem das Objekt mit geschlossener Rahmenlinie gefüllt wird.
Der Füllstand wird mit der aktuellen Hintergrundfarbe dargestellt. Der nicht gefüllte Hintergrund ist transparent.

FillStyle2 (RT Professional)

Beschreibung

Legt das Füllmuster des Balkens fest oder gibt es zurück.

FlashBackColor (RT Professional)

Beschreibung

TRUE, wenn das Blinken des Hintergrunds des Objektes aktiviert ist. BOOLEAN Schreib-Lese-Zugriff

FlashBorderColor (RT Professional)

Beschreibung

TRUE, wenn das Blinken der Linie des Objektes aktiviert ist. BOOLEAN Schreib-Lese-Zugriff.

FlashRate (RT Professional)

Beschreibung

Legt die Blinkfrequenz fest oder gibt sie zurück. Wertebereich von 0 bis 2.
0 = langsam
1 = mittel

2 = schnell

FlashRateBackColor (RT Professional)

Beschreibung

Legt die Blinkfrequenz für den Objekthintergrund fest oder gibt sie zurück. Wertebereich von 0 bis 2.

0 = langsam

1 = mittel

2 = schnell

FormatType (RT Professional)

Beschreibung

Legt den Datentyp des angegebenen Objekts fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FormatType**[=IOFieldDataFormat]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "IOField".

IOFieldDataFormat

hmiOutputFormatString (2): String

hmiOutputFormatDecimal (1): Dezimal

hmiOutputFormatHexadecimal (3): Hexadezimal

hmiOutputFormatBinary (0): Binär

hmiOutputFormatDate (4): Datum

hmiOutputFormatTime (5): Zeit

hmiOutputFormatDateTime (6): Datum und Zeit

Siehe auch

IOField (Seite 1418)

GridBackColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe fest für die Rezepturanzeige in Runtime.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**GridBackColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Hintergrundfarbe für die Rezepturanzeige in Runtime festlegt.

GridForeColor (RT Professional)

Beschreibung

Legt die Schriftfarbe für die Rezepturanzeige fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**GridForeColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Schriftfarbe für die Rezepturanzeige festlegt.

Hysteresis-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn die Anzeige mit Hysterese erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

HysteresisRange (RT Professional)

Beschreibung

Legt die Hysterese in % des Anzeigewertes fest oder gibt sie zurück.
Die Eigenschaft Hysteresis muss den Wert TRUE haben, damit die Hysterese berechnet werden kann.

InsertEnable (RT Professional)

Beschreibung

Legt fest, ob in Runtime in die Rezepturanzeige eingegeben werden kann.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**InsertEnable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn in die Rezepturanzeige in Runtime eingegeben werden kann.

ItemBorderBackColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe der Trennlinien in der Auswahlliste des Objektes Textliste fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff. Die Hintergrundfarbe ist nur sichtbar, wenn die Eigenschaft ItemBorderStyle > 0 gesetzt ist.

ItemBorderColor (RT Professional)

Beschreibung

Legt die Farbe der Trennlinien in der Auswahlliste des Objektes Textliste fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

ItemBorderWidth (RT Professional)

Beschreibung

Legt die Trennlinienstärke der Auswahlliste des Objektes Textliste in Pixel fest oder gibt sie zurück.

LeftComma-Eigenschaft (RT Professional)

Beschreibung

Legt die Anzahl der Vorkommastellen (0 bis 20) fest oder gibt sie zurück.

LimitHigh4 (RT Professional)

Beschreibung

Legt den oberen Grenzwert für "Reserve 4" fest oder gibt ihn zurück.
Die Eigenschaft CheckLimitHigh4 muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve 4" überwacht werden kann.
Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft TypeLimitHigh4 fest.

LimitHigh5 (RT Professional)

Beschreibung

Legt den oberen Grenzwert für "Reserve 5" fest oder gibt ihn zurück.
Die Eigenschaft CheckLimitHigh5 muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve 5" überwacht werden kann.
Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft TypeLimitHigh5 fest.

LimitLow4 (RT Professional)

Beschreibung

Legt den unteren Grenzwert für "Reserve 4" fest oder gibt ihn zurück.
Die Eigenschaft CheckLimitLow4 muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve 4" überwacht werden kann.
Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft TypeLimitLow4 fest.

LimitLow5 (RT Professional)

Beschreibung

Legt den unteren Grenzwert für "Reserve 5" fest oder gibt ihn zurück.
Die Eigenschaft CheckLimitLow5 muss auf TRUE gesetzt sein, damit der Grenzwert "Reserve 5" überwacht werden kann.
Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft TypeLimitLow5 fest.

LimitMax (RT Professional)

Beschreibung

Legt den oberen Grenzwert als Absolutwert abhängig vom Datenformat fest oder gibt ihn zurück.
Überschreitet der anzuzeigende Wert den oberen Grenzwert, wird er durch eine Folge von *** als nicht darstellbar gekennzeichnet.

LimitMin (RT Professional)

Beschreibung

Legt den unteren Grenzwert als Absolutwert abhängig vom Datenformat fest oder gibt ihn zurück.
Überschreitet der anzuzeigende Wert den oberen Grenzwert, wird er durch eine Folge von *** als nicht darstellbar gekennzeichnet.

LockStatus (RT Professional)

Beschreibung

TRUE, wenn eine gesperrte Mess-Stelle angezeigt werden soll. BOOLEAN Schreib-Lese-Zugriff.

LockText (RT Professional)

Beschreibung

Legt die Beschriftung der Schaltflächen für eine gesperrte Mess-Stelle fest.
Die Eigenschaft LockStatus muss den Wert TRUE haben, damit die Beschriftung angezeigt werden kann.

LockTextColor (RT Professional)

Beschreibung

Legt die Farbe der Schaltflächenbeschriftung für eine gesperrte Mess-Stelle fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Die Eigenschaft LockStatus muss den Wert TRUE haben, damit die Hintergrundfarbe angezeigt werden kann.

LongStrokesBold (RT Professional)

Beschreibung

TRUE, wenn bei der Darstellung der Skala die langen Abschnitte fett angezeigt werden sollen.
BOOLEAN Schreib-Lese-Zugriff.

LongStrokesOnly (RT Professional)

Beschreibung

TRUE, wenn bei der Darstellung der Skala nur die langen Abschnitte angezeigt werden sollen.
BOOLEAN Schreib-Lese-Zugriff.

LongStrokesSize (RT Professional)

Beschreibung

Legt fest, ob nur die großen Skalenstriche dargestellt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowLargeTicksOnly**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Bar

BOOLEAN

Optional. TRUE, wenn nur die großen Skalenstriche dargestellt werden.

LongStrokesTextEach (RT Professional)

Beschreibung

Gibt den Wert zurück, der festlegt, welche Abschnitte bei der Darstellung der Skala beschriftet werden sollen (1 = jeder Abschnitt, 2 = jeder zweite Abschnitt, usw.). Nur Lese-Zugriff

LowerLimitColor(i) (RT Professional)

Beschreibung

Legt die Farbe fest, die Variablenwerte der angegebene Kurve kennzeichnet, die unterhalb des Wertes von "LowerLimitValue(i)" liegen. Ob die Angabe ausgewertet wird ist abhängig von der Eigenschaft "LowerLimitEnabled(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LowerLimitColor**(i)[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe festlegt, die Variablenwerte der angegebene Kurve kennzeichnet, die unterhalb des Wertes von "LowerLimitValue(i)" liegen.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

LowerLimitEnabled(i) (RT Professional)

Beschreibung

Legt fest, ob die Angabe von "LowerLimitColor(i)" verwendet wird, um die Variablenwerte zu kennzeichnen, die unterhalb des Wertes von "LowerLimitValue(i)" liegen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LowerLimitEnabled**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn die Angabe von "LowerLimitColor(i)" verwendet wird, um die Variablenwerte zu kennzeichnen, die unterhalb des Wertes von "LowerLimitValue(i)" liegen.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

LowerLimitValue(i) (RT Professional)

Beschreibung

Legt fest, ob die Variablenwerte die den Wert von "LowerLimitValue(i)" unterschreiten, mit der in "LowerLimitColor(i)" festgelegten Farbe gekennzeichnet werden. Ob die Angabe ausgewertet wird, ist abhängig vom Attribut "LowerLimitEnabled(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LowerLimitValue**(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, die festlegt, ob die Variablenwerte die den Wert von "LowerLimitValue(i)" unterschreiten, mit der in "LowerLimitColor(i)" festgelegten Farbe gekennzeichnet werden.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

Marker-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn die Grenzwerte als Skalenwert angezeigt werden sollen. BOOLEAN Schreib-Lese-Zugriff.

MaximizeButton-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn das Objekt in Runtime maximiert werden kann. Nur Lese-Zugriff.

MCGUBackColorOff (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Unquittiert" die Farbe des Hintergrunds für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCGUBackColorOn (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Unquittiert" die Farbe des Hintergrunds für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCGUBackFlash (RT Professional)

Beschreibung

TRUE, wenn der Hintergrund beim unquittierten Gehen einer Meldung blinken soll. BOOLEAN Schreib-Lese-Zugriff.

MCGUTextColorOff (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Unquittiert" die Farbe des Texts für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCGUTextColorOn (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Unquittiert" die Hintergrundfarbe des Texts für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCGUTextFlash (RT Professional)

Beschreibung

TRUE, wenn die Schrift beim unquittierten Gehen einer Meldung blinken soll. BOOLEAN Schreib-Lese-Zugriff.

MCKOBackColorOff (RT Professional)

Beschreibung

Legt für den Zustand "Gekommen" die Farbe des Hintergrunds für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKOBackColorOn (RT Professional)

Beschreibung

Legt für den Zustand "Gekommen" die Farbe des Hintergrunds für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKOBackFlash (RT Professional)

Beschreibung

TRUE, wenn der Hintergrund beim Kommen einer Meldung blinken soll. BOOLEAN Schreib-Lese-Zugriff.

MCKOTextColorOff (RT Professional)

Beschreibung

Legt für den Zustand "Gekommen" die Farbe des Texts für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKOTextColorOn (RT Professional)

Beschreibung

Legt für den Zustand "Gekommen" die Farbe des Hintergrunds des Texts für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKOTextFlash (RT Professional)

Beschreibung

TRUE, wenn die Schrift beim Kommen einer Meldung blinken soll. BOOLEAN Schreib-Lese-Zugriff.

MCKQBackColorOff (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Quittiert" die Farbe des Hintergrunds für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKQBackColorOn (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Quittiert" die Farbe des Hintergrunds für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKQBackFlash (RT Professional)

Beschreibung

TRUE, wenn der Hintergrund beim quittierten Gehen einer Meldung blinken soll. BOOLEAN Schreib-Lese-Zugriff.

MCKQTextColorOff (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Quittiert" die Farbe des Texts für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKQTextColorOn (RT Professional)

Beschreibung

Legt für den Zustand "Gegangen Quittiert" die Farbe des Hintergrunds des Texts für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

MCKQTextFlash (RT Professional)

Beschreibung

TRUE, wenn die Schrift beim quittierten Gehen einer Meldung blinken soll. BOOLEAN Schreib-Lese-Zugriff.

MCText (RT Professional)

Beschreibung

Legt die Beschriftung für die jeweilige Meldeklasse fest oder gibt sie zurück.

MessageClass (RT Professional)

Beschreibung

Legt die jeweilige Grenzwertverletzung (Alarm High, Alarm Low, Warnung High, Warnung Low, ...) fest, für welche die Einstellungen "Anzeigetext", "Gekommen -", "Gekommen Quittiert -" und "Gegangen Unquittiert -" projiziert werden.

NumberLines (RT Professional)

Beschreibung

Legt für das Objekt Textliste fest, wieviele Zeilen die Auswahlliste enthalten soll oder gibt den Wert zurück. Ist die Anzahl der projizierten Texte größer als dieser Wert, so erhält die Auswahlliste eine vertikale Bildlaufleiste.

ObjectName (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt den Namen des angegebenen Objekts zurück.

Zugriff in Runtime: Lesen

Syntax

Object.ObjectName

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem". Diese Eigenschaft ist eine Standardeigenschaft des ScreenItem-Objekts und damit bei allen Ausprägungen verfügbar.

Siehe auch

Line (Seite 1422)
Polyline (Seite 1459)
Ellipse (Seite 1386)
Circle (Seite 1369)
EllipseSegment (Seite 1388)
CircleSegment (Seite 1372)
EllipticalArc (Seite 1391)
CircularArc (Seite 1374)
Rectangle (Seite 1471)
Polygon (Seite 1456)
TextField (Seite 1505)
IOField (Seite 1418)
SymbolicIOField (Seite 1491)
Button (Seite 1360)
Switch (Seite 1488)
GraphicView (Seite 1412)
GraphicIOField (Seite 1409)
Bar (Seite 1353)
Clock (Seite 1377)
Gauge (Seite 1406)
Slider (Seite 1481)
SymbolLibrary (Seite 1496)

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)
AlarmControl (Seite 1338)
HTMLBrowser (Seite 1416)
CheckBox (Seite 1366)
OptionGroup (Seite 1452)
WindowSlider (Seite 1538)
Connector (Seite 1379)
ScreenWindow (Seite 1477)
DiskSpaceView (Seite 1384)
ChannelDiagnose (Seite 1364)
ScriptDiagnostics (Seite 1479)
Group (Seite 1415)

OnTop (RT Professional)

Beschreibung

TRUE, wenn das Objekt in Runtime immer im Vordergrund bleibt. Nur Lese-Zugriff.

OperationMessage-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn bei einer erfolgten Bedienung eine Meldung ausgegeben werden soll. BOOLEAN Schreib-Lese-Zugriff.

Die Bedienung wird an das Meldesystem gesandt und archiviert. Über das Meldesystem kann eine Meldung z. B. in einer Meldezeile ausgegeben werden.

Besonderheit bei E/A-Feld, Textliste und Schieberegler

Der Grund für die Bedienung kann nur eingegeben werden, wenn die Eigenschaft "OperationReport" auf TRUE gesetzt ist.

OperationReport-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn der Grund für eine Bedienung mitprotokolliert werden soll. BOOLEAN Schreib-Lese-Zugriff.
Bei der Bedienung des Objekts in Runtime erscheint dann ein Dialog, in dem der Bediener den Grund der Bedienung als Text eingeben kann. Die Bedienung wird an das Meldesystem gesandt und archiviert.

Orientation (RT Professional)

Beschreibung

Legt die Textrichtung des angegebenen Objekts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Orientation**[=TextOrientation]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "TextField", "IOField", "SymbolicIOField", "Button", "Switch", "Bar", "OnlineTrendControl", "FunctionTrendControl", "Checkbox", "OptionGroup" oder "WindowSlider".

TextOrientation

hmiTextHorizontal (0): Der Text wird horizontal dargestellt.

hmiTextRotated90Degree (-1): Der Text wird vertikal dargestellt und links ausgerichtet.

hmiTextRotated270Degree (1): Der Text wird vertikal dargestellt und rechts ausgerichtet.

Siehe auch

TextField (Seite 1505)

IOField (Seite 1418)

SymbolicIOField (Seite 1491)

Button (Seite 1360)

Switch (Seite 1488)

Bar (Seite 1353)

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

CheckBox (Seite 1366)

OptionGroup (Seite 1452)

WindowSlider (Seite 1538)

OutputFormat (RT Professional)

Beschreibung

Gibt den Wert für die Darstellung des Ausgabewerts zurück. Die Darstellung ist abhängig vom Datenformat. Nur Lese-Zugriff.

OutputValue (RT Professional)

Beschreibung

Legt die Voreinstellung für den anzuzeigenden Wert fest oder gibt sie zurück. In Runtime wird dieser Wert verwendet, wenn beim Start des Bildes die zugehörige Variable nicht angeschlossen oder nicht aktualisiert ist.

PersistentRTAuthorization (RT Professional)

Beschreibung

Legt die Bedienberechtigung fest, die notwendig ist, um die Persistenz-Einstellung im Runtime verändern zu können. Der einzugebende Wert entspricht der Nummer, die die gewünschte Berechtigung im Benutzerverwaltung besitzt. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "AllowPersistence". Dies gilt nicht für die Meldeanzeige.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PersistentRTAuthorization**[=RtAuthorization]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "AlarmControl", "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

RtAuthorization

Optional. Ein Wert oder eine Konstante, welche die Bedienberechtigung fest, um die Persistenz-Einstellung verändern zu können.

Siehe auch

AlarmControl (Seite 1338)

PersistentRuntime (RT Professional)

Beschreibung

Legt fest, ob veränderte Einstellungen des Fensters auch nach einem Bildwechsel erhalten bleiben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PersistentRuntime**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn veränderte Einstellungen des Fensters auch nach einem Bildwechsel erhalten bleiben.

Bemerkungen

Die Eigenschaft "AllowPersistence" legt fest, ob die Angabe ausgewertet wird.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

PersistentRuntimeAuthorization (RT Professional)

Beschreibung

Legt die Zugriffsberechtigung fest, um die Persistenz-Einstellung im Runtime zu verändern. Der einzugebende Wert entspricht der Nummer, welche die gewünschte Berechtigung in der Benutzerverwaltung besitzt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PersistentRuntimeAuthorization**[=HmiRTAuthorization]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

HmiRTAuthorization

Optional. Ein Wert oder eine Konstante, die die Zugriffsberechtigung festlegt, um die Persistenz-Einstellung im Runtime zu verändern.

Bemerkungen

Die Eigenschaft "AllowPersistence" legt fest, ob die Angabe ausgewertet wird. Dies gilt nicht für die Meldeanzeige.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

PicDeactReferenced (RT Professional)

Beschreibung

TRUE, wenn das zugeordnete Bild für den Zustand "Deaktiviert" im Objekt RoundButton gespeichert wird. Sonst wird nur der zugehörige Objektverweis gespeichert. Nur Lese-Zugriff.

PicDeactTransparent (RT Professional)

Beschreibung

Legt fest, welche Farbe des zugeordneten Bitmap-Objekts (.bmp, .dib) für den Zustand "Deaktiviert" auf "transparent" gesetzt werden soll oder gibt die Farbe zurück. LONG Schreib-Lese-Zugriff.

Die Farbe wird nur dann auf "transparent" gesetzt, wenn die Eigenschaft "PicDeactUseTransColor" den Wert TRUE hat.

PicDeactUseTransColor (RT Professional)

Beschreibung

TRUE, wenn die mit der Eigenschaft "PicDeactTransparent" festgelegte Transparentfarbe für den Zustand "Deaktiviert" verwendet werden soll. BOOLEAN Schreib-Lese-Zugriff.

PicDownTransparent (RT Professional)

Beschreibung

Legt fest, welche Farbe des zugeordneten Bitmap-Objekts (.bmp, .dib) für den Zustand "Ein" auf "transparent" gesetzt werden soll oder gibt die Farbe zurück. LONG Schreib-Lese-Zugriff. Die Farbe wird nur dann auf "transparent" gesetzt, wenn die Eigenschaft "PicDownUseTransColor" den Wert TRUE hat.

PicDownUseTransColor (RT Professional)

Beschreibung

TRUE, wenn die mit der Eigenschaft "PicDownTransparent" festgelegte Transparentfarbe für den Zustand "Ein" verwendet werden soll. BOOLEAN Schreib-Lese-Zugriff.

PicTransColor (RT Professional)

Beschreibung

Legt fest, welche Farbe des zugeordneten Bitmap-Objekts (.bmp, .dib) auf "transparent" gesetzt werden soll oder gibt die Farbe zurück. LONG Schreib-Lese-Zugriff. Die Farbe wird nur dann auf "transparent" gesetzt, wenn die Eigenschaft "PicUseTransColor" den Wert TRUE hat.

PictureName (RT Professional)

Beschreibung

Legt Pfad und Dateinamen des Hintergrundbildes im Prozessbild fest oder gibt diese zurück. LONG Schreib-Lese-Zugriff.

PictureUp (RT Professional)

Beschreibung

Legt das Bild fest, das im Zustand "Aus" angezeigt wird oder gibt den Bildnamen zurück. Das Bild (*.BMP oder *.DIB) muss sich im Ordner "GraCS" des aktuellen Projekts befinden, damit es eingebunden werden kann.

PicUpTransparent (RT Professional)

Beschreibung

Legt fest, welche Farbe des zugeordneten Bitmap-Objekts (.bmp, .dib) für den Zustand "Aus" auf "transparent" gesetzt werden soll oder gibt die Farbe zurück. LONG Schreib-Lese-Zugriff. Die Farbe wird nur dann auf "transparent" gesetzt, wenn die Eigenschaft "PicUpUseTransColor" den Wert TRUE hat.

PicUpUseTransColor (RT Professional)

Beschreibung

TRUE, wenn die mit der Eigenschaft "PicUpTransparent" festgelegte Transparentfarbe für den Zustand "Aus" verwendet werden soll. BOOLEAN Schreib-Lese-Zugriff.

PicUseTransColor (RT Professional)

Beschreibung

TRUE, wenn die mit der Eigenschaft "PicDeactTransparent" festgelegte Transparentfarbe für den Zustand "Deaktiviert" verwendet werden soll. BOOLEAN Schreib-Lese-Zugriff.

Pinned (RT Professional)

Beschreibung

Legt fest, ob die Darstellung des Bildfensters in Runtime von dem Prozessbild abhängt, in dem das Bildfenster projiziert wurde.

PointCount (RT Professional)

Beschreibung

Legt die Anzahl der Eckpunkte fest oder gibt sie zurück. Jeder Eckpunkt hat Positionskordinaten und wird über einen Index identifiziert.

PrintJobName (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt den Druckauftrag fest, den die Druckfunktion der Schaltfläche "Drucken" in der Symbolleiste auslöst. Der empfohlene Druckauftrag für das Control ist voreingestellt.

Über die Auswahl Schaltfläche öffnen Sie den Dialog "Druckauftrag auswählen" zur Auswahl eines Druckauftrags.

Syntax

Object.**PrintJobName**]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "AlarmControl", "FunctionTrendControl", "OnlineTableControl", "OnlineTrendControl", "TrendRulerControl", "UserArchiveControl".

Siehe auch

AlarmControl (Seite 1338)

PrintVisibleColumnsOnly (RT Professional)

Beschreibung

Legt fest, ob nur die aktuell sichtbaren Spalten beim Schnelldruck ausgegeben werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PrintVisibleColumnsOnly**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn nur die aktuell sichtbaren Spalten beim Schnelldruck ausgegeben werden.

Process (RT Professional)

Beschreibung

Legt die Voreinstellung für den anzuzeigenden Wert fest oder gibt sie zurück.

In Runtime wird dieser Wert verwendet, wenn beim Start des Bildes die zugehörige Variable nicht angeschlossen oder nicht aktualisiert ist.

ReadOnly (RT Professional)

Beschreibung

Legt fest, ob das Control Lese- und Schreibzugriff in Runtime hat.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ReadOnly**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn das Control Lese- und Schreibzugriff in Runtime hat.

ReferenceRotationLeft (RT Professional)

Beschreibung

Legt die x-Koordinate des Referenzpunktes fest, um den das Objekt in Runtime gedreht werden soll oder gibt sie zurück.
Der Wert der x-Koordinate ist relativ zur Objektbreite. Geben Sie den Wert ausgehend von der linken Kante des objektumfassenden Rechteckes in Prozent an.

ReferenceRotationTop (RT Professional)

Beschreibung

Legt die y-Koordinate des Referenzpunktes fest, um den das Objekt in Runtime gedreht werden soll oder gibt sie zurück.
Der Wert der y-Koordinate ist relativ zur Objekthöhe. Geben Sie den Wert ausgehend von der oberen Kante des objektumfassenden Rechteckes in Prozent an.

Relevant-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn das Objekt für die Bildung der Sammelanzeige berücksichtigt wird. BOOLEAN Schreib-Lese-Zugriff.

RightComma-Eigenschaft (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen (0 bis 20) fest oder gibt sie zurück.

RTPersistencePasswordLevel (RT Professional)

Beschreibung

Zeigt die Berechtigung für die Online-Projektierung. Über die Auswahl Schaltfläche können Sie die Berechtigung ändern. Berechtigungen werden in der Benutzerverwaltung projiziert.

Syntax

Object.RTPersistencePasswordLevel

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "AlarmControl", "OnlineTrendControl", "TrendRulerControl", "UserArchiveControl" oder "OnlineTableControl" (Runtime Professional) bzw. "HMITrendView" (Runtime Advanced).

Siehe auch

AlarmControl (Seite 1338)

UserArchiveControl (Seite 1528)

SameSize (RT Professional)

Beschreibung

TRUE, wenn alle vier Buttons beim Objekt GroupDisplay die gleiche Größe haben. BOOLEAN Schreib-Lese-Zugriff.

ScaleTicks (RT Professional)

Beschreibung

Legt die Anzahl der Segmente fest, in die der Balken durch die großen Teilstriche der Skala unterteilt wird:

0-100: Objekt kann maximal in 100 Segmente unterteilt werden

= 0: Die optimale Anzahl der Segmente wird automatisch festgelegt

ScrollBars (RT Professional)

Beschreibung

TRUE, wenn das Objekt in Runtime Bildlaufleisten hat. Nur Lese-Zugriff.

ScrollPositionX (RT Professional)

Beschreibung

Legt die horizontale Verschiebung der Bildlaufleiste in einem Bildfenster mit Rollbalken fest oder gibt den Wert zurück.

Das Bild wird im Bildfenster mit der horizontalen bzw. vertikalen Verschiebung der Bildlaufleisten dargestellt. Wenn Sie das Bild ausgeschnitten darstellen wollen, in der die Bildlaufleisten sich am linken und oberen Rand des Bildes befinden, verwenden Sie für den Ursprung dieses Ausschnitts die Eigenschaften "OffsetLeft" und "OffsetTop".

ScrollPositionY (RT Professional)

Beschreibung

Legt die vertikale Verschiebung der Bildlaufleiste in einem Bildfenster mit Rollbalken fest oder gibt den Wert zurück.

Das Bild wird im Bildfenster mit der horizontalen bzw. vertikalen Verschiebung der Bildlaufleisten dargestellt. Wenn Sie das Bild ausgeschnitten darstellen wollen, in der die Bildlaufleisten sich am linken und oberen Rand des Bildes befinden, verwenden Sie für den Ursprung dieses Ausschnitts die Eigenschaften "OffsetLeft" und "OffsetTop".

SelBGColor (RT Professional)

Beschreibung

Legt die Hintergrundfarbe für den ausgewählten Eintrag beim Objekt Textliste fest oder gibt ihn zurück. LONG Schreib-Lese-Zugriff.

SelectionMode (RT Professional)

Beschreibung

Legt fest, ob und wie eine Meldezeile selektiert werden kann.

.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object. **SelectionMode**[=Mode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "MessageView".

Mode

0 - NoSelection: Verhindert die Selektion einer Meldung. Eine Quittierung wirkt sich immer auf die älteste anstehende Meldung aus.

1 - Cell: Ermöglicht die Selektion von Feldern in der Meldezeile. Eine Quittierung wirkt sich immer auf die selektierte Meldung aus.

2 - Line: Ermöglicht die Selektion einer Meldezeile. Eine Quittierung wirkt sich immer auf die selektierte Meldung aus.

SelectionMode (RT Professional)

Beschreibung

Legt fest, ob und wie eine Meldezeile ausgewählt werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SelectionMode**[=AlarmViewAdvancedSelectionMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

AlarmViewAdvancedSelectionMode

hmiAlarmViewAdvancedSelectionModeNone (0): Verhindert die Selektion einer Meldung. Eine Quittierung wirkt sich immer auf die älteste anstehende Meldung aus.

hmiAlarmViewAdvancedSelectionModeCell (1): Ermöglicht die Selektion von Feldern in der Meldezeile. Eine Quittierung wirkt sich immer auf die selektierte Meldung aus.

hmiAlarmViewAdvancedSelectionModeLine (2): Ermöglicht die Selektion einer Meldezeile. Eine Quittierung wirkt sich immer auf die selektierte Meldung aus.

Siehe auch

AlarmControl (Seite 1338)

SelTextColor (RT Professional)

Beschreibung

Legt die Textfarbe des ausgewählten Eintrages des Objektes Textliste fest oder gibt sie zurück.
LONG Schreib-Lese-Zugriff.

ShortenCellText (RT Professional)

Beschreibung

Legt fest, ob die Inhalte der Felder einer Meldezeile bei zu kleiner Spaltenbreite abgekürzt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShortenCellText**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Inhalte der Felder einer Meldezeile bei zu kleiner Spaltenbreite abgekürzt werden.

Siehe auch

AlarmControl (Seite 1338)

ShortenColumnHeaderText (RT Professional)

Beschreibung

Legt fest, ob die Inhalte der Felder einer Titelzeile bei zu kleiner Spaltenbreite abgekürzt werden sollen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShortenColumnHeaderText**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Inhalte der Felder einer Titelzeile bei zu kleiner Spaltenbreite abgekürzt werden sollen.

Siehe auch

AlarmControl (Seite 1338)

ShowVerticalGridlines (RT Professional)

Beschreibung

Legt fest, ob die Spalten der Meldeanzeige durch vertikale Trennlinien getrennt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowVerticalGridlines**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Spalten des Meldefensters durch vertikale Trennlinien getrennt werden.

Siehe auch

AlarmControl (Seite 1338)

ShowXValuesExponential(i) (RT Professional)

Beschreibung

Legt fest, ob ein Wert der X-Koordinate in Exponentialschreibweise dargestellt wird. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "XAxisMode(i)".
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowXValuesExponential**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn in Wert der X-Koordinate in Exponentialschreibweise dargestellt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Sie verwenden die Funktion "Anzeige Wert an dieser Stelle", um die X-Koordinate des Messwerts anzeigen zu lassen.

Siehe auch

FunctionTrendControl (Seite 1398)

ShowYValuesExponential(i) (RT Professional)

Beschreibung

Legt fest, ob ein Wert der Y-Koordinate in Exponentialschreibweise dargestellt wird. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "XAxisMode(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowYValuesExponential**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn ein Wert der Y-Koordinate in Exponentialschreibweise dargestellt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Sie verwenden die Funktion "Anzeige Wert an dieser Stelle", um die Y-Koordinate des Messwerts anzeigen zu lassen.

Siehe auch

FunctionTrendControl (Seite 1398)

SignificantMask-Eigenschaft (RT Professional)

Beschreibung

Wird in Runtime zur Darstellung der aktiven Meldeklasse mit der höchsten Priorität benötigt. Der Wert der SignificantMask-Eigenschaft stellt einen systeminternen Ausgabewert dar und erfordert keine spezifische Projektierung durch den Anwender. Die Aktualisierung erfolgt in Runtime durch Anklicken des Objektes.

SortByTimeEnable (RT Advanced)

Beschreibung

Legt fest, ob die Sortierichtung in Runtime geändert werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SortByTimeEnable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Sortierichtung in Runtime geändert werden kann.

Siehe auch

AlarmView (Seite 1350)

SortOnColumnHeaderClick (RT Professional)

Beschreibung

Legt fest, ob die Meldeblöcke über die Spaltenüberschrift der Meldeblöcke sortiert werden können.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SortOnColumnHeaderClick**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Meldeblöcke über die Spaltenüberschrift der Meldeblöcke sortiert werden können.

Siehe auch

AlarmControl (Seite 1338)

SortTimeAscending (Panels, Comfort Panels, RT Advanced)

Beschreibung

Legt fest, ob die zuletzt eingegangene Meldung im Objekt "MessageView" oben angezeigt wird (aufsteigende Sortierreihenfolge).

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.SortTimeAscending [= BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "MessageView".

BOOLEAN

Optional. TRUE, wenn die zuletzt eingegangene Meldung oben angezeigt wird.

SortTimeEnable (Panels, Comfort Panels, RT Advanced)

Beschreibung

Legt fest, ob die Sortierung der Meldungen nach der Uhrzeit im Objekt "MessageView" geändert werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.SortTimeEnable [= BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "MessageView".

BOOLEAN

Optional. TRUE, wenn die Sortierung vom Bediener am Gerät geändert werden kann.

StartPointLeft (RT Professional)

Beschreibung

Legt den horizontalen Abstand vom linken Bildrand in Pixel fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StartPointLeft**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "Line".

Int

Optional. Ein Wert, der den horizontalen Abstand des Startpunkts vom linken Bildrand in Pixel festlegt.

Siehe auch

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)

CircularArc (Seite 1374)

TagName (RT Professional)

Beschreibung

Die Eigenschaft "Index" referenziert eine Kurve. "TagName" legt fest, welche Variable mit dieser Kurve verknüpft ist. Die Angabe erfolgt in der Form "Archivname\Variablenname" zur Darstellung von Variablen eines Prozesswertarchives bzw. "Variablenname" zur Darstellung einer internen oder externen Variablen, die nicht in einem Archiv gespeichert wird.

TimeAxisShowGridLines(i) (RT Professional)

Beschreibung

Legt fest, ob das Kurvenfenster mit zur x-Achse parallelen Rasterlinien dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeAxisShowGridLines**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrend".

BOOLEAN

Optional. TRUE, wenn das Kurvenfenster mit zur x-Achse parallelen Rasterlinien dargestellt wird.

Bemerkungen

Der Abstand zweier Rasterlinien kann über die Eigenschaft "TimeAxisGridLineInterval(i)" verändert werden.

Siehe auch

OnlineTrendControl (Seite 1443)

TimeColumnFormat(i) (RT Professional)

Beschreibung

Legt das Format der Zeitspalten für das angegebene Spaltenpaar fest. Der Parameter i gibt die Nummer des Spaltenpaars an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeColumnFormat**(i)[=TimeFormat]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

TimeFormat

- (0): Die Angaben erfolgen in der Form hh:mm
- (1): Die Angaben erfolgen in der Form hh:mm:ss
- (2): Die Angaben erfolgen in der Form hh:mm:ss.ms
- (3): Die Angaben erfolgen in der Form hh:mm (Volle Stunden)
- (4): Die Angaben erfolgen in der Form hh:mm:ss (Volle Minuten)
- (5): Die Angaben erfolgen in der Form hh:mm:ss.ms (Volle Sekunden)
- (6): Die Angaben erfolgen in der Form dd_mm_yy_hh_mm
- (7): Die Angaben erfolgen in der Form dd_mm_yy_hh_mm (Volle Stunde)
- (8): Die Angaben erfolgen in der Form dd_mm_yy
- (9): Die Angaben erfolgen in der Form dd_mm_yy (Ganzer Tag)

Siehe auch

OnlineTableControl (Seite 1434)

TitleBackColor (RT Professional)

Beschreibung

Gibt die Hintergrundfarbe der Tabellenüberschrift für den gewählten Zustand an. Über die Schaltfläche öffnen Sie den Farbauswahl-Dialog.

Titleline (RT Professional)

Beschreibung

TRUE, wenn das Control eine Titelleiste besitzt und im Runtime verschoben werden kann.
BOOLEAN Schreib-Lese-Zugriff.

ToleranceHigh-Eigenschaft (RT Professional)

Beschreibung

Legt den Grenzwert für "Toleranz high" fest oder gibt ihn zurück.
Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft TypeToleranceHigh fest.
Die Überwachung des Grenzwerts ist nur wirksam, wenn die Eigenschaft CheckToleranceHigh auf "True" gesetzt ist.

ToleranceLow-Eigenschaft (RT Professional)

Beschreibung

Legt den Grenzwert für "Toleranz low" fest oder gibt ihn zurück.
Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft TypeToleranceLow fest.
Die Überwachung des Grenzwerts ist nur wirksam, wenn die Eigenschaft CheckToleranceLow auf "True" gesetzt ist.

TopConnectedConnectionPointIndex (RT Professional)

Beschreibung

Gibt die Indexnummer des oberen Verbinderpunktes an oder setzt sie.
Long Schreib-Lese-Zugriff

TopConnectedObjectName (RT Professional)

Beschreibung

Gibt den Objektnamen des Objektes an, das am oberen Verbinderpunkt angedockt ist oder setzt ihn.

Long Schreib-Lese-Zugriff

Trend (RT Professional)

Beschreibung

TRUE, wenn die Tendenz (steigend oder fallend) des zu überwachenden Messwertes mit einem kleinen Pfeil angezeigt werden soll. BOOLEAN Schreib-Lese-Zugriff.

TrendTag (RT Professional)

Beschreibung

Legt fest, welche Variable mit der angegebenen Kurve verknüpft ist.

Sie geben die Werte entweder in der Form "Archivname\Variablenname" ein, zur Darstellung von Variablen eines Prozesswertarchives, oder in der Form "Variablenname" zur Darstellung einer internen oder externen Variablen, die nicht in einem Archiv gespeichert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.TrendTag[=HmiTag]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrend".

HmiTag

Optional. Ein Wert oder eine Konstante, welche die Variable festlegt, die mit der angegebenen Kurve verknüpft ist.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

TypeAlarmHigh (RT Professional)

Beschreibung

TRUE, wenn der obere Grenzwert, bei dem Alarm ausgelöst wird, prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeAlarmLow (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert, bei dem Alarm ausgelöst wird, prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeLimitHigh4 (RT Professional)

Beschreibung

TRUE, wenn der obere Grenzwert "Reserve 4" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeLimitHigh5 (RT Professional)

Beschreibung

TRUE, wenn der obere Grenzwert "Reserve 5" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeLimitLow4 (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Reserve 4" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeLimitLow5 (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Reserve 5" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeToleranceHigh (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Toleranz high" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeToleranceLow (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Toleranz low" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeWarningHigh (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Warning high" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

TypeWarningLow (RT Professional)

Beschreibung

TRUE, wenn der untere Grenzwert "Warning low" prozentual ausgewertet wird. FALSE, wenn die Auswertung absolut erfolgen soll. BOOLEAN Schreib-Lese-Zugriff.

UncertainStateColor(i) (RT Professional)

Beschreibung

Legt die Farbe fest für die Kennzeichnung von Werten mit unsicherem Status.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UncertainStateColor**(i)[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe fest für die Kennzeichnung von Werten mit unsicherem Status.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Wenn der Anfangswerte eines Wertes nach dem Aktivieren des Runtime nicht bekannt ist oder ein Ersatzwert verwendet wird, dann hat dieser Werte einen unsicheren Status.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

UncertainStateEnabled(i) (RT Professional)

Beschreibung

Legt fest, ob Werte mit unsicherem Status mit der in "ReplacementColor(i)" festgelegten Farbe gekennzeichnet werden.

Werte, deren Anfangswert nach dem Aktivieren des Runtime nicht bekannt ist, oder für die ein Ersatzwert verwendet wird, besitzen einen unsicheren Status.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UncertainStateEnabled**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn Werte mit unsicherem Status mit der in "ReplacementColor(i)" festgelegten Farbe gekennzeichnet werden.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

UpdateEnable (RT Professional)

Beschreibung

Legt fest, ob die Rezepturanzeige in Runtime geändert werden kann.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UpdateEnable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn die Rezepturanzeige in Runtime geändert werden kann.

UpperLimitColor(i) (RT Professional)

Beschreibung

Legt die Farbe fest, die Variablenwerte kennzeichnet, die oberhalb des Wertes von "UpperLimitValue(i)" liegen. Ob die Angabe ausgewertet wird ist abhängig vom Wert der Eigenschaft "UpperLimitEnabled(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UpperLimitColor**(i)[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Variablenwerte kennzeichnet, die oberhalb des Wertes von "UpperLimitValue(i)" liegen.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)

UpperLimitEnabled(i) (RT Professional)

Beschreibung

Legt fest, ob die Angabe von "UpperLimitColor(i)" verwendet wird, um die Variablenwerte zu kennzeichnen, die oberhalb des Wertes von "UpperLimitValue(i)" liegen.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UpperLimitEnabled**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn die Angabe von "UpperLimitColor(i)" verwendet wird, um die Variablenwerte zu kennzeichnen, die oberhalb des Wertes von "UpperLimitValue(i)" liegen.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)

UpperLimitValue(i) (RT Professional)

Beschreibung

Legt fest, ob die Variablenwerte, die den Wert von "UpperLimitValue(i)" überschreiten, mit der in "UpperLimitColor(i)" festgelegten Farbe gekennzeichnet werden. Ob die Angabe ausgewertet wird ist abhängig vom Wert der Eigenschaft "UpperLimit(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.UpperLimitValue(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, die festlegt, ob die Variablenwerte, die den Wert von "UpperLimitValue(i)" überschreiten, mit der in "UpperLimitColor(i)" festgelegten Farbe gekennzeichnet werden.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

UseAllServers (RT Professional)

Beschreibung

Legt fest, ob die im Meldefenster darzustellenden Daten von allen an einem verteilten System beteiligten Servern angefordert werden. Auf den beteiligten Servern muss das Alarm Logging aktiviert sein.

Zugriff in Runtime: Nur Lesen

Syntax

Object.UseAllServers[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmView".

BOOLEAN

Optional. TRUE, wenn die im Meldefenster darzustellenden Daten von allen an einem verteilten System beteiligten Servern angefordert werden.

Siehe auch

AlarmControl (Seite 1338)

UserValue1-Eigenschaft (RT Professional)

Beschreibung

Legt einen beliebigen Wert fest oder gibt ihn zurück.
Den Wert können Sie z.B. von einem Skript auswerten lassen. In Runtime wird dieser Wert weder ausgewertet noch angezeigt.

UserValue2-Eigenschaft (RT Professional)

Beschreibung

Legt einen beliebigen Wert fest oder gibt ihn zurück.
Den Wert können Sie z.B. von einem Skript auswerten lassen. In Runtime wird dieser Wert weder ausgewertet noch angezeigt.

UserValue3-Eigenschaft (RT Professional)

Beschreibung

Legt einen beliebigen Wert fest oder gibt ihn zurück.
Den Wert können Sie z.B. von einem Skript auswerten lassen. In Runtime wird dieser Wert weder ausgewertet noch angezeigt.

UserValue4-Eigenschaft (RT Professional)

Beschreibung

Legt einen beliebigen Wert fest oder gibt ihn zurück.
Den Wert können Sie z.B. von einem Skript auswerten lassen. In Runtime wird dieser Wert weder ausgewertet noch angezeigt.

ValueAxisDecimalPrecision(i) (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen für den Wertebereich der angegebenen Kurve fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueAxisDecimalPrecision(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend" oder "FunctionTrendView".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen für den Wertebereich der angegebenen Kurve festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ValueAxisShowGridLines(i) (RT Professional)

Beschreibung

Legt fest, ob das Kurvenfenster mit zur Y-Achse parallelen Rasterlinien dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ValueAxisShowGridLines(i)**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn das Kurvenfenster mit zur Y-Achse parallelen Rasterlinien dargestellt wird.

Bemerkungen

Der Abstand zweier Rasterlinien kann über die Eigenschaft "ValueAxisGridLineInterval(i)" verändert werden.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ValueCollectionPrecision(i) (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen fest, die in dieser Wertespalte angezeigt werden. Der Parameter *i* gibt die Nummer des Spaltenpaars an. Es können maximal 16 Nachkommastellen angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueColumnPrecision(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

Int

Optional. Ein Wert oder eine Konstante, welche Legt die Anzahl der Nachkommastellen festlegt, die in dieser Wertespalte angezeigt werden.

Siehe auch

OnlineTableControl (Seite 1434)

WarningHigh (RT Professional)

Beschreibung

Legt den oberen Grenzwert "Warning High" fest oder gibt ihn zurück.

Damit der Grenzwert überwacht wird, muss die Eigenschaft "CheckWarningHigh" auf TRUE gesetzt sein.

Die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "ColorWarningHigh" und "TypeWarningHigh" festgelegt.

WarningLow (RT Professional)

Beschreibung

Legt den unteren Grenzwert "Warning Low" fest oder gibt ihn zurück.

Damit der Grenzwert überwacht wird, muss die Eigenschaft "CheckWarningLow" auf TRUE gesetzt sein.

Die Darstellung beim Erreichen des Grenzwerts und die Art der Auswertung werden über die Eigenschaften "ColorWarningLow" und "TypeWarningLow" festgelegt.

StyleSettings (RT Professional)

Beschreibung

Legt fest, in welchem Stil das Objekt dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StyleSettings**[=WinCCStyle]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit folgender Ausprägung:

- Button
- RoundButton
- WindowsSlider

WinCCStyle

Optional. Ein Wert oder eine Konstante, welche den Stil festlegt, in dem das Objekt dargestellt wird.

Benutzerdefiniert	Stellt das Objekt entsprechend den eigenen Einstellungen dar.
Global	Stellt das Objekt im global eingestellten Design dar.
Windows-Stil	Stellt das Objekt im Windows-Stil dar.

WindowBorder (RT Professional)

Beschreibung

TRUE, wenn das Fenster in Runtime mit Rahmen dargestellt wird. Nur Lese-Zugriff.

XAxisBegin(i) (RT Professional)

Beschreibung

Legt die untere Grenze der X-Achse einer mit der Eigenschaft "CurrentCurveIndex" referenzierten Kurve fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "XAxisAutorange(i)" und "ShareXAxis".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisBegin**(i)[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

String

Optional. Ein Wert oder eine Konstante, welche die untere Grenze der X-Achse einer mit der Eigenschaft "CurrentCurveIndex" referenzierten Kurve festlegt.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisEnd(i) (RT Professional)

Beschreibung

Legt die obere Grenze der X-Achse einer mit "CurrentCurveIndex" referenzierten Kurve fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "XAxisAutorange(i)" und "ShareXAxis".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisEnd(i)[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

String

Optional. Ein Wert oder eine Konstante, welche die obere Grenze der X-Achse einer mit "CurrentCurveIndex" referenzierten Kurve festlegt.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisShowGridLines(i) (RT Professional)

Beschreibung

Legt fest, ob das Kurvenfenster mit zur X-Achse parallelen Rasterlinien dargestellt wird. Der Abstand zweier Rasterlinien kann über die Eigenschaft "XAxisGridLineInterval(i)" verändert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisShowGridLines**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

BOOLEAN

Optional. TRUE, wenn das Kurvenfenster mit zur X-Achse parallelen Rasterlinien dargestellt wird.

Siehe auch

FunctionTrendControl (Seite 1398)

2 (Panels, Comfort Panels, RT Advanced, RT Professional)

Activate (RT Professional)

Beschreibung

Legt fest, ob die darzustellenden Daten vom Meldeserver angefordert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Activate**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die darzustellenden Daten vom Meldeserver angefordert werden.

Siehe auch

AlarmControl (Seite 1338)

AdjustBorder3DWithStyle (RT Professional)

Beschreibung

Legt fest, dass die 3D-Rahmenbreite im allgemeinen Windows-Stil dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AdjustBorder3DWithStyle**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Button".

BOOLEAN

Optional. TRUE, wenn die 3D-Rahmenbreite im allgemeinen Windows-Stil dargestellt wird.

Siehe auch

Button (Seite 1360)

AlarmFilter (RT Professional)

Beschreibung

Legt ein SQL-Statement zur Auswahl der im Meldefenster angezeigten Meldungen fest.

Alarmlist and hitlist filtering and sorting settings

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlarmFilter**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

String

Optional. Ein Wert oder eine Konstante, welche ein SQL-Statement zur Auswahl der im Meldefenster angezeigten Meldungen festlegt.

Siehe auch

AlarmControl (Seite 1338)

AngleAlpha (RT Professional)

Beschreibung

Legt den Tiefenwinkel α für den 3D-Effekt des "3DBarGraph"-Objektes fest oder gibt ihn zurück. Wertebereich in Grad von 0 bis 90.

AngleBeta (RT Professional)

Beschreibung

Legt den Tiefenwinkel β für den 3D-Effekt des "3DBarGraph"-Objektes fest oder gibt ihn zurück. Wertebereich in Grad von 0 bis 90.

Axe-Eigenschaft (RT Professional)

Beschreibung

Definiert die Lage des 3D-Balkens im Koordinatensystem oder gibt den Wert zurück. Wertebereich von 0 bis 2.

0: Der 3D-Balken wird auf der X-Achse dargestellt.

1: Der 3D-Balken wird auf der Y-Achse dargestellt.

2: Der 3D-Balken wird auf der Z-Achse dargestellt.

AxisSection (RT Professional)

Beschreibung

Legt den Abstand zweier langer Achsabschnitte fest oder gibt ihn zurück. Die Angabe des Abstandes erfolgt dabei in Skaleneinheiten und ist abhängig von den projizierten Minimal- und Maximalwerten.

Background (RT Professional)

Beschreibung

TRUE, wenn der Hintergrund beim 3DBarGraph-Objekt sichtbar sein soll. BOOLEAN Schreib-Lese-Zugriff.

BarDepth (RT Professional)

Beschreibung

Legt die Balkentiefe in Pixel fest oder gibt sie zurück.

BarHeight-Eigenschaft (RT Professional)

Beschreibung

Legt die Balkenhöhe in Pixel fest oder gibt sie zurück.

BaseX-Eigenschaft (RT Professional)

Beschreibung

Legt den horizontalen Abstand in Pixel des rechten Balkenrands zum linken Rand des Objektfeldes fest oder gibt ihn zurück.

BaseY-Eigenschaft (RT Professional)

Beschreibung

Legt den vertikalen Abstand des unteren Balkenrands zum oberen Rand des Objektfeldes fest oder gibt ihn zurück.

CloseButton (RT Professional)

Beschreibung

TRUE, wenn das Fenster eine "Schließen"-Schaltfläche besitzt. Nur Lese-Zugriff.

ColumnSizingEnable (RT Professional)

Beschreibung

Legt fest, ob die Breite der Spalten des Meldefensters veränderbar ist. Die Breite der Spalten kann nur verändert werden, wenn die Eigenschaft "AutoScroll" nicht aktiviert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColumnSizingEnable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Breite der Spalten des Meldefensters veränderbar ist.

Siehe auch

AlarmControl (Seite 1338)

ForeFlashColorOff (RT Professional)

Beschreibung

Legt die Farbe des Texts für den Blinkzustand "Aus" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

ForeFlashColorOn (RT Professional)

Beschreibung

Legt die Farbe des Texts für den Blinkzustand "Ein" fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

InsertData(i) (RT Professional)

Beschreibung

Legt fest, welche Daten für die aktive Kurve eingefügt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**InsertData**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn "DataIndex(i)" ignoriert wird und die Daten im Datenpuffer hinten angefügt werden.

FALSE, wenn die Daten an die Position "DataIndex(i)" des Datenbuffers eingefügt werden.

Bemerkungen

Bei jeder Operation mit "InsertData(i)" wird die Kurvenanzeige neu gezeichnet.

Siehe auch

FunctionTrendControl (Seite 1398)

Layer00Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 0 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer00Value und Layer00Color festgelegt.

Layer00Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 0 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer00Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer00Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 0" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer00Checked den Wert TRUE hat.

Layer01Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 1 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer01Value und Layer01Color festgelegt.

Layer01Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 1 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer01Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer01Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 1" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer01Checked den Wert TRUE hat.

Layer02Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 2 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer02Value und Layer02Color festgelegt.

Layer02Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 2 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer02Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer02Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 2" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer02Checked den Wert TRUE hat.

Layer03Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 3 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer03Value und Layer03Color festgelegt.

Layer03Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 3 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer03Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer03Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 3" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer03Checked den Wert TRUE hat.

Layer04Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 4 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer04Value und Layer04Color festgelegt.

Layer04Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 4 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer04Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer04Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 4" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer04Checked den Wert TRUE hat.

Layer05Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 5 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer05Value und Layer05Color festgelegt.

Layer05Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 5 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer05Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer05Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 5" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer05Checked den Wert TRUE hat.

Layer06Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 6 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer06Value und Layer06Color festgelegt.

Layer06Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 6 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer06Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer06Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 6" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer06Checked den Wert TRUE hat.

Layer07Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 7 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer07Value und Layer07Color festgelegt.

Layer07Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 7 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer07Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer07Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 7" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer07Checked den Wert TRUE hat.

Layer08Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 8 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer08Value und Layer08Color festgelegt.

Layer08Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 8 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer08Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer08Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 8" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer08Checked den Wert TRUE hat.

Layer09Checked-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn die Grenze 9 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer09Value und Layer09Color festgelegt.

Layer09Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 9 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer09Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer09Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 9" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer09Checked den Wert TRUE hat.

Layer10Checked (RT Professional)

Beschreibung

TRUE, wenn die Grenze 10 überwacht werden soll. BOOLEAN Schreib-Lese-Zugriff.
Grenzwert und Darstellung werden mit den Eigenschaften Layer10Value und Layer10Color festgelegt.

Layer10Color (RT Professional)

Beschreibung

Legt die Farbe für die Grenze 10 fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.
Wenn die Überwachung des Grenzwerts aktiviert ist (Eigenschaft Layer10Checked), erhält der Balken mit dem Erreichen des Grenzwerts die Farbe dieses Attributs.

Layer10Value (RT Professional)

Beschreibung

Legt den Wert für "Grenze 10" fest oder gibt ihn zurück.
Die Überwachung ist nur dann wirksam, wenn Eigenschaft Layer10Checked den Wert TRUE hat.

LightEffect-Eigenschaft (RT Professional)

Beschreibung

TRUE, wenn der Lichteffekt eingeschaltet ist. BOOLEAN Schreib-Lese-Zugriff.

ListType-Eigenschaft (RT Professional)

Beschreibung

Gibt beim Objekt Textliste den dargestellten Datentyp zurück. Nur Lese-Zugriff.

Wertebereich von 0 bis 2.

0 = Dezimal

1 = Binär

2 = Bit

Localizable (RT Professional)

Beschreibung

Legt fest, ob für jede Runtime-Sprache eine Schriftart zugewiesen und formatiert werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Localizable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl", "OnlineTableControl", "AlarmControl" oder "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn für jede Runtime-Sprache eine Schriftart zugewiesen und formatiert werden kann.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

AlarmControl (Seite 1338)

Max-Eigenschaft (RT Professional)

Beschreibung

Legt den Absolutwert bei voller Wertanzeige fest oder gibt ihn zurück. Ist die Skalenanzeige aktiv, so wird dieser Wert angezeigt.

PersistentRTCSAuthorization (RT Professional)

Beschreibung

Legt die Bedienberechtigung fest, um die Persistenz-Einstellung in Runtime und ES verändern zu können. Der einzugebende Wert entspricht der Nummer, die die gewünschte Berechtigung im Benutzerverwaltung besitzt. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "AllowPersistence". Dies gilt nicht für die Meldeanzeige.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PersistentRTCSAuthorization**[=RtAuthorization]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "AlarmControl", "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

RtAuthorization

Optional. Ein Wert oder eine Konstante, welche die Bedienberechtigung fest, um die Persistenz-Einstellung verändern zu können.

Siehe auch

AlarmControl (Seite 1338)

PersistentToDownload (RT Professional)

Beschreibung

Legt fest, ob die Einstellungen bis zum nächsten Download vom ES erhalten bleiben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PersistentToDownload**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn die Einstellungen bis zum nächsten Download vom ES erhalten bleiben.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

PersistentToDownloadAuthorization (RT Professional)

Beschreibung

Legt die Zugriffsberechtigung fest, um die Einstellungen bis zum nächsten Download vom ES ändern zu können.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PersistentToDownloadAuthorization**[=HmiRTAuthorization]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

HmiRTAuthorization

Optional. Ein Wert oder eine Konstante, die die Zugriffsberechtigung festlegt, um die Einstellungen bis zum nächsten Download vom ES ändern zu können.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

PicDownReferenced (RT Professional)

Beschreibung

TRUE, wenn das zugeordnete Bild für den Zustand "Ein" gespeichert wird. Sonst wird nur der zugehörige Objektverweis gespeichert. Nur Lese-Zugriff.

PicReferenced (RT Professional)

Beschreibung

TRUE, wenn das zugeordnete Bild referenziert und nicht im Objekt gespeichert wird. Nur Lese-Zugriff.

Position (RT Professional)

Beschreibung

Legt die Voreinstellung für die Position des Schiebers fest.

In Runtime wird dieser Wert als Anfangswert verwendet.

Für die Bedienung des mit diesem Attribut verbundenen Prozesswerts ist es notwendig, den Prozesswert auch mit dem Ereignis "Position" zu verbinden. Sie finden das Ereignis "Position" in der Registerkarte "Ereignis", im Themenbaum unter SliderCtrl\Propertythemen\Control Eigenschaften\Wert.

PredefinedAngles (RT Professional)

Beschreibung

Legt die Tiefendarstellung beim Objekt 3DBarGraph fest oder gibt sie zurück. Wertebereich von 0 bis 3.

0 = Kavalier

1 = Isometrisch

2 = Axonometrisch

3 = frei definiert

PrintConfiguration (RT Professional)

Beschreibung

Legt das Protokoll für die Druckausgabe fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**PrintConfiguration**[=HMIPrintConfiguration]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl", "FunctionTrendControl", "OnlineTableControl", "AlarmControl" oder "UserArchiveControl".

HMIPrintConfiguration

Optional. Ein Wert oder eine Konstante, die das Protokoll für die Druckausgabe festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)
AlarmControl (Seite 1338)

ProjectPath (RT Professional)

Beschreibung

Enthält Pfad und Namen des zugehörigen Projekts.
Zugriff in Runtime: Nur Lesen

Syntax

Object.**ProjectPath**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

String

Optional. Ein Wert oder eine Konstante, welche Pfad und Namen des zugehörigen Projekts enthält.

Siehe auch

AlarmControl (Seite 1338)

RowSizingEnable (RT Professional)

Beschreibung

Legt fest, ob die Höhe der Zeile verändert werden kann.
Die Eigenschaft "RowSizingEnable" ist nur dann deaktiviert, wenn beide Eigenschaften "RowSizingEnable" und "AdaptFontSizeToLineHeight" den Wert FALSE haben.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RowSizingEnable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Höhe der Zeile verändert werden kann.

Siehe auch

AlarmControl (Seite 1338)

Scrollable (RT Professional)

Beschreibung

Legt fest, ob die DXF-Bildanzeige im Runtime-Modus die Scrollfunktionen unterstützt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Scrollable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "DXFView".

BOOLEAN

Optional. TRUE, wenn die DXF-Bildanzeige im Runtime-Modus die Scrollfunktionen unterstützt.

ShowMessagesAtDate (Panels, Comfort Panels, RT Advanced)

Beschreibung

Legt fest, ab welchem Zeitpunkt die Meldungen angezeigt werden.

Hinweis

Diese Funktion filtert keine Meldungen heraus, sondern positioniert den Anzeigebereich der Meldeliste ab dem angegebenen Datum.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ShowMessagesAtDate [= DATE]

Object

Erforderlich. Ein Objekt vom Typ "MessageView".

Bemerkung

Wenn Sie der Eigenschaft "ShowMessagesAtDate" SmartTags zuweisen wollen, dann müssen Sie die Zuweisung wie folgt formulieren:

```
'Beispiel 1
HmiRuntime.Screens("Screen_1").ScreenItems("Alarm
view_1").ShowMessagesAtDate = SmartTags("intern_1").Value

'Beispiel 2
HmiRuntime.Screens("Screen_1").ScreenItems("Alarm
view_1").ShowMessagesAtDate = CDate(SmartTags("intern_1"))
SmartTag "intern_1" muss vom Typ "DateTime" sein.
```

Siehe auch

AlarmView (Seite 1350)

Sort (RT Professional)

Beschreibung

Legt Sortierkriterien in SQL-Syntax für die Datenbank fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Sort**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

String

Optional. Ein Wert oder eine Konstante, welche die Sortierkriterien in SQL-Syntax für die Datenbank festlegt.

TableFocusOnButtonCommand (RT Professional)

Beschreibung

Legt fest, ob beim Klicken auf einen Button in einem Skript in Runtime die Tabelle des Controls aktiviert wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TableFocusOnButtonCommand**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmView".

BOOLEAN

Optional. TRUE, wenn beim Klicken auf einen Button in einem Skript in Runtime die Tabelle des Controls aktiviert wird.

Siehe auch

AlarmControl (Seite 1338)

TimeRangeBase(i) (RT Professional)

Beschreibung

Legt die Einheit fest für die Bestimmung der Zeitintervalle der angegebenen Kurve.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeRangeBase(i)**[=TagLoggingTimeUnit]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

TagLoggingTimeUnit

(1): Die Zeitintervalle werden in Millisekunden festgelegt.

hmiCurveTimeRangeBase500ms (500): Die Zeitintervalle werden in 500 ms festgelegt.

hmiCurveTimeRangeBaseSecond (1000): Die Zeitintervalle werden in Sekunden festgelegt.

hmiCurveTimeRangeBaseMinute (60000): Die Zeitintervalle werden in Minuten festgelegt.

hmiCurveTimeRangeBaseHour (3600000): Die Zeitintervalle werden in Stunden festgelegt.

hmiCurveTimeRangeBaseDay (86400000): Die Zeitintervalle werden in Tagen festgelegt.

Bemerkungen

Der für die Kurve i darzustellende Zeitbereich ergibt sich aus der Multiplikation der Werte "TimeRangeBase(i)" und "TimeRangeFactor(i)", wobei der Wert von "TimeRangeBase(i)" in Millisekunden interpretiert wird. Die Eigenschaften "TimeRangeBase(i)" und "TimeRangeFactor(i)" werden nur ausgewertet, wenn die Eigenschaft "TimeRange" den Wert "-1" hat.

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)

TimeRangeFactor(i) (RT Professional)

Beschreibung

Legt den Faktor fest für die Bestimmung der Zeitintervalle der angegebenen Kurve.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeRangeFactor**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend", "FunctionTrendView" oder "TableView".

Int

Optional. Ein Wert oder eine Konstante, welche den Faktor festlegt für die Bestimmung der Zeitintervalle der angegebenen Kurve.

Bemerkungen

Der für die Kurve i darzustellende Zeitbereich ergibt sich aus der Multiplikation der Werte "TimeRangeBase(i)" und "TimeRangeFactor(i)", wobei der Wert von "TimeRangeBase(i)" in Millisekunden interpretiert wird. Die Eigenschaften "TimeRangeBase(i)" und "TimeRangeFactor(i)" werden nur ausgewertet, wenn die Eigenschaft "TimeRange" den Wert "-1" hat.

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)

UseTimeRange(i) (RT Professional)

Beschreibung

Legt fest, wie der darzustellende Zeitbereich für die angegebene Kurve definiert ist.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseTimeRange**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend", "FunctionTrendView" oder "TableView".

BOOLEAN

Optional. TRUE, wenn der darzustellende Zeitbereich durch einen Startzeitpunkt ("TimeAxisBeginTime(i)") und einen Endzeitpunkt ("TimeAxisEndTime") festgelegt ist.

FALSE, wenn der darzustellende Zeitbereich durch einen Startzeitpunkt ("TimeAxisBeginTime(i)") und einen Zeitbereich "TimeRangeBase(i)" und "TimeRangeFactor(i)" festgelegt ist.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

ValidateFormatPattern (RT Professional)

Beschreibung

Legt fest, dass das Feld "Darstellungsformat" auf bestimmte Zeichen überprüft wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ValidateFormatPattern**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "IOField".

BOOLEAN

Optional. TRUE, wenn das Feld "Darstellungsformat" auf bestimmte Zeichen überprüft wird.

Siehe auch

IOField (Seite 1418)

ValueAxisBegin(i) (RT Professional)

Beschreibung

Legt die untere Grenze des darzustellenden Wertebereichs der angegebenen Kurve fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "Autorange" und "ShareValueAxis".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ValueAxisBegin**(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, welche die untere Grenze des darzustellenden Wertebereichs der angegebenen Kurve festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ValueAxisEnd(i) (RT Professional)

Beschreibung

Legt die obere Grenze des darzustellenden Wertebereichs der angegebenen Kurve fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "Autorange" und "ShareValueAxis".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ValueAxisEnd**(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, welche die obere Grenze des darzustellenden Wertebereichs der angegebenen Kurve festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ZeroPointValue (RT Professional)

Beschreibung

Legt den Wert des Nullpunktes der Skalenanzeige fest.

Legt den absoluten Wert für den Nullpunkt fest oder gibt ihn zurück.

Zoomable (RT Professional)

Beschreibung

Legt fest, ob die DXF-Bildanzeige im Runtime-Modus die Zoomfunktionen unterstützt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Zoomable**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "DXFView".

BOOLEAN

Optional. TRUE, wenn die DXF-Bildanzeige im Runtime-Modus die Zoomfunktionen unterstützt.

3 (Panels, Comfort Panels, RT Advanced, RT Professional)

Active (RT Professional)

Beschreibung

Legt fest, ob die darzustellenden Daten vom Archivserver angefordert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Active**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn die darzustellenden Daten vom Archivserver angefordert werden.

Bemerkungen

Wenn Sie dieses Attribut nicht setzen, dann verringern sich die Bildaufschlagszeiten. Ändern Sie den Wert bei Bedarf dynamisch.

Um die "Activate"-Eigenschaft von der "Activate"-Methode zu unterscheiden, wird die Eigenschaft über "Object" angesprochen.

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

AdaptFontSizeToLineHeight (RT Professional)

Beschreibung

Legt fest, ob die Schriftgröße der Höhe der Zeile angepasst wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

`Object.AdaptFontSizeToLineHeight[=BOOLEAN]`

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "AlarmControl".

BOOLEAN

Optional. TRUE, wenn die Schriftgröße der Höhe der Zeile angepasst wird.

Siehe auch

AlarmControl (Seite 1338)

AddAssignment (RT Professional)

Beschreibung

Fügt einen Eintrag hinzu.

AdjustRulerWindow (RT Professional)

Beschreibung

Legt fest, ob das Linealfenster bei jedem Aufblenden der Kurvenanzeige angepasst wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AdjustRulerWindow**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

BOOLEAN

Optional. TRUE, wenn das Linealfenster bei jedem Aufblenden der Kurvenanzeige angepasst wird.

Bemerkungen

TRUE, wenn Sie das Linealfenster verschieben und dann aus- und wieder einblenden, wird es wieder an der ursprünglichen Position mit der ursprünglichen Größe dargestellt.

Siehe auch

OnlineTrendControl (Seite 1443)

AlarmHigh (RT Professional)

Beschreibung

Legt den oberen Grenzwert fest, bei dem Alarm ausgelöst wird oder gibt ihn zurück. Die Art der Auswertung (prozentual oder absolut) legen Sie mit der Eigenschaft "TypeAlarmHigh" fest.

Zugriff in Runtime: Schreiben und Lesen

Syntax

Object.**AlarmHigh**[=REAL]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

REAL

Wert für den oberen Grenzwert

Siehe auch

Bar (Seite 1353)

Alignment (RT Professional)

Beschreibung

Legt die Darstellung der Skala (links/rechts oder oben/unten) abhängig von der Lage des Bar-Objekts fest oder gibt sie zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Alignment**[=ScalePosition]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "Bar".

ScalePosition

1: Rechts oder unten .

0: Links oder oben.

Siehe auch

Bar (Seite 1353)

AlignmentLeft (RT Professional)

Beschreibung

Legt die horizontale Ausrichtung des Texts fest oder gibt sie zurück. Wertebereich von 0 bis 2.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AlignmentLeft**[=HorizontalAlignment]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "TextField", "IOField", "SymbolicIOField", "Button", "RoundButton", "CheckBox", "OptionGroup", "MultilineEdit", "ComboBox", "ListBox".

HorizontalAlignment

0: Ausrichtung links.

1: Ausrichtung zentriert

2: Ausrichtung rechts

AllowPersistence (RT Professional)

Beschreibung

Legt fest, ob die Persistenz einstellbar ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**AllowPersistence**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl", "FunctionTrendControl" oder "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn die Persistenz einstellbar ist.

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[FunctionTrendControl \(Seite 1398\)](#)

[OnlineTableControl \(Seite 1434\)](#)

BackColor2 (RT Professional)

Beschreibung

Legt die Balkenfarbe für die Anzeige des aktuellen Werts fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

BackFillStyleReadOnlySpecial (RT Professional)

Beschreibung

Legt fest, dass das Feld "Füllart Hintergrund" nur gelesen werden kann.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BackFillStyleReadOnlySpecial**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "IOField".

BOOLEAN

Optional. TRUE, wenn das Feld "Füllart Hintergrund" nur gelesen werden kann.

Siehe auch

IOField (Seite 1418)

BarStartValue (RT Professional)

Beschreibung

Legt den Wert des Nullpunktes der Skala fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BarStartValue**[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

Double

Optional. Ein Wert oder eine Konstante, welche den Wert des Nullpunktes der Skala festlegt.

Siehe auch

Bar (Seite 1353)

BarWidth (RT Professional)

Beschreibung

Legt die Balkenbreite in Pixel fest oder gibt sie zurück.

BeginTime(i) (RT Professional)

Beschreibung

Legt den Startzeitpunkt des in der Kurvenanzeige darzustellenden Zeitbereichs fest. Die Eigenschaft "TagProviderType(i)" muss den Wert -1 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**BeginTime**(i)[=Time]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "FunctionTrendControl" oder "OnlineTableControl".

Time

Optional. Ein Wert oder eine Konstante, die den Startzeitpunkt des in der Kurvenanzeige darzustellenden Zeitbereichs festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Bei der Änderung von "BeginTime(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "BeginTime(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

BorderColorBottom (RT Professional)

Beschreibung

Legt die Randfarbe für den unteren/rechten Teil des Objekts fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

BorderColorTop (RT Professional)

Beschreibung

Legt die Randfarbe für den oberen/linken Teil des Objekts fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

CentrePoint (RT Professional)

Beschreibung

Legt die Position des Mittelpunkts fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CentrePoint**[=Point]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "EllipseSegment", "CircleSegment", "EllipticalArc" oder "CircularArc".

Point

Optional. Ein Wert, der die Position des Mittelpunkts festlegt.

Siehe auch

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)

CircularArc (Seite 1374)

CentrePointLeft (RT Professional)

Beschreibung

Legt den horizontalen Abstand des Mittelpunkts vom linken Bildrand in Pixel fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CentrePointLeft**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "EllipseSegment", "CircleSegment", "EllipticalArc" oder "CircularArc".

Int

Optional. Ein Wert, der den horizontalen Abstand des Mittelpunkts vom linken Bildrand in Pixel festlegt.

Siehe auch

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)

CircularArc (Seite 1374)

CentrePointTop (RT Professional)

Beschreibung

Legt den vertikalen Abstand des Mittelpunkts vom oberen Bildrand in Pixel fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CentrePointTop**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "EllipseSegment", "CircleSegment", "EllipticalArc" oder "CircularArc".

Int

Optional. Ein Wert, der den vertikalen Abstand des Mittelpunkts vom oberen Bildrand in Pixel festlegt.

Siehe auch

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)

CircularArc (Seite 1374)

ColorBottom (RT Professional)

Beschreibung

Legt die Farbe für den unteren/rechten Anschlag des Slider-Objektes fest oder gibt sie zurück.
LONG Schreib-Lese-Zugriff.

ColorChangeType (RT Professional)

Beschreibung

TRUE, wenn bei einer Farbänderung (z.B. beim Erreichen eines Grenzwerts) der Farbumschlag segmentweise erfolgen soll. Bei FALSE gilt der Farbumschlag für den gesamten Balken. BOOLEAN Schreib-Lese-Zugriff.

ColorTop (RT Professional)

Beschreibung

Legt die Farbe für den oberen/linken Anschlag des Slider-Objektes fest oder gibt sie zurück.
LONG Schreib-Lese-Zugriff.

ColumnColor(i) (RT Professional)

Beschreibung

Legt die Farbe des Spaltenpaares i fest. Der Parameter i gibt die Nummer des Spaltenpaares an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColumnColor(i)**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe des Spaltenpaares i festlegt.

Siehe auch

OnlineTableControl (Seite 1434)

ColumnDisplayName(i) (RT Professional)

Beschreibung

Legt den Namen für das angegebene Spaltenpaar fest. Der Parameter i gibt die Nummer des Spaltenpaars an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColumnDisplayName(i)**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

String

Optional. Ein Wert oder eine Konstante, welche den Namen für das angegebene Spaltenpaar festlegt.

Siehe auch

OnlineTableControl (Seite 1434)

ColumnUpdateEnabled(i) (RT Professional)

Beschreibung

Legt fest, ob für das angegebene Spaltenpaar statisch oder dynamisch dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ColumnUpdateEnabled(i)**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn das Spaltenpaar statisch dargestellt wird.

Bemerkungen

Die Eigenschaft "CurrentColumnIndex" referenziert das im Augenblick aktive Spaltenpaar.

Siehe auch

OnlineTableControl (Seite 1434)

Command (RT Professional)

Beschreibung

Legt fest, ob das Aktualisieren der im Control dargestellten Werte erzwungen wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Command**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "OnlineTableControl".

String

Optional. Ein Wert oder eine Konstante, welche festlegt, ob die Aktualisierung der im Control dargestellten Werte erzwungen wird.

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

Comment (RT Professional)

Beschreibung

Liest oder setzt den Kommentar des Alarm-Objekts.

CommonTimeAxisColor (RT Professional)

Beschreibung

Legt die Farbe der gemeinsamen X-Achse fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CommonTimeAxisColor**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der gemeinsamen X-Achse festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)

ConfigureTimeAxis(i) (RT Professional)

Beschreibung

Legt fest, ob in der Kurvenanzeige für alle Kurven eine gemeinsame Zeitachse verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ConfigureTimeAxis(i)**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

BOOLEAN

Optional. TRUE, wenn in der Kurvenanzeige für alle Kurven eine gemeinsame Zeitachse verwendet wird.

Siehe auch

OnlineTrendControl (Seite 1443)

CurrentCurveIndex (RT Professional)

Beschreibung

Legt den Index der aktiven Kurve fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CurrentCurveIndex**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "FunctionTrendControl" oder "OnlineTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche den Index der im Augenblick aktiven Kurve festlegt.

Bemerkungen

Die Eigenschaft "CurrentCurveIndex" wird von anderen Eigenschaften ausgewertet, um deren Einstellungen einer bestimmten Kurve zuordnen zu können. Gültige Werte für den Index bewegen sich im Bereich von 0 bis - 1. Die Eigenschaft "CurvesCount" enthält die Anzahl der anzuzeigenden Kurven.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

CurveColor(i) (RT Professional)

Beschreibung

Legt die Farbe der angegebenen Kurve fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CurveColor(i)**[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe der angegebenen Kurve festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

CurveLineWidth(i) (RT Professional)

Beschreibung

Legt die Strichstärke der angegebenen Kurve fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CurveLineWidth(i)**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Strichstärke der angegebenen Kurve festlegt. Wertebereich von 0 bis 10.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

CurvesCount (RT Professional)

Beschreibung

Gibt die Anzahl der projizierten Kurven oder Spaltenpaare (sichtbare und unsichtbare) des Fensters zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CurvesCount**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der projizierten Kurven oder Spaltenpaare (sichtbare und unsichtbare) des Fensters zurückgibt.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

CurveUpdateEnabled(i) (RT Professional)

Beschreibung

Legt fest, ob für die angegebene Kurve statisch oder dynamisch dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**CurveUpdateEnabled(i)**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

BOOLEAN

Optional. TRUE, wenn die Kurve statisch dargestellt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

DataIndex(i) (RT Professional)

Beschreibung

Gibt den aktuellen Index der Daten der aktuellen Kurve zurück.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DataIndex(i)**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche den aktuellen Index der Daten der aktuellen Kurve zurückgibt.

Siehe auch

FunctionTrendControl (Seite 1398)

DataLogTag (RT Professional)

Beschreibung

Legt fest, welche Variable mit der angegebenen Kurve verknüpft ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DataLogTag**[=HmiLoggingTag]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "OnlineTableControl".

HmiLoggingTag

Optional. Ein Wert oder eine Konstante, die welche Variable festlegt, die mit der angegebenen Kurve verknüpft ist.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

DataX(i) (RT Professional)

Beschreibung

Fügt einen einzelnen Datensatz ein und muss vor dem Aufruf von "InsertData(i)" gesetzt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DataX(i)**[=object]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

object

Optional. Ein Wert oder eine Konstante, die einen einzelnen Datensatz einfügt.

Siehe auch

FunctionTrendControl (Seite 1398)

DataXY(i) (RT Professional)

Beschreibung

Fügt mehrere Datensätze als Array mit Wertepaaren ein und muss vor dem Aufruf von "InsertData(i)" gesetzt werden.

Die Daten im Array werden übernommen, wenn "DataX(i)" vom Typ VT_EMPTY ist. Im anderen Fall wird im Attribut "InsertData(i)" das einzelne Wertepaar verwendet, das sich aus "DataX(i)" und "DataY(i)" ergibt.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DataXY(i)**[=object]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

object

Optional. Ein Wert oder eine Konstante, die mehrere Datensätze als Array mit Wertepaaren einfügt.

Siehe auch

FunctionTrendControl (Seite 1398)

DataY(i) (RT Professional)

Beschreibung

Fügt einen einzelnen Datensatz ein und muss vor dem Aufruf von "InsertData(i)" gesetzt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DataY**(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, welche einen einzelnen Datensatz einfügt.

Siehe auch

FunctionTrendControl (Seite 1398)

DeleteData(i) (RT Professional)

Beschreibung

Legt fest, welche Daten des Datenpuffers der aktiven Kurve gelöscht werden.

TRUE:

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DeleteData**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn alle Daten der Kurve gelöscht werden.

FALSE, wenn das Wertepaar an der Position "DataIndex(i)" gelöscht wird.

Siehe auch

FunctionTrendControl (Seite 1398)

Direction (RT Professional)

Beschreibung

Legt die Balkenrichtung bzw. die Lage des Slider-Objektes fest oder gibt sie zurück.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Direction** [=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "Bar" oder "Slider".

BOOLEAN

0 = oben

1 = unten

2 = links

3 = rechts

DisplayName(i) (RT Professional)

Beschreibung

Legt den Namen für die angegebene Kurve fest.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DisplayName(i)**[=String]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

String

Optional. Ein Wert oder eine Konstante, welche den Namen für die angegebene Kurve festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

FunctionTrendControl (Seite 1398)

DrawEnhancedHTMLBrowser (RT Professional)

Beschreibung

Legt den Stil des HTML-Browsers fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**DrawEnhancedHTMLBrowser**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "HTMLBrowser".

BOOLEAN

Optional. TRUE, wenn ein erweiterter HTML-Browser angezeigt wird.

Siehe auch

HTMLBrowser (Seite 1416)

Editable(i) (RT Professional)

Beschreibung

Legt fest, ob ein Spaltenpaar editierbar ist. Der Parameter i gibt die Nummer des Spaltenpaars an.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**Editable**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn ein Spaltenpaar editierbar ist. Das Spaltenpaar wird über den Parameter i referenziert.

Siehe auch

OnlineTableControl (Seite 1434)

EditEnabled (RT Professional)

Beschreibung

Legt fest, ob der Editiermodus für eine Zelle aktiviert ist, wenn deren Eigenschaft "Editable(i)" den Wert TRUE hat.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EditEnabled**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn der Editiermodus für eine Zelle aktiviert ist, wenn deren Eigenschaft "Editable(i)" den Wert TRUE hat.

Siehe auch

OnlineTableControl (Seite 1434)

EndPoint (RT Professional)

Beschreibung

Legt die Position des Endpunktes fest.

Zugriff in Runtime:

Syntax

Object.**EndPoint**[=Point]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "EllipseSegment", "CircleSegment", "EllipticalArc" oder "CircularArc".

Point

Optional. Ein Wert, der die Position des Endpunktes festlegt.

Siehe auch

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)

CircularArc (Seite 1374)

EndPointLeft (RT Professional)

Beschreibung

Legt den horizontalen Abstand des Endpunkts vom linken Bildrand in Pixel fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EndPointLeft**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "EllipseSegment", "CircleSegment", "EllipticalArc" oder "CircularArc".

Int

Optional. Ein Wert, der den horizontalen Abstand des Endpunkts vom linken Bildrand in Pixel festlegt.

Siehe auch

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)

CircularArc (Seite 1374)

EndPointTop (RT Professional)

Beschreibung

Legt den vertikalen Abstand des Endpunkts vom oberen Bildrand in Pixel fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EndPointTop**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "EllipseSegment", "CircleSegment", "EllipticalArc" oder "CircularArc".

Int

Optional. Ein Wert, der den vertikalen Abstand des Endpunkts vom oberen Bildrand in Pixel festlegt.

Siehe auch

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)

CircularArc (Seite 1374)

EndTime(i) (RT Professional)

Beschreibung

Legt den Endzeitpunkt des in der Kurvenanzeige darzustellenden Zeitbereichs fest. Die Eigenschaft "TagProviderType(i)" muss den Wert -1 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**EndTime**(i)[=Time]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "FunctionTrendControl" oder "OnlineTabelControl".

Time

Optional. Ein Wert oder eine Konstante, die den Endzeitpunkt des im Kurvenfenster darzustellenden Zeitbereichs festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Bei der Änderung von "EndTime(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "EndTime(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

OnlineTableControl (Seite 1434)

ExportXML (Panels, Comfort Panels, RT Advanced)

ExportXML

Wird nur intern verwendet.

Das Attribut ist mit dem Namen **ExportXML** dynamisierbar.

ExtendedOperation (RT Professional)

Beschreibung

TRUE, wenn der Schieberegler auf den zugehörigen Endwert (Minimalwert/Maximalwert) gestellt wird. Dies geschieht bei Mausklick auf den Bereich außerhalb der aktuellen Reglereinstellung. BOOLEAN Schreib-Lese-Zugriff.

ExtraSpaceForLabelDisplay (RT Professional)

Beschreibung

Legt den zusätzlichen Platz für die Beschriftung fest.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ExtraSpaceForLabelDisplay**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

Int

Optional. Ein Wert, der den zusätzlichen Platz für die Beschriftung festlegt.

Siehe auch

Bar (Seite 1353)

FlashFlashPicture (RT Professional)

Beschreibung

TRUE, wenn das Blinken des Blinkbildes aktiviert ist. BOOLEAN Schreib-Lese-Zugriff.

FlashForeColor (RT Professional)

Beschreibung

TRUE, wenn das Blinken des Texts aktiviert ist. BOOLEAN Schreib-Lese-Zugriff.

FlashPicReferenced (RT Professional)

Beschreibung

TRUE, wenn die zugeordnete Blinkgrafik gespeichert werden soll. Sonst wird nur der zugehörige Objektverweis gespeichert. Nur Lese-Zugriff.

FlashPicture (RT Professional)

Beschreibung

Gibt die Blinkgrafik zurück. Nur Lese-Zugriff.
Die Grafik (*.BMP oder *.DIB) muss sich im Ordner "GraCS" des aktuellen Projekts befinden, damit sie eingebunden werden kann.
Die Eigenschaft "FlashPicReferenced" legt in diesem Zusammenhang fest, ob die Blinkgrafik zusammen mit dem Objekt gespeichert oder referenziert wird.

FlashRateBorderColor (RT Professional)

Beschreibung

Legt die Blinkfrequenz für die Linie des Objektes fest oder gibt sie zurück. Wertebereich von 0 bis 2.

0 = langsam

1 = mittel

2 = schnell

FlashRateFlashPic (RT Professional)

Beschreibung

Legt die Blinkfrequenz für die Blinkgrafik fest oder gibt sie zurück. Wertebereich von 0 bis 2.

0 = langsam

1 = mittel

2 = schnell

FlashRateForeColor (RT Professional)

Beschreibung

Legt die Blinkfrequenz für die Beschriftung des Objektes fest oder gibt sie zurück. Wertebereich von 0 bis 2.

0 = langsam

1 = mittel

2 = schnell

FormatPatternReadOnlySpecial (RT Professional)

Beschreibung

Legt fest, dass das Feld "Darstellungsformat" nur gelesen werden kann.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FormatPatternReadOnlySpecial**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "IOField".

BOOLEAN

Optional. TRUE, wenn das Feld "Darstellungsformat" nur gelesen werden kann.

Siehe auch

IOField (Seite 1418)

FreezeProviderConnections (RT Professional)

Beschreibung

Legt fest, ob Eigenschaften zur Datenanbindung ("TagProviderType(i)", "Source.."...) geändert werden können, ohne dass die Änderung sofort wirksam wird. Beim Verändern von z. B. "XDataLogTag(i)" können unzulässige Kombinationen mit "YDataLogTag(i)" entstehen.

Deshalb muss vor dem Ändern eines Attributs zur Datenanbindung "FreezeProviderConnections" den Wert TRUE haben. Nach dem Ändern aller Eigenschaften zur Datenanbindung wird "FreezeProviderConnection" auf FALSE gesetzt und die Änderungen werden wirksam.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**FreezeProviderConnections**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn Eigenschaften zur Datenanbindung ("TagProviderType(i)", "Source.."...) geändert werden können, ohne dass die Änderung sofort wirksam wird.

Siehe auch

FunctionTrendControl (Seite 1398)

GetSelectionText (RT Professional)

Beschreibung

Gibt den ausgewählten Text des Objekts "MultiLineEdit" zurück.

HideTagNames (RT Professional)

Beschreibung

Legt fest, ob die Variablennamen in f(t)-Kurvenanzeige eingeblendet sind.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**HideTagNames**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

BOOLEAN

Optional. TRUE, wenn die Variablennamen eingeblendet sind.

Siehe auch

OnlineTrendControl (Seite 1443)

LanguageSwitch (RT Professional)

Beschreibung

Gibt den Wert zurück, der festlegt, wo die sprachabhängigen Zuordnungstexte abgelegt werden. Nur Lese-Zugriff.

TRUE, wenn die Texte im Projekttexte-Editor verwaltet werden. Übersetzungen in andere Sprachen erfolgen im Projekttexte-Editor.

FALSE, wenn die Texte direkt im Objekt verwaltet werden. Übersetzungen in andere Sprachen können mittels Export/Import-Funktion vorgenommen werden.

LeaveMarginForBorder (RT Professional)

Beschreibung

Legt fest, ob ein zusätzlicher Rand für Rahmen freigelassen wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LeaveMarginForBorder**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn ein zusätzlicher Rand für Rahmen freigelassen wird.

Siehe auch

Bar (Seite 1353)

LeaveMarginForMarkers (RT Professional)

Beschreibung

Legt fest, ob ein zusätzlicher Rand für Markierungen freigelassen wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**LeaveMarginForMarkers**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn ein zusätzlicher Rand für Marker freigelassen wird.

Siehe auch

Bar (Seite 1353)

Min (RT Professional)

Beschreibung

Legt den Absolutwert bei kleinster Wertanzeige fest oder gibt ihn zurück. Ist die Skalanzeige aktiv, so wird dieser Wert angezeigt.

NumberOfValues(i) (RT Professional)

Beschreibung

Legt die Anzahl der Werte fest, die im Kurvenfenster dargestellt werden. Die Eigenschaft "TagProviderType(i)" muss den Wert -1 haben. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "UseMeasurePoints(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**NumberOfValues**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Werte festlegt, die im Kurvenfenster dargestellt werden.

Bemerkungen

Number of value pairs for curve i (from tag provider)

Siehe auch

FunctionTrendControl (Seite 1398)

Parent (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt das Bild zurück, in dem das angegebene Objekt eingebettet ist.

Zugriff in Runtime: Lesen

Syntax

Object.Parent

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem". Diese Eigenschaft ist eine Standardeigenschaft des ScreenItem-Objekts und damit bei allen Ausprägungen verfügbar.

Anwendungsbeispiel

Das folgende Beispiel schreibt den Namen des Grundbilds in die Variable "BaseScreenName":

```
'VBS_Example_Parent
Dim objScrItem, BaseScreenName
Set objScrItem = HMIRuntime.Screens(1).ScreenItems(1)
BaseScreenName = "Name of BaseScreen: " & objScrItem.Parent.ObjectName
```

Siehe auch

Line (Seite 1422)

Polyline (Seite 1459)

Ellipse (Seite 1386)

Circle (Seite 1369)

EllipseSegment (Seite 1388)

CircleSegment (Seite 1372)

EllipticalArc (Seite 1391)
CircularArc (Seite 1374)
Rectangle (Seite 1471)
Polygon (Seite 1456)
TextField (Seite 1505)
IOField (Seite 1418)
SymbolicIOField (Seite 1491)
Button (Seite 1360)
Switch (Seite 1488)
GraphicView (Seite 1412)
GraphicIOField (Seite 1409)
Bar (Seite 1353)
Clock (Seite 1377)
Gauge (Seite 1406)
Slider (Seite 1481)
SymbolLibrary (Seite 1496)
OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)
OnlineTableControl (Seite 1434)
AlarmControl (Seite 1338)
HTMLBrowser (Seite 1416)
CheckBox (Seite 1366)
OptionGroup (Seite 1452)
WindowSlider (Seite 1538)
Connector (Seite 1379)
ScreenWindow (Seite 1477)
DiskSpaceView (Seite 1384)
ChannelDiagnose (Seite 1364)
ScriptDiagnostics (Seite 1479)
Group (Seite 1415)

PicUpReferenced (RT Professional)

Beschreibung

TRUE, wenn das zugeordnete Bild für den Zustand "Aus" im Objekt gespeichert wird. Sonst wird nur der zugehörige Objektverweis gespeichert. Nur Lese-Zugriff.

RemoveAllAssignment (RT Professional)

Beschreibung

Löscht alle Einträge.

RemoveAssignment (RT Professional)

Beschreibung

Entfernt einen Eintrag.

RowNumber (RT Professional)

Beschreibung

Gibt die Zeilennummer des Row-Objekts eines Tabellen-Controls an.

Siehe auch

AlarmControl (Seite 1338)

RulerPrecision(i) (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen fest, mit denen ein Messwert der angegebene Kurve angezeigt wird, wenn er über die Funktion "Anzeige Wert an dieser Stelle" ermittelt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RulerPrecision**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen festlegt, mit denen ein Messwert der angegebene Kurve angezeigt wird, wenn er über die Funktion "Anzeige Wert an dieser Stelle" ermittelt wird.

Siehe auch

OnlineTrendControl (Seite 1443)

RulerXPrecision(i) (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen fest, mit denen ein Messwert der X-Koordinate angezeigt wird. Ob die Angabe ausgewertet wird, ist abhängig vom Wert des Attributes "XAxisMode(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RulerXPrecision(i)**[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen festlegt, mit denen ein Messwert der X-Koordinate angezeigt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Sie verwenden die Funktion "Anzeige Wert an dieser Stelle", um die X-Koordinate des Messwerts anzeigen zu lassen.

Siehe auch

FunctionTrendControl (Seite 1398)

RulerYPrecision(i) (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen fest, mit denen ein Messwert der Y-Koordinate angezeigt wird. Ob die Angabe ausgewertet wird, ist abhängig vom Wert des Attributes "XAxisMode(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**RulerYPrecision**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen festlegt, mit denen ein Messwert der Y-Koordinate angezeigt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Sie verwenden die Funktion "Anzeige Wert an dieser Stelle", um die Y-Koordinate des Messwerts anzeigen zu lassen.

Siehe auch

FunctionTrendControl (Seite 1398)

SetAllAssignment (RT Professional)

Beschreibung

Setzt alle Einträge.

SetpointTrendArchiveStartId(i) (RT Professional)

Beschreibung

Legt Anfangsposition fest, ab der die Werte der Sollkurve aus der Rezeptur gelesen werden. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "ShowSetpointTrend(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SetpointTrendArchiveStartId**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anfangsposition festlegt, ab der die Werte der Sollkurve aus der Rezeptur gelesen werden.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

FunctionTrendControl (Seite 1398)

SetpointTrendColor(i) (RT Professional)

Beschreibung

Legt die Farbe einer Sollkurve fest. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "ShowSetpointTrend(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SetpointTrendColor**(i)[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe einer Sollkurve festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

FunctionTrendControl (Seite 1398)

SetpointTrendNumberOfValues(i) (RT Professional)

Beschreibung

Legt die Anzahl der Wertepaare einer Sollkurve fest. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "ShowSetpointTrend(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SetpointTrendNumberOfValues**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Wertepaare einer Sollkurve festlegt.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

FunctionTrendControl (Seite 1398)

SetProps (RT Professional)

Beschreibung

Setzt den Wert der Eigenschaft.

SetSelection (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Legt eine Liste der Zeilen, die selektiert sind.

ShareTimeAxis (RT Professional)

Beschreibung

Legt fest, ob die Kurven des Kurvenfensters mit einer gemeinsamen X-Achse dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShareTimeAxis**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

BOOLEAN

Optional. TRUE, wenn die Kurven des Kurvenfensters mit einer gemeinsamen X-Achse dargestellt werden.

Siehe auch

OnlineTrendControl (Seite 1443)

ShareTimeColumn (RT Professional)

Beschreibung

Legt fest, ob im Tabellenfenster eine gemeinsame Zeitspalte verwendet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShareTimeColumn**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn im Tabellenfenster eine gemeinsame Zeitspalte verwendet wird.

Siehe auch

OnlineTableControl (Seite 1434)

ShareValueAxis (RT Professional)

Beschreibung

Legt fest, ob die Kurven des Kurvenfensters mit einer gemeinsamen Y-Achse dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShareValueAxis**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrendControl".

BOOLEAN

Optional. TRUE, wenn die Kurven des Kurvenfensters mit einer gemeinsamen Y-Achse dargestellt werden.

Siehe auch

OnlineTrendControl (Seite 1443)

ShareXAxis (RT Professional)

Beschreibung

Legt fest, ob die Kurven des Kurvenfensters mit einer gemeinsamen X-Achse dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShareXAxis**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn die Kurven des Kurvenfensters mit einer gemeinsamen X-Achse dargestellt werden.

Siehe auch

FunctionTrendControl (Seite 1398)

ShareYAxis (RT Professional)

Beschreibung

Legt fest, ob die Kurven des Kurvenfensters mit einer gemeinsamen Y-Achse dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShareYAxis**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn die Kurven des Kurvenfensters mit einer gemeinsamen Y-Achse dargestellt werden.

Siehe auch

FunctionTrendControl (Seite 1398)

ShowBorder (RT Professional)

Beschreibung

Legt fest, ob das Fenster mit Rahmen dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowBorder**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn das Fenster mit Rahmen dargestellt wird.

ShowColumn(i) (RT Professional)

Beschreibung

Legt fest, ob ein über die Eigenschaft "CurrentColumnIndex" referenziertes Spaltenpaar sichtbar ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowColumn(i)**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTableControl".

BOOLEAN

Optional. TRUE, wenn ein über die Eigenschaft "CurrentColumnIndex" referenziertes Spaltenpaar sichtbar ist.

Siehe auch

OnlineTableControl (Seite 1434)

ShowCurve(i) (RT Professional)

Beschreibung

Legt fest, ob eine über die Eigenschaft "CurrentCurveIndex" referenzierte Kurve sichtbar ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowCurve**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn eine über die Eigenschaft "CurrentCurveIndex" referenzierte Kurve sichtbar ist.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ShowInputControls (Panels, Comfort Panels, RT Advanced)

Beschreibung

Legt fest, ob es zusätzliche Felder für die Eingabe von Adresse und Kennwort gibt.

Syntax

Object.**ShowInputControls**

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "SmartClientView".

ShowMainFrame (RT Professional)

Beschreibung

Legt fest, ob das gesamte Balken-Objekt mit einem Rahmen versehen wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowMainFrame**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Option. TRUE, wenn das gesamte Balkenobjekt mit einem Rahmen versehen wird.

Siehe auch

Bar (Seite 1353)

ShowOverflowIndicator (RT Professional)

Beschreibung

Legt fest, ob eine Überlaufsanzeige eingeblendet wird, wenn der Prozesswert die Grenzwerte über- bzw. unterschreitet.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowOverflowIndicator**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

TRUE, wenn eine Überlaufsanzeige eingeblendet wird, wenn der Prozesswert die Grenzwerte über- bzw. unterschreitet.

Siehe auch

Bar (Seite 1353)

ShowSetpointTrend(i) (RT Professional)

Beschreibung

Legt fest, ob die zu einer über "CurrentCurveIndex" referenzierten Kurve gehörende Sollkurve dargestellt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ShowSetpointTrend**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn die zu einer über "CurrentCurveIndex" referenzierten Kurve gehörende Sollkurve dargestellt wird.

Siehe auch

FunctionTrendControl (Seite 1398)

SmallChange (RT Professional)

Beschreibung

Legt fest, um wieviel Schritte der Regler mit einem Mausklick verschoben werden kann oder gibt diesen Wert zurück.

StartPoint (RT Professional)

Beschreibung

Legt den Beginn des Objektes fest oder gibt ihn zurück.

StatusbarShowArchiveName (RT Professional)

Beschreibung

Legt fest, ob der Archivname in der Statuszeile angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarShowArchiveName**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn der Archivname in der Statuszeile angezeigt wird.

StatusbarShowColumn (RT Professional)

Beschreibung

Legt fest, ob die aktuelle Anzahl der ausgewählten Datensatz-Spalten angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarShowColumn**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn die aktuelle Anzahl der ausgewählten Datensatz-Spalten angezeigt wird.

StatusbarShowRecord (RT Professional)

Beschreibung

Legt fest, ob die Feldkoordinaten des ausgewählten Datensatzes in der Statusleiste angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarShowRecord**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn die Feldkoordinaten des ausgewählten Datensatzes in der Statusleiste angezeigt werden.

StatusbarShowRow (RT Professional)

Beschreibung

Legt fest, ob die aktuelle Anzahl der ausgewählten Datensatz-Zeilen angezeigt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarShowRow**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn die aktuelle Anzahl der ausgewählten Datensatz-Zeilen angezeigt werden.

StatusbarShowText (RT Professional)

Beschreibung

Legt fest, ob der aktuelle Status der Datenbank in der Statusleiste angezeigt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**StatusbarShowText**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "UserArchiveControl".

BOOLEAN

Optional. TRUE, wenn der aktuelle Status der Datenbank in der Statusleiste angezeigt wird.

SwapDimensionsWithOrientation (RT Professional)

Beschreibung

Legt fest, ob die Werte für Höhe und Breite des Balkens bei einer Änderung der Balkenausrichtung automatisch getauscht werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**SwapDimensionsWithOrientation**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn die Werte für Höhe und Breite des Balkens bei einer Änderung der Balkenausrichtung automatisch getauscht werden.

Siehe auch

Bar (Seite 1353)

TimeAxisShowLargeIncrements(i) (RT Professional)

Beschreibung

Legt fest, ob die Zeitachse mit langen Teilstrichen skaliert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeAxisShowLargeIncrements**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrend".

BOOLEAN

Optional. TRUE, wenn die Zeitachse mit langen Teilstrichen skaliert wird.

Bemerkungen

Der Abstand zweier langer Teilstriche kann über die Eigenschaft "TimeAxisLargeIncrementSize(i)" verändert werden.

Siehe auch

OnlineTrendControl (Seite 1443)

TimeAxisShowSmallIncrements(i) (RT Professional)

Beschreibung

Legt fest, ob die Zeitachse mit kurzen Teilstrichen skaliert wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeAxisShowSmallIncrements**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "OnlineTrend".

BOOLEAN

Optional. TRUE, wenn die Zeitachse mit kurzen Teilstrichen skaliert wird.

Bemerkungen

Der Abstand zweier kurzer Teilstriche kann über die Eigenschaft "TimeAxisSmallIncrementSize(i)" verändert werden.

Siehe auch

OnlineTrendControl (Seite 1443)

TimeJumpColor(i) (RT Professional)

Beschreibung

Legt die Farbe fest, die im Archiv vorhandenen Zeitsprünge kennzeichnet. Ob die Angabe ausgewertet wird ist abhängig von der Eigenschaft "TimeJumpEnabled(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeJumpColor**(i)[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend" oder "TableView".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe festlegt, welche die im Archiv vorhandenen Zeitsprünge kennzeichnet.

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

TimeJumpEnabled(i) (RT Professional)

Beschreibung

Legt fest, ob die im Archiv vorhandenen Zeitsprünge mit der in "TimeJumpColor(i)" festgelegten Farbe gekennzeichnet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeJumpEnabled**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend" oder "TableView".

BOOLEAN

Optional. TRUE, wenn die im Archiv vorhandenen Zeitsprünge mit der in "TimeJumpColor(i)" festgelegten Farbe gekennzeichnet werden.

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

TimeOverlapColor(i) (RT Professional)

Beschreibung

Legt die Farbe fest, die im Archiv vorhandene Zeitüberlappungen kennzeichnet. Ob die Angabe ausgewertet wird, ist abhängig vom Attribut "TimeOverlapEnabled(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeOverlapColor**(i)[=Color]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend" oder "TableView".

Color

Optional. Ein Wert oder eine Konstante, welche die Farbe festlegt, welche die im Archiv vorhandenen Zeitüberlappungen kennzeichnet.

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

TimeOverlapEnabled(i) (RT Professional)

Beschreibung

Legt fest, ob die im Archiv vorhandenen Zeitüberlappungen mit der in "TimeOverlapColor(i)" festgelegten Farbe gekennzeichnet werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**TimeOverlapEnabled**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend" oder "TableView".

BOOLEAN

Optional. TRUE, wenn die im Archiv vorhandenen Zeitüberlappungen mit der in "TimeOverlapColor(i)" festgelegten Farbe gekennzeichnet werden.

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

Type (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt den Typ des angegebenen Objekts als STRING zurück. Namensschema: Hmi<Objektbezeichnung>, z.B. HmiCircle für das Bildobjekt "Kreis".

Zugriff in Runtime: Lesen

Syntax

Object.Type

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem".

UnselBGColor-Eigenschaft (RT Professional)

Beschreibung

Legt beim Objekt TextList die Hintergrundfarbe der Auswahlliste bei nicht ausgewählten Einträgen fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

UnselTextColor-Eigenschaft (RT Professional)

Beschreibung

Legt beim Objekt Textliste die Textfarbe in der Auswahlliste bei nicht ausgewählten Einträgen fest oder gibt sie zurück. LONG Schreib-Lese-Zugriff.

UpdateCycle-Eigenschaft (RT Professional)

Beschreibung

Gibt die Art und Häufigkeit für die Aktualisierung des Bildfensters in Runtime zurück. Nur Lese-Zugriff.

UseBarBorderConstraints (RT Professional)

Beschreibung

Legt fest, ob der Balkenrahmen auf bestimmte Werte begrenzt ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseBarBorderConstraints**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn der Balkenrahmen auf bestimmte Werte begrenzt ist.

Siehe auch

Bar (Seite 1353)

UseEffectiveProcessValue (RT Professional)

Beschreibung

Legt fest, ob ein aktueller Prozesswert aus Maximalwert, Minimalwert, Durchschnitt aus den letzten 15 Werten und Hysterese benutzt wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseEffectiveProcessValue**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn ein aktueller Prozesswert aus Maximalwert, Minimalwert, Durchschnitt aus den letzten 15 Werten und Hysterese benutzt wird.

Siehe auch

Bar (Seite 1353)

UseGDI (RT Professional)

Beschreibung

Legt fest, ob die Ellipse über GDI oder GDI+ gezeichnet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseGDI**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Ellipse".

BOOLEAN

Optional. TRUE, wenn die Ellipse über GDI gezeichnet wird.

Siehe auch

Ellipse (Seite 1386)

UseMeasurePoints(i) (RT Professional)

Beschreibung

Legt fest, ob der Zeitbereich für die angegebene Kurve durch Messpunkte oder den Endzeitpunkt bestimmt wird. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "TimeAxisEndTime" und "TimeAxisBeginTime(i)".

Define time range for curve i through measure points (TRUE) or end time (FALSE) (for time based tag providers)

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseMeasurePoints**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn ob der Zeitbereich für die angegebene Kurve durch Messpunkte bestimmt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

UseMultipleLimits (RT Professional)

Beschreibung

Legt fest, ob ein oder mehrere Grenzwertpaare als Markierung bzw. Linie dargestellt werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseMultipleLimits**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn mehrere Grenzwertpaare als Markierung bzw. Linie dargestellt werden.

Siehe auch

Bar (Seite 1353)

UseScaleConstraints (RT Professional)

Beschreibung

Legt fest, ob der Abstand zwischen zwei großen Teilstrichen der Skala aus den Minimum- und Maximumwert errechnet wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseScaleConstraints**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn der Abstand zwischen zwei großen Teilstrichen der Skala aus den Minimum- und Maximumwert errechnet wird.

Siehe auch

Bar (Seite 1353)

UseScaledBarBorder (RT Professional)

Beschreibung

Legt fest, ob das objektumfassende Rechteck abhängig von der Skala dargestellt wird oder vordefiniert ist.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseScaledBarBorder**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn das objektumfassende Rechteck abhängig von der Skala dargestellt wird.

Siehe auch

Bar (Seite 1353)

UseSeparateDiagrams (RT Professional)

Beschreibung

Legt fest, ob die Kurven gestaffelt dargestellt werden.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseSeparateDiagrams**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrend" oder "FunctionTrendView".

BOOLEAN

Optional. TRUE, wenn die Kurven gestaffelt dargestellt werden

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

UseSimplePresicionOffset (RT Professional)

Beschreibung

Legt fest, wie die Feldlänge für die Beschriftung der Skala berechnet wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UseSimplePresicionOffset**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "Bar".

BOOLEAN

Optional. TRUE, wenn die Feldlänge für die Beschriftung der Skala berechnet wird.

Siehe auch

Bar (Seite 1353)

UserArchiveNumberOfValues(i) (RT Professional)

Beschreibung

Legt die Anzahl der Werte fest, die aus der Rezeptur geladen werden. Die Eigenschaft "TagProviderType(i)" muss den Wert -2 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UserArchiveNumberOfValues**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Werte festlegt, die aus der Rezeptur geladen werden.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Bei der Änderung von "UserArchiveNumberOfValues(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "UserArchiveNumberOfValues(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

UserArchiveStartId(i) (RT Professional)

Beschreibung

Legt den Datensatz fest, ab dem die Werte für die angegebene Kurve aus der Rezeptur geladen werden. Die Eigenschaft "TagProviderType(i)" muss den Wert -2 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UserArchiveStartId**(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

Int

Optional. Ein Wert oder eine Konstante, welche den Datensatz festlegt, ab dem die Werte für die angegebene Kurve aus der Rezeptur geladen werden.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Bei der Änderung von "UserArchiveStartId(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "UserArchiveStartId(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

UserData (RT Professional)

Beschreibung

Legt den Wert fest, der beim Ausführen eines benutzerdefinierten Menüeintrags oder Symbols an das VB-Script übergeben wird.

Verwenden Sie im Editor "Menüs und Symbolleisten" das Feld "Daten", um an die Prozedur einen Parameter zu übergeben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**UserData**[=String]

Object

Erforderlich. Ein Objekt vom Typ "Item".

String

Optional. Ein Wert oder eine Konstante, die beim Ausführen eines benutzerdefinierten Menüeintrags oder Symbols an das VB-Script übergeben wird.

ValueAxisGridLineInterval(i) (RT Professional)

Beschreibung

Legt den Abstand zweier Rasterlinien fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "ValueAxisShowGridLines(i)" und "TimeAxisShowGridLines(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueAxisGridLineInterval(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, welche den Abstand zweier Rasterlinien festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ValueAxisLargeIncrementSize(i) (RT Professional)

Beschreibung

Legt den Abstand zwischen zwei langen Teilstrichen der Skalierung fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "ValueAxisShowLargeIncrements(i)" und "TimeAxisShowLargeIncrements(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueAxisLargeIncrementSize(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, welche den Abstand zwischen zwei langen Teilstrichen der Skalierung festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)

ValueAxisShowLargeIncrements(i) (RT Professional)

Beschreibung

Legt fest, ob die Werteachse mit langen Teilstrichen skaliert wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ValueAxisShowLargeIncrements**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn die Werteachse mit langen Teilstrichen skaliert wird.

Bemerkungen

Der Abstand zweier langer Teilstriche kann über die Eigenschaft "ValueAxisLargeIncrementSize(i)" verändert werden.

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)

ValueAxisShowSmallIncrements(i) (RT Professional)

Beschreibung

Legt fest, ob die Wertachse mit kurzen Teilstrichen skaliert wird.
Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**ValueAxisShowSmallIncrements**(i)[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

BOOLEAN

Optional. TRUE, wenn die Wertachse mit kurzen Teilstrichen skaliert wird.

Bemerkungen

Der Abstand zweier kurzer Teilstriche kann über die Eigenschaft "ValueAxisSmallIncrementSize(i)" verändert werden.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ValueAxisSmallIncrementSize(i) (RT Professional)

Beschreibung

Legt den Abstand zwischen zwei kurzen Teilstrichen der Skalierung fest. Ob die Angabe ausgewertet wird, ist abhängig von den Eigenschaften "ValueAxisShowSmallIncrements(i)" und "TimeAxisShowSmallIncrements(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.ValueAxisSmallIncrementSize(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit den Ausprägungen "OnlineTrendControl" oder "FunctionTrendControl".

Double

Optional. Ein Wert oder eine Konstante, welche den Abstand zwischen zwei kurzen Teilstrichen der Skalierung festlegt.

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

XAxisDecimalPrecision(i) (RT Professional)

Beschreibung

Legt die Anzahl der Nachkommastellen fest, mit denen der Skalierungswerte für die X-Achse angegeben wird.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisDecimalPrecision(i)[=Int]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

Int

Optional. Ein Wert oder eine Konstante, welche die Anzahl der Nachkommastellen festlegt, mit denen der Skalierungswerte für die X-Achse angegeben wird.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisGridLineInterval(i) (RT Professional)

Beschreibung

Legt den Abstand zweier Rasterlinien der X-Achse fest. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "XAxisShowGridLines(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisGridLineInterval(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

Double

Optional. Ein Wert oder eine Konstante, welche den Abstand zweier Rasterlinien der X-Achse festlegt.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisLargelIncrementSize(i) (RT Professional)

Beschreibung

Legt den Abstand zweier langer Teilstrichen der Skalierung der X-Achse fest. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "XAxisShowLargelIncrements(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisLargelIncrementSize(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

Double

Optional. Ein Wert oder eine Konstante, welche den Abstand zweier langer Teilstrichen der Skalierung der X-Achse festlegt.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisMode(i) (RT Professional)

Beschreibung

Legt fest, ob im Kurvenfenster für alle Kurven eine gemeinsame Zeitachse verwendet wird.

Specifies time units used by tag logging controls.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XAxisMode(i)[=TrendViewTimeAxisMode]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendControl".

TrendViewTimeAxisMode

(1): Die Zeitachse wird skaliert.

(2): Kein Zeitbereich ist festgelegt. Nur Stichproben werden angezeigt.

(3): Der Zeitbereich wird über Variablen festgelegt.

(4): Der Zeitbereich wird durch konstante Werte festgelegt.

Bemerkungen

Legt fest, ob im Kurvenfenster für alle Kurven eine gemeinsame X-Achse verwendet wird.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisShowLargeIncrements(i) (RT Professional)

Beschreibung

Legt fest, ob die X-Achse mit langen Teilstrichen skaliert wird. Der Abstand zweier langer Teilstriche kann über die Eigenschaft "XAxisLargeIncrementSize(i)" verändert werden

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisShowLargeIncrements(i)**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

BOOLEAN

Optional. TRUE, wenn die X-Achse mit langen Teilstrichen skaliert wird.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisShowSmallIncrements(i) (RT Professional)

Beschreibung

Legt fest, ob die X-Achse mit kurzen Teilstrichen skaliert wird. Der Abstand zweier kurzer Teilstriche kann über die Eigenschaft "XAxisSmallIncrementSize(i)" verändert werden.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisShowSmallIncrements(i)**[=BOOLEAN]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

BOOLEAN

Optional. TRUE, wenn die X-Achse mit kurzen Teilstrichen skaliert wird.

Siehe auch

FunctionTrendControl (Seite 1398)

XAxisSmallIncrementSize(i) (RT Professional)

Beschreibung

Legt den Abstand zweier kurzer Teilstriche der X-Achsen-Skalierung fest. Ob die Angabe ausgewertet wird, ist abhängig vom Wert der Eigenschaft "XAxisShowSmallIncrements(i)".

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XAxisSmallIncrementSize**(i)[=Double]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

Double

Optional. Ein Wert oder eine Konstante, welche den Abstand zweier kurzer Teilstriche der X-Achsen-Skalierung festlegt.

Siehe auch

FunctionTrendControl (Seite 1398)

XDataLogTag(i) (RT Professional)

Beschreibung

Legt die Variable fest, die entlang der X-Achse dargestellt wird. Die Eigenschaft "TagProviderType(i)" muss den Wert -1 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.**XDataLogTag**(i)[=HmiLoggingTag]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

HmiLoggingTag

Optional. Ein Wert oder eine Konstante, die die Variable festlegt, die entlang der X-Achse dargestellt wird.

Bemerkungen

Bei der Änderung von "XOnlineTag(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "XOnlineTag(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

XOnlineTag(i) (RT Professional)

Beschreibung

Legt die Variable fest, die entlang der X-Achse dargestellt wird. Die Eigenschaft "TagProviderType(i)" muss den Wert -1 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.XOnlineTag(i)[=HmiTag]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

HmiTag

Optional. Ein Wert oder eine Konstante, die die Variable festlegt, die entlang der X-Achse dargestellt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Bei der Änderung von "XOnlineTag(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "XOnlineTag(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

YDataLogTag(i) (RT Professional)

Beschreibung

Legt die Variable fest, die entlang der Y-Achse dargestellt wird. Die Eigenschaft "TagProviderType(i)" muss den Wert -1 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YDataLogTag(i)[=HmiLoggingTag]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

HmiLoggingTag

Optional. Ein Wert oder eine Konstante, die die Variable festlegt, die entlang der Y-Achse dargestellt wird.

Bemerkungen

Bei der Änderung von "YOnlineTag(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "YOnlineTag(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

YOnlineTag(i) (RT Professional)

Beschreibung

Legt die Variable fest, die entlang der Y-Achse dargestellt wird. Die Eigenschaft "TagProviderType(i)" muss den Wert -1 haben.

Zugriff in Runtime: Lesen und Schreiben

Syntax

Object.YOnlineTag(i)[=HmiTag]

Object

Erforderlich. Ein Objekt vom Typ "ScreenItem" mit der Ausprägung "FunctionTrendView".

HmiTag

Optional. Ein Wert oder eine Konstante, welche die Variable festlegt, die entlang der Y-Achse dargestellt wird.

Bemerkungen

Der Parameter i gibt die Nummer der Kurve an.

Bei der Änderung von "YOnlineTag(i)" können unzulässige Kombinationen mit anderen Attributen zur Datenanbindung entstehen. Deshalb muss vor einer Änderung von "YOnlineTag(i)" die sofortige Übernahme der Änderung mit "FreezeProviderConnections" verhindert werden.

Siehe auch

FunctionTrendControl (Seite 1398)

3.5.6 Methoden (Panels, Comfort Panels, RT Advanced, RT Professional)

3.5.6.1 Methoden A-G (Panels, Comfort Panels, RT Advanced, RT Professional)

Activate (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Aktiviert das Permanentfenster oder das Grundbild.

Zum Aktivieren eines nicht angewählten Bildes verwenden Sie die Eigenschaft "BaseScreenName".

Die Activate-Methode können Sie nur mit den folgenden bedienbaren Bildobjekten sinnvoll verwenden. Bei nicht bedienbaren Bildobjekten, z.B. Rechteck, wird eine Fehlermeldung ausgegeben.

- E/A-Feld
- Schalter
- Symbolbibliothek
- Kurvenanzeige
- f(x)-Kurvenanzeige
- HTML-Browser
- Schieberegler
- Grafisches E/A-Feld
- Symbolisches E/A-Feld

- Schaltfläche
- Meldeanzeige
- Benutzeranzeige
- Rezepturanzeige
- Sm@rtClient-Anzeige
- Status/Steuern

Syntax

Expression.Activate

Expression

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "Screen" oder "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

ScreenItem (Seite 1323)
Screen (Seite 1320)
ChannelDiagnose (Seite 1364)
CheckBox (Seite 1366)
Circle (Seite 1369)
CircleSegment (Seite 1372)
CircularArc (Seite 1374)
Clock (Seite 1377)
Connector (Seite 1379)
DateTimeField (Seite 1382)
DiskSpaceView (Seite 1384)
Ellipse (Seite 1386)
EllipseSegment (Seite 1388)
EllipticalArc (Seite 1391)
Gauge (Seite 1406)
GraphicIOField (Seite 1409)
GraphicView (Seite 1412)
HTMLBrowser (Seite 1416)

IOField (Seite 1418)
Rectangle (Seite 1471)
ScriptDiagnostics (Seite 1479)
Switch (Seite 1488)
SymbolicIOField (Seite 1491)
SymbolLibrary (Seite 1496)
TextField (Seite 1505)
TrendView (Seite 1516)
TubeArcObject (Seite 1520)
TubeDoubleTeeObject (Seite 1522)
TubePolyline (Seite 1524)
TubeTeeObject (Seite 1526)
UserView (Seite 1536)
WindowSlider (Seite 1538)
StatusForce (Seite 1486)
SmartClientView (Seite 1484)
Slider (Seite 1481)
ScreenWindow (Seite 1477)
RoundButton (Seite 1473)
Polyline (Seite 1459)
Polygon (Seite 1456)
OptionGroup (Seite 1452)
MultiLineEdit (Seite 1429)
MediaPlayer (Seite 1427)
Listbox (Seite 1424)
Line (Seite 1422)
Bar (Seite 1353)
Button (Seite 1360)
OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
TrendRulerControl (Seite 1508)
UserArchiveControl (Seite 1528)
FunctionTrendControl (Seite 1398)
AlarmView (Seite 1350)
AlarmControl (Seite 1338)

ActivateDynamic (RT Professional)

Beschreibung

Aktiviert dynamisch einen Trigger und den festgelegten Zyklus für eine Eigenschaft während Runtime. Dazu ist an der Eigenschaft ein VB-Skript erforderlich und ein Trigger, der auf "Auf Anforderung" gesetzt ist. Bei jeder Aktivierung des Triggers kann ein anderer Aktivierungszyklus verwendet werden.

Syntax

```
Ausdruck.ActivateDynamic (ByVal bstrPropertyName As String, ByVal  
bstrCycleName As String)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

Parameter	Beschreibung
bstrPropertyName	Name der Eigenschaft, auf die sich der Trigger bezieht.
bstrCycleName	Name des Aktivierungszyklus, z.B. "CycleTime1s".

Siehe auch

[ChannelDiagnose \(Seite 1364\)](#)
[CheckBox \(Seite 1366\)](#)
[Circle \(Seite 1369\)](#)
[CircleSegment \(Seite 1372\)](#)
[CircularArc \(Seite 1374\)](#)
[Clock \(Seite 1377\)](#)
[Connector \(Seite 1379\)](#)
[DiskSpaceView \(Seite 1384\)](#)
[Ellipse \(Seite 1386\)](#)
[EllipseSegment \(Seite 1388\)](#)
[EllipticalArc \(Seite 1391\)](#)
[Gauge \(Seite 1406\)](#)
[GraphicIOField \(Seite 1409\)](#)
[GraphicView \(Seite 1412\)](#)
[HTMLBrowser \(Seite 1416\)](#)

IOField (Seite 1418)
Rectangle (Seite 1471)
ScriptDiagnostics (Seite 1479)
SymbolicIOField (Seite 1491)
SymbolLibrary (Seite 1496)
TextField (Seite 1505)
TubeArcObject (Seite 1520)
TubeDoubleTeeObject (Seite 1522)
TubePolyline (Seite 1524)
TubeTeeObject (Seite 1526)
UserView (Seite 1536)
WindowSlider (Seite 1538)
Slider (Seite 1481)
ScreenWindow (Seite 1477)
RoundButton (Seite 1473)
Polyline (Seite 1459)
Polygon (Seite 1456)
OptionGroup (Seite 1452)
MultiLineEdit (Seite 1429)
MediaPlayer (Seite 1427)
Listbox (Seite 1424)
Line (Seite 1422)
Bar (Seite 1353)
Button (Seite 1360)
OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
TrendRulerControl (Seite 1508)
UserArchiveControl (Seite 1528)
FunctionTrendControl (Seite 1398)
AlarmControl (Seite 1338)

Add (RT Professional)

Beschreibung für TagSet-Objekt

Fügt eine Variable zur Auflistung hinzu. Die Variable kann über den Namen oder über eine Referenz zu einem Tag-Objekt hinzugefügt werden.

Syntax

```
Ausdruck.Add [Tag]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "TagSet" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
Tag	Name einer WinCC Variable oder Referenz auf ein Tag-Objekt, das der Auflistung hinzugefügt werden soll.

Beispiel

Im folgenden Beispiel wird ein TagSet-Objekt erzeugt und eine Variable hinzugefügt.

```
'VBS170  
Dim group  
Set group = HMIRuntime.Tags.CreateTagSet  
group.Add "Motor1"
```

Tag-Objekte können auch wie folgt aufgenommen werden.

```
'VBS171  
Dim Tag  
Set Tag = HMIRuntime.Tags("Motor2")  
Dim group2  
Set group2 = HMIRuntime.Tags.CreateTagSet  
group2.Add Tag
```

Beschreibung für DataSet-Objekt

Fügt der Auflistung einen Wert oder eine Objektreferenz hinzu.

Hinweis

Das DataSet-Objekt unterstützt keine Klassen.
Objekte vom Typ Screen, Screens, ScreenItem, ScreenItems, Tag und TagSet können nicht in die DataSet-Auflistung aufgenommen werden.
Bei Objektreferenzen muss sichergestellt sein, dass die Objekte multithreadfähig sind.

Syntax

```
Ausdruck.Add [vtName], [vtUserData]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "DataSet" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vtName	Name, unter dem der Wert oder die Variable zu der Auflistung hinzugefügt werden soll.
vtUserData	Wert, der in der Auflistung hinzugefügt werden soll.

Beispiel

In diesem Beispiel wird ein Wert in die DataSet-Auflistung aufgenommen.

```
'VBS172  
HMIRuntime.DataSet.Add "Motor1",23
```

Siehe auch

[DataSet \(Auflistung\) \(Seite 1311\)](#)
[OnlineTrendControl \(Seite 1443\)](#)

AttachDB-Methode (RT Professional)

Beschreibung

Führt die Tastenfunktion "Backup verbinden" des Control aus.

Syntax

```
Ausdruck.AttachDB()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[OnlineTableControl](#) (Seite 1434)

[FunctionTrendControl](#) (Seite 1398)

[AlarmControl](#) (Seite 1338)

CalculateStatistic (RT Professional)

Beschreibung

Führt die Tastenfunktion "Statistik berechnen" der f(t)-Kurvenanzeige und der Tabellenanzeige aus.

Syntax

```
Ausdruck.CalculateStatistic()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[OnlineTableControl](#) (Seite 1434)

CopyRows (RT Professional)

Beschreibung

Führt die Tastenfunktion "Zeilen kopieren" des Control aus.

Syntax

Ausdruck.CopyRows ()

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

UserArchiveControl (Seite 1528)

AlarmControl (Seite 1338)

Create (RT Professional)

Beschreibung

Erzeugt ein neues Alarm-Objekt.

Syntax

Ausdruck.Create (VARIANT vtApplication)

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "Alarm" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vtApplication	Name des Alarm-Objekts (optional)

Siehe auch

OnlineTrendControl (Seite 1443)

CreateTagSet (RT Professional)

Beschreibung

Erzeugt ein neues TagSet-Objekt. Dieses Objekt kann für optimierte Multi-Tag-Zugriffe benutzt werden.

Syntax

```
Ausdruck.CreateTagSet ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "TagSet" zurückgibt.

Parameter

VARIANT

Beispiel

Das folgende Beispiel zeigt wie man ein TagSet-Objekt erzeugt.

```
'VBS168  
'Build a Reference to the TagSet Object  
Dim group  
Set group = HMIRuntime.Tags.CreateTagSet
```

Siehe auch

OnlineTrendControl (Seite 1443)

CutRows (RT Professional)

Beschreibung

Führt die Tastenfunktion "Zeilen ausschneiden" der Rezepturanzeige aus.

Syntax

```
Ausdruck.CutRows ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

UserArchiveControl (Seite 1528)

DeactivateDynamic (RT Professional)

Beschreibung

Deaktiviert in/während Runtime den verwendeten Trigger der "ActivateDynamic"-Methode für die festgelegte Eigenschaft.

Syntax

```
Ausdruck.DeactivateDynamic(ByVal bstrPropertyName As String)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

String

Parameter	Beschreibung
bstrPropertyName	Name der Eigenschaft, auf die sich der Trigger bezieht.

Siehe auch

ChannelDiagnose (Seite 1364)

CheckBox (Seite 1366)

Circle (Seite 1369)

CircleSegment (Seite 1372)

CircularArc (Seite 1374)
Clock (Seite 1377)
Connector (Seite 1379)
DiskSpaceView (Seite 1384)
Ellipse (Seite 1386)
EllipseSegment (Seite 1388)
EllipticalArc (Seite 1391)
Gauge (Seite 1406)
GraphicIOField (Seite 1409)
GraphicView (Seite 1412)
HTMLBrowser (Seite 1416)
IOField (Seite 1418)
Rectangle (Seite 1471)
ScriptDiagnostics (Seite 1479)
SymbolicIOField (Seite 1491)
SymbolLibrary (Seite 1496)
TextField (Seite 1505)
TubeArcObject (Seite 1520)
TubeDoubleTeeObject (Seite 1522)
TubePolyline (Seite 1524)
TubeTeeObject (Seite 1526)
UserView (Seite 1536)
WindowSlider (Seite 1538)
Slider (Seite 1481)
ScreenWindow (Seite 1477)
RoundButton (Seite 1473)
Polyline (Seite 1459)
Polygon (Seite 1456)
OptionGroup (Seite 1452)
MultiLineEdit (Seite 1429)
MediaPlayer (Seite 1427)
Listbox (Seite 1424)
Line (Seite 1422)
Bar (Seite 1353)
Button (Seite 1360)

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
TrendRulerControl (Seite 1508)
FunctionTrendControl (Seite 1398)
AlarmControl (Seite 1338)

DeleteRows (RT Professional)

Beschreibung

Führt die Tastenfunktion "Zeilen löschen" der Rezepturanzeige aus.

Syntax

```
Ausdruck.DeleteRows ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)
UserArchiveControl (Seite 1528)

DetachDB (RT Professional)

Beschreibung

Führt die Tastenfunktion "Backup trennen" des Control aus.

Syntax

```
Ausdruck.DetachDB ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
FunctionTrendControl (Seite 1398)
AlarmControl (Seite 1338)

Edit (RT Professional)

Beschreibung

Führt die Tastenfunktion "Bearbeiten" der Tabellenanzeige aus.

Syntax

```
Ausdruck.Edit ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
AlarmControl (Seite 1338)

Export (RT Professional)

Beschreibung

Führt die Tastenfunktion "Archiv exportieren" oder "Daten exportieren" des Control aus.

Syntax

```
Ausdruck.Export ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Siehe auch

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
TrendRulerControl (Seite 1508)
UserArchiveControl (Seite 1528)
FunctionTrendControl (Seite 1398)
AlarmControl (Seite 1338)

GetColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Rezepturanteige als Typ "ICCAxUAColumn" zurück.

Syntax

```
Ausdruck.GetColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Spalte der Rezepturanzeige.

Beispiel

```
'VBS312  
Dim ctrl  
Dim objColumn  
Set ctrl = ScreenItems("RecipeControl")  
Set objColumn = ctrl.GetColumn("Field1")  
objColumn.Length = 30  
Set objColumn = ctrl.GetColumn(3)  
objColumn.Align = 2
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "Column" schreiben Sie z. B. "objColumn.Align" statt "objColumn.ColumnAlign".

Siehe auch

OnlineTrendControl (Seite 1443)

UserArchiveControl (Seite 1528)

GetColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Spalten-Objekte der Rezepturanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetColumnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'VBS313
Dim ctrl
Dim coll
Dim field
Set ctrl = ScreenItems("RecipeControl")
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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[UserArchiveControl \(Seite 1528\)](#)

GetHistlistColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Spalten-Objekte der Hitliste der Meldeanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetHitlisteColumnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[AlarmControl \(Seite 1338\)](#)

GetHitlistColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Hitliste der Meldeanzeige als Typ "ICCAxMessageColumn" zurück.

Syntax

```
Ausdruck.GetHitlistColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Spalte der Hitliste

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "HitlistColumn" schreiben Sie z. B. "objHitlistColumn.Visible" statt "objHitlistColumn.HitlistColumnVisible".

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[AlarmControl](#) (Seite 1338)

GetMessageBlock (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Meldeblock-Objekt der Meldeanzeige als Typ "ICCAxMessageBlock" zurück.

Syntax

```
Ausdruck.GetMessageBlock(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name des Meldeblocks.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "MessageBlock" schreiben Sie z. B. "objMsgBlock.Align" statt "objMsgBlock.MessageBlockAlign".

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[AlarmControl](#) (Seite 1338)

GetMessageBlockCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Meldeblock-Objekte der Meldeanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetMessageBlockCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "MessageBlock" schreiben Sie z. B. "msgblock.Align" statt "msgblock.MessageBlockAlign".

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[AlarmControl](#) (Seite 1338)

GetMessageColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Meldeanzeige als Typ "ICCAxMessageColumn" zurück.

Syntax

```
Ausdruck.GetMessageColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Spalte in der Meldeliste.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "MessageColumn" schreiben Sie z. B. "objMessColumn.Visible" statt "objMessColumn.MessageColumnVisible".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[AlarmControl \(Seite 1338\)](#)

GetOperatorMessage (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Bedienmeldungs-Objekt der Meldeanzeige als Typ "ICCAxOperatorMessage" zurück.

Syntax

Ausdruck.GetOperatorMessage(ByVal vIndex As Variant)

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Bedienmeldung

Beispiel

```
'VBS320
Dim ctrl
Dim objOpMess
Set ctrl = ScreenItems("AlarmControl")
Set objOpMess = ctrl.GetOperatorMessage(0)
objOpMess.Source1 = "Number"
objOpMess.SourceType1 = 1
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "OperatorMessage" schreiben Sie z. B. "objOpMess.Source1" statt "objOpMess.OperatorMessageSource1".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[AlarmControl \(Seite 1338\)](#)

GetMessageColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Spalten-Objekte der Meldeanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetMessageColumnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[AlarmControl \(Seite 1338\)](#)

GetOperatorMessageCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Bedienmeldungs-Objekte der Meldeanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetOperatorMessageCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

OnlineTrendControl (Seite 1443)

AlarmControl (Seite 1338)

GetRow (RT Professional)

Beschreibung

Gibt das mit der Zeilennummer bezeichnete Zeilen-Objekt der auf Tabellen basierenden Controls als Typ "ICCAxDataRow" zurück.

Syntax

```
Ausdruck.GetRow (ByVal IRow As Long)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

Long

Parameter	Beschreibung
IRow	Nummer der gewünschten Zeile des Controls.

Beispiel

```
'VBS356
Dim coll
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.GetRow(0).CellText(lCellIndex) & " "
    HMIRuntime.trace ctrl.GetRow(lIndex).CellText(lCellIndex) & " "
  Next
  HMIRuntime.trace vbNewLine
Next
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "Row" schreiben Sie z. B. "objRow.CellCount" statt "objRow.RowCellCount".

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

AlarmControl (Seite 1338)

GetRowCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Zeilen-Objekte der auf Tabellen basierenden Controls als Typ "ICCAxDataRowCollection" zurück.

Syntax

```
Ausdruck.GetRowCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften der ICCAxDataRowCollection

Die ICCAxDataRowCollection bezieht sich auf Runtime-Daten. Die Daten können nur gelesen werden. Ein Hinzufügen und Ändern der Daten ist nicht möglich.

Folgende Eigenschaften sind bei der ICCAxDataRowCollection verfügbar:

- Count - ermittelt die Anzahl der Zeilen in der Collection
- Item - Zugriff auf eine einzelne Zeile innerhalb der Collection über die Zeilennummer. Die Nummerierung läuft von 1 bis Count. Zurückgegeben wird ein Row-Objekt.

Beispiel

```
'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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[OnlineTableControl \(Seite 1434\)](#)

[TrendRulerControl \(Seite 1508\)](#)

[UserArchiveControl \(Seite 1528\)](#)

[AlarmControl \(Seite 1338\)](#)

GetRulerBlock (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Block-Objekt der Auswertetabelle als Typ "ICCAxRulerBlock" zurück.

Syntax

```
Ausdruck.GetRulerBlock(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name des Blocks in der Auswerteta- belle

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "RulerBlock" schreiben Sie z. B. "objRulerBlock.Caption" statt "objRulerBlock.BlockCaption".

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[TrendRulerControl](#) (Seite 1508)

GetRulerBlockCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Block-Objekte der Auswertetabelle als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetRulerBlockCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "RulerBlock" schreiben Sie z. B. "rulerblock.Align" statt "rulerblock.RulerBlockAlign".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[TrendRulerControl \(Seite 1508\)](#)

GetRulerColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Spalten-Objekt der Auswertetabelle als Typ "ICCAxRulerColumn" zurück.

Syntax

```
Ausdruck.GetRulerColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Spalte der Auswertetabelle.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "RulerColumn" schreiben Sie z. B. "objRulercol.Visible" statt "objRulercol.ColumnVisible".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[TrendRulerControl \(Seite 1508\)](#)

GetRulerColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Spalten-Objekte der Auswertetabelle als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetRulerColumnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

OnlineTrendControl (Seite 1443)

TrendRulerControl (Seite 1508)

GetRulerData (RT Professional)

Beschreibung

Gibt den Wert der aufgerufenen Kurve an der Linealposition zurück.

Syntax

```
Ausdruck.GetRulerData (ByVal RulerIndex As Long, pvValue As Variant,  
Optional pvTimeStamp As Variant, Optional pvFlags As Varian) Long
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "Trend" zurückgibt.

Parameter

Parameter	Beschreibung
RulerIndex	0 =Lineal
pvValue	Wert der X-Achse
pvTimeStamp	Zeitpunkt bzw. Wert der Y-Achse
pvFlags	Qualitycode

Beispiel

```
'VBS326  
Dim ctrl  
Dim objTrend  
Dim objIOField1  
Dim objIOField2  
Dim rulvalue  
Dim rultime  
Set ctrl = ScreenItems( "Controll1" )  
Set objTrend = ctrl.GetTrend( "Trend 1" )  
Set objIOField1 = ScreenItems( "I/O Field1" )  
Set objIOField2 = ScreenItems( "I/O Field2" )  
objTrend.GetRulerData 0, rulvalue, rultime  
objIOField1.OutputValue = rulvalue  
objIOField2.OutputValue = rultime
```

Siehe auch

OnlineTrendControl (Seite 1443)

GetSelectedRow (RT Professional)

Beschreibung

Gibt das markierte Zeilen-Objekt eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.

Syntax

```
Ausdruck.GetSelectedRow()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "Row" schreiben Sie z. B. "objRow.CellCount" statt "objRow.RowCellCount".

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

AlarmControl (Seite 1338)

GetSelectedRows (RT Professional)

Beschreibung

Gibt bei Mehrfachselektion die markierten Zeilen-Objekte eines tabellenbasierenden Controls als Typ "ICCAxDataRow" zurück.

Syntax

```
Ausdruck.GetSelectedRows ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "Row" schreiben Sie z. B. "objRow.CellCount" statt "objRow.RowCellCount".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[OnlineTableControl \(Seite 1434\)](#)

[TrendRulerControl \(Seite 1508\)](#)

[UserArchiveControl \(Seite 1528\)](#)

[AlarmControl \(Seite 1338\)](#)

GetStatisticAreaColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Spalten-Objekt des Statistikbereichfensters der Auswertetabelle als Typ "ICCAxRulerColumn" zurück.

Syntax

```
Ausdruck.GetStatisticAreaColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Spalte des Statistikbereichfensters.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "StatisticAreaColumn" schreiben Sie z. B. "objStatAreaCol.Visible" statt "objStatAreaCol.ColumnVisible".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[TrendRulerControl \(Seite 1508\)](#)

GetStatisticAreaColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Spalten-Objekte des Statistikbereichfensters der Auswertetabelle als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetStatisticAreaColumnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

OnlineTrendControl (Seite 1443)

TrendRulerControl (Seite 1508)

GetStatisticResultColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Spalten-Objekt des Statistikfensters der Auswertetabelle als Typ "ICCAxRulerColumn" zurück.

Syntax

```
Ausdruck.GetStatisticResultColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Spalte des Statistikfensters.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "StatisticResultColumn" schreiben Sie z. B. "objStatResCol.Visible" statt "objStatResCol.ColumnVisible".

Siehe auch

OnlineTrendControl (Seite 1443)

TrendRulerControl (Seite 1508)

GetStatisticResultColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Spalten-Objekte des Statistikfensters der Auswertetabelle als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetStatisticResultColumnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[TrendRulerControl \(Seite 1508\)](#)

GetStatusBarElement (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Element der Statuszeile des Controls als Typ "ICCAxStatusBarElement" zurück.

Syntax

```
Ausdruck.GetStatusBarElement(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name des Elements der Statuszeile.

Beispiel

```
'VBS331
Dim ctrl
Dim objStatusBar
Set ctrl = ScreenItems("Controll1")
Set objStatusBar = ctrl.GetStatusBarElement(1)
objStatusBar.Visible = FALSE
Set objStatusBar = ctrl.GetStatusBarElement(3)
objStatusBar.Width = 10
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "StatusBarElement" schreiben Sie z. B. "objStatusBar.Visible" statt "objStatusBar.StatusBarElementVisible".

Siehe auch

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
TrendRulerControl (Seite 1508)
UserArchiveControl (Seite 1528)
FunctionTrendControl (Seite 1398)
AlarmControl (Seite 1338)

GetStatusBarElementCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Elemente der Statuszeile des Controls als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetStatusBarElementCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'VBS332
Dim ctrl
Dim coll
Dim statelement
Set ctrl = ScreenItems.Item("Controll")
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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "StatusBarElement" schreiben Sie z. B. "statelement.Name" statt "statelement.StatusBarElementName".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[OnlineTableControl \(Seite 1434\)](#)

[TrendRulerControl \(Seite 1508\)](#)

[UserArchiveControl \(Seite 1528\)](#)

FunctionTrendControl (Seite 1398)

AlarmControl (Seite 1338)

GetTimeAxis (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Zeitachsen-Objekt der f(t)-Kurvenanzeige als Typ "ICCAxTimeAxis" zurück.

Syntax

```
Ausdruck.GetTimeAxis (ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Zeitachse.

Beispiel

```
'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
'The format to be used for date and time entries depends on the
'regional settings and language options in the operating system.
objTimeAxis.BeginTime = "06.04.2010 9:33:18"
'objTimeAxis.BeginTime = "04/06/2010 9:33:18"
objTimeAxis.MeasurePoints = 100
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "TimeAxis" schreiben Sie z. B. "objTimeAx.Visible" statt "objTimeAx.TimeAxisVisible".

Siehe auch

OnlineTrendControl (Seite 1443)

GetTimeAxisCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Zeitachsen-Objekte der f(t)-Kurvenanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetTimeAxisCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
'The format to be used for date and time entries depends on the
'regional settings and language options in the operating system.
objTimeAxis1.BeginTime = "01.01.2010 0:00:00"
'objTimeAxis1.BeginTime = "01/01/2010 0:00:00"
objTimeAxis1.EndTime = "31.12.2010 11:59:59"
'objTimeAxis1.EndTime = "12/31/2010 11:59:59"
objTimeAxis2.TrendWindow = objTrendWnd.Name
objTimeAxis2.Label = "2011"
objTimeAxis2.RangeType = 1
objTimeAxis2.BeginTime = "01.01.2011 0:00:00"
'objTimeAxis2.BeginTime = "01/01/2011 0:00:00"
objTimeAxis2.EndTime = "31.12.2011 11:59:59"
'objTimeAxis2.EndTime = "12/31/2011 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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "TimeAxis" schreiben Sie z. B. "objTimeAxis1.Label" statt "objTimeAxis1.TimeAxisLabel".

Siehe auch

OnlineTrendControl (Seite 1443)

GetTimeColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Zeitspalten-Objekt der Tabellenanzeige als Typ "ICCAxTimeColumn" zurück.

Syntax

```
Ausdruck.GetTimeColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Zeitspalte.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "TimeColumn" schreiben Sie z. B. "objTimeColumn.ShowDate" statt "objTimeColumn.TimeColumnShowDate".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[OnlineTableControl \(Seite 1434\)](#)

GetTimeColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Zeitspalten-Objekte der Tabellenanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetTimeColumnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
'The format to be used for date and time entries depends on the
'regional settings and language options in the operating system.
objTimeCol1.BeginTime = "01.01.2010 0:00:00"
'objTimeCol1.BeginTime = "01/01/2010 0:00:00"
objTimeCol1.EndTime = "31.12.2010 11:59:59"
'objTimeCol1.EndTime = "12/31/2010 11:59:59"
objTimeCol2.Caption = "2011"
objTimeCol2.RangeType = 0
objTimeCol2.BeginTime = "01.01.2011 0:00:00"
'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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[OnlineTableControl \(Seite 1434\)](#)

GetToolBarButton (RT Professional)

Beschreibung

Gibt die mit Namen oder Index bezeichnete Tastenfunktion der Symbolleiste des Controls als Typ "ICCAxToolBarButton" zurück.

Syntax

```
Ausdruck.GetToolBarButton(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Tastenfunktion der Symbolleiste.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "ToolBarButton" schreiben Sie z. B. "toolbu.Index" statt "toolbu.ToolBarButtonIndex".

Siehe auch

- OnlineTrendControl (Seite 1443)
- OnlineTableControl (Seite 1434)
- TrendRulerControl (Seite 1508)
- UserArchiveControl (Seite 1528)
- FunctionTrendControl (Seite 1398)
- AlarmControl (Seite 1338)

GetToolBarButtonCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Tastenfunktionen der Symbolleiste des Controls als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetToolBarButtonCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Methoden sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

- OnlineTrendControl (Seite 1443)
- OnlineTableControl (Seite 1434)
- TrendRulerControl (Seite 1508)
- UserArchiveControl (Seite 1528)
- FunctionTrendControl (Seite 1398)
- AlarmControl (Seite 1338)

GetTrend (RT Professional)

Beschreibung

Gibt die mit Namen oder Index bezeichnete Kurve der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxTrend" bzw. "ICCAxFunctionTrend" zurück.

Syntax

```
Ausdruck.GetTrend(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Kurve.

Beispiel für Runtime Professional

```
'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"
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "Trend" schreiben Sie z. B. "objTrend.PointStyle" statt "objTrend.TrendPointStyle".

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

GetTrendCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Kurven der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetTrendCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel für Runtime Professional

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "Trend" schreiben Sie z. B. "objTrend.TagName" statt "objTrend.TrendTagName".

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[FunctionTrendControl](#) (Seite 1398)

GetTrendWindow (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Kurvenfenster-Objekt der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxTrendWindow" zurück.

Syntax

```
Ausdruck.GetTrendWindow(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name des Kurvenfensters.

Beispiel für Runtime Professional

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "TrendWindow" schreiben Sie z. B. "objTrendWnd.Visible" statt "objTrendWnd.TrendWindowVisible".

Siehe auch

[OnlineTrendControl](#) (Seite 1443)

[FunctionTrendControl](#) (Seite 1398)

GetTrendWindowCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Kurvenfenster-Objekte der f(t)-Kurvenanzeige oder der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetTrendWindowCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel für Runtime Professional

```
'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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[FunctionTrendControl \(Seite 1398\)](#)

GetValueAxis (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Wertachsen-Objekt der f(t)-Kurvenanzeige als Typ "ICCAxValueAxis" zurück.

Syntax

Ausdruck.GetValueAxis(ByVal vIndex As Variant)

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Werteachse.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "ValueAxis" schreiben Sie z. B. "objValueAx.Visible" statt "objValueAx.ValueAxisVisible".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

GetValueAxisCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Wertachsen-Objekte der f(t)-Kurvenanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetValueAxisCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "ValueAxis" schreiben Sie z. B. "objValueAxis1.Label" statt "objValueAxis1.ValueAxisLabel".

Siehe auch

OnlineTrendControl (Seite 1443)

GetValueColumn (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Wertspalten-Objekt der Tabellenanzeige als Typ "ICCAxValueColumn" zurück.

Syntax

```
Ausdruck.GetValueColumn(ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Wertspalte der f(t)-Kurvenanzeige.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "ValueColumn" schreiben Sie z. B. "objValueColumn.Precisions" statt "objValueColumn.ValueColumnPrecisions".

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

GetValueColumnCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Wertspalten-Objekte der f(t)-Kurvenanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetValueColulmnCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[OnlineTableControl \(Seite 1434\)](#)

GetXAxis (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete X-Achsen-Objekt der f(x)-Kurvenanzeige als Typ "ICCAxValueAxis" zurück.

Syntax

```
Ausdruck.GetXAxis (ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der X-Achse.

Beispiel

```
'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)
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "XAxis" schreiben Sie z. B. "objXAx.Visible" statt "objXAx.XAxisVisible".

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

GetXAxisCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller X-Achsen-Objekte der f(x)-Kurvenanzeige als Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetXAxisCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "XAxis" schreiben Sie z. B. "objXAxis1.Label" statt "objXAxis1.XAxisLabel".

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[FunctionTrendControl \(Seite 1398\)](#)

GetYAxis (RT Professional)

Beschreibung

Gibt das mit Namen oder Index bezeichnete Y-Achsen-Objekt der f(x)-Kurvenanzeige als Typ "ICCAxValueAxis" zurück.

Syntax

```
Ausdruck.GetYAxis (ByVal vIndex As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
vIndex	Index oder Name der Y-Achse.

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "YAxis" schreiben Sie z. B. "objYAx.Visible" statt "objYAx.YAxisVisible".

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

GetYAxisCollection (RT Professional)

Beschreibung

Gibt die Auflistung aller Y-Achsen-Objekte der f(x)-Kurvenanzeige vom Typ "ICCAxCollection" zurück.

Syntax

```
Ausdruck.GetYAxisCollection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Eigenschaften und Funktionen der ICCAxCollection

Folgende Eigenschaften sind bei der ICCAxCollection verfügbar:

- Count
- Item

Folgende Funktionen sind bei der ICCAxCollection verfügbar:

- AddItem(vName) As Object
- RemoveItem(vIndex)

Beispiel

```
'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
```

Hinweis

Wenn Sie über die "Get..."-Methoden auf Eigenschaften über die Auflistung des Control-Objekts und nicht über das Control-Objekt zugreifen, müssen Sie das Präfix der Eigenschaft mit dem Namen der Auflistung weglassen.

Bei der Auflistung "YAxis" schreiben Sie z. B. "objYAxis1.Label" statt "objYAxis1.YAxisLabel".

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

3.5.6.2 Methoden H-R (Panels, Comfort Panels, RT Advanced, RT Professional)

HideAlarm (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Meldung ausblenden" der Meldeanzeige aus.

Syntax

```
Ausdruck.HideAlarm()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

InsertData (RT Professional)

Beschreibung

Fügt Daten zur aufgerufenen Kurve hinzu.

Syntax

```
Ausdruck.InsertData(dblAxisX As Variant, dblAxisY As Variant)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "Trend" zurückgibt.

Parameter

Parameter	Beschreibung
dblAxisX	Wert der X-Achse
dblAxisY	Wert der Y-Achse

Beispiel

```
'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 = CDbI(lngFactor * 0.02)
dblAxisY = CDbI(dblAxisX * dblAxisX + 2 * dblAxisX + 1)
objTrend.InsertData dblAxisX, dblAxisY
Next
```

Item (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt ein Element aus einer Auflistung zurück.

Syntax

Expression.Item(Index)

Expression

Erforderlich. Ein Ausdruck, der eine Auflistung zurückgibt.

Parameter

Index

Der Name oder die Indexnummer eines Elements der Auflistung:

- ScreenItems-Auflistung: Verwenden Sie den Objektnamen, z.B. "HmiRuntime.Screens(1).ScreenItems("Circle")", oder die Indexnummer.
- Screens-Auflistung: Verwenden Sie entweder den Namen oder die Indexnummer.
- SmartTags-Auflistung: Als Index können Sie in der SmartTags-Auflistung nur den Variablennamen verwenden. Eine Aufzählung aller Variablen ist daher nicht möglich.

Wenn der übergebene Wert mit keinem Element in der Auflistung übereinstimmt, tritt ein Fehler auf. Der Rückgabewert wird mit dem Wert "Nothing" belegt.

```
On Error Resume Next
Dim screen
Set screen = HmiRuntime.Screens("Screen_1")
If (screen is Nothing)
then...
Else...
End If
```

Um eine optimale Unterstützung per Autovervollständigung zu haben, verwenden Sie am Besten eine kombinierte Adressierung über Bildname und Objektname, .z. B.
`"HmiRuntime.Screens("Screen").ScreenItems("Circle")"`.

Beispiel

Die Item-Methode ist die Standardmethode für Auflistungen. Deshalb liefern die beiden folgenden Beispiele das gleiche Ergebnis:

```
'VBS_Example_Item
HMIRuntime.Screens.Item(1)
HMIRuntime.Screens(1)
```

Beide Anweisungen referenzieren jeweils das Grundbild.

Siehe auch

[ScreenItems \(Auflistung\) \(Seite 1325\)](#)

[ScreenItem \(Seite 1323\)](#)

LockAlarm (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Meldung sperren" der Meldeanzeige aus.

Syntax

```
Ausdruck.LockAlarm()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

LoopInAlarm (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Loop in Alarm" der Meldeanzeige aus.

Syntax

```
Ausdruck.LoopInAlarm()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveAxis (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Achsenbereich verschieben" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.MoveAxis()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)
FunctionTrendControl (Seite 1398)

MoveToFirst (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Erste Zeile" des Controls aus.

Syntax

```
Ausdruck.MoveToFirst()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
UserArchiveControl (Seite 1528)

MoveToFirstLine (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Erste Meldung" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToFirstLine()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveToFirstPage (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Erste Seite" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToFirstPage()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveToLast (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Letzter Datensatz" des Controls aus.

Syntax

```
Ausdruck.MoveToLast()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
UserArchiveControl (Seite 1528)

MoveToLastLine (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Letzte Meldung" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToLastLine()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveToLastPage (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Letzte Seite" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToLastPage()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveToNext (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Nächster Datensatz" des Controls aus.

Syntax

```
Ausdruck.MoveToNext ( )
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

UserArchiveControl (Seite 1528)

MoveToNextLine (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Nächste Meldung" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToNextLine ( )
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveToNextPage (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Nächste Seite" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToNextPage ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveToPrevious (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Vorhergehender Datensatz" des Controls aus.

Syntax

```
Ausdruck.MoveToPrevious ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

UserArchiveControl (Seite 1528)

MoveToPreviousLine (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Vorhergehende Meldung" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToPreviousLine()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

MoveToPreviousPage (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Vorhergehende Seite" der Meldeanzeige aus.

Syntax

```
Ausdruck.MoveToPreviousPage()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

NextColumn (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Nächste Spalte" der Tabellenanzeige aus.

Syntax

```
Ausdruck.NextColumn()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTableControl (Seite 1434)

NextTrend (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Nächste Kurve" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.NextTrend()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

OneToOneView (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Originalansicht" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.OneToOneView()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

PasteRows (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Zeilen einfügen" der Rezepturanzeige aus.

Syntax

```
Ausdruck.PasteRows()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

PreviousColumn (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Vorhergehende Spalte" der Tabellenanzeige aus.

Syntax

```
Ausdruck.PreviousColumn()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTableControl (Seite 1434)

PreviousTrend (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Vorhergehende Kurve" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.PreviousTrend()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

Print (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Drucken" des Controls aus.

Syntax

```
Ausdruck.Print()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

FunctionTrendControl (Seite 1398)

AlarmControl (Seite 1338)

QuitHorn (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Quittierung zentraler Melder" der Meldeanzeige aus.

Syntax

```
Ausdruck.QuitHorn()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

QuitSelected (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Einzelquittierung" der Meldeanzeige aus.

Syntax

```
Ausdruck.QuitSelected()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

QuitVisible (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Sammelquittierung" der Meldeanzeige aus.

Syntax

```
Ausdruck.QuitVisible()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ReadTags (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Variablen lesen" der Rezepturanzeige aus.

Syntax

```
Ausdruck.ReadTags ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

Read (RT Professional)

Beschreibung für Tag-Objekt

Liest den Zustand einer Variablen (Tag-Objekt) kurz nach dem Aufrufzeitpunkt aus. Gleichzeitig wird das Tag-Objekt mit den gelesenen Werten versorgt. Beim Lesen der Variablen wird der Wert, der Quality Code und der Zeitstempel der Variablen ermittelt. Über die Eigenschaft "LastError" kann ermittelt werden, ob der Aufruf erfolgreich war.

Die Eigenschaften "Name" und "Tagprefix" werden dadurch nicht verändert.

Wird der Wert der Variablen erfolgreich gelesen, werden die Eigenschaften des Tag-Objektes mit folgenden Werten belegt:

Eigenschaft	Belegung
Value	Wert der Variablen
Name	Variablenname (unverändert)
QualityCode	Qualitätseinstufung
TimeStamp	aktueller Zeitstempel der Variablen
LastError	0
ErrorDescription	" "

Wird der Variablenwert nicht erfolgreich gelesen, werden die Eigenschaften des Tag-Objektes mit folgenden Werten belegt:

Eigenschaft	Belegung
Value	VT_Empty
Name	Variablenname (unverändert)
QualityCode	Bad Out of Service
TimeStamp	0
LastError	Fehlercode der Leseoperation
ErrorDescription	Fehlerbeschreibung zu LastError

Syntax

```
Ausdruck.Read ([Readmode])
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Tag-Objekt zurückgibt. Der Rückgabewert der Read-Methode ist der Wert der ausgelesenen Variablen.

Parameter

Durch den optionalen Parameter "Readmode" wird zwischen zwei Lesearten unterschieden:

Parameter	Beschreibung
0	Der Wert der Variablen wird aus dem Prozessabbild gelesen (cache). 0 ist der Defaultwert.
1	Der Wert der Variablen wird direkt aus dem AS oder dem Kanal gelesen (direct).

Wird der Parameter "Readmode" weggelassen, wird standardmäßig aus dem Prozessabbild gelesen. Der Rückgabewert der Read-Methode ist der ausgelesene Variablenwert als VARIANT.

Lesen aus dem Prozessabbild

Beim Lesen aus dem Prozessabbild wird die Variable angemeldet und von da an zyklisch aus dem Automatisierungssystem angefordert. Der Zyklus der Anmeldung ist dabei vom projektierten Trigger abhängig. Der Wert wird aus dem Variablenabbild von WinCC gelesen. Bei Bildabwahl werden die Variablen wieder abgemeldet. Der Aufruf zeichnet sich durch Folgendes aus:

- Der Wert wird aus dem Variablenabbild von WinCC gelesen
- Der Aufruf ist im Vergleich zum direkten Lesen schneller (Außer beim ersten Aufruf: Der erste Aufruf dauert prinzipiell länger, da der Wert aus dem AS gelesen und angemeldet werden muss.)
- Die Dauer des Aufrufes ist nicht von Buslast oder vom AS abhängig

Verhalten in Aktionen mit Variablentrigger

Alle im Variablentrigger enthaltenen Variablen sind bereits bei Bildanwahl bekannt und werden mit der angegebenen Überwachungszeit angemeldet. Da alle Variablen auf einmal angefordert werden, kann vom Kanal die bestmögliche Optimierung erzielt werden. Wird innerhalb einer Aktion eine Variable mit Read angefordert, die im Trigger enthalten ist, liegt der Wert bereits vor und wird dem Aufruf direkt übergeben. Wird eine Variable angefordert, die nicht im Trigger enthalten ist, ist das Verhalten wie beim Standardtrigger.

Verhalten in Aktionen mit zyklischem Trigger

Beim ersten Aufruf wird die Variable mit der halben Zykluszeit angemeldet. Bei jedem weiteren Aufruf liegt dann der Wert vor.

Verhalten in ereignisgetriggerten Aktionen

Beim ersten Aufruf wird die Variable im Modus "bei Änderung" angemeldet. Prozessvariablen, die im Modus "bei Änderung" angemeldet sind, entsprechen einem zyklischen Leseauftrag mit einer Zykluszeit von 1s.

Wenn mit einem Ereignis (z.B. Mausklick) ein Wert asynchron angefordert wird, wird die Variable in das Variablenabbild aufgenommen. Die Variable wird ab diesem Zeitpunkt zyklisch aus dem AS angefordert und erhöht somit die Grundlast. Um diese Erhöhung der Grundlast zu umgehen, können Sie den Wert auch synchron lesen. Der synchrone Aufruf verursacht zwar einmalig eine höhere Kommunikationslast, aber die Variable wird nicht in das Variablenabbild aufgenommen.

Direktes Lesen

Beim direkten Lesen wird der aktuelle Wert zurückgeliefert. Die Variable wird nicht zyklisch angemeldet, sondern der Wert wird einmalig aus dem AS angefordert. Direktes Lesen hat folgende Eigenschaften:

- Der Wert wird explizit aus dem AS gelesen
- Der Aufruf dauert im Vergleich zum Lesen aus dem Prozessabbild länger
- Die Dauer des Aufrufes ist u.a. von der Buslast und vom AS abhängig

Beispiel

Lesen einer Variablen direkt aus dem AS oder dem Kanal

```
'VBS100
Dim objTag
Dim vntValue
Set objTag = HMIRuntime.Tags("Tagname")
vntValue = objTag.Read(1)      'Read direct
MsgBox vntValue
```

Lesen einer Variablen aus dem Prozessabbild

```
'VBS101
Dim objTag
Dim vntValue
Set objTag = HMIRuntime.Tags("Tagname")
vntValue = objTag.Read      'Read from cache
MsgBox vntValue
```

Beschreibung für TagSet-Objekt

Das TagSet-Objekt bietet die Möglichkeit, mehrere Variablen mit einem Aufruf zu lesen.

Die Funktionsweise ist dabei weitgehend mit der eines Tag-Objektes identisch. Nachfolgend werden nur die Abweichungen beschrieben.

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "TagSet" zurückgibt.

Lesen aus dem Prozessabbild

Das TagSet-Objekt bietet den Vorteil, dass es mit einem Leseauftrag mehrere Variablen anfordern kann. Dabei werden die Variablen gesammelt im Prozessabbild angemeldet und somit die Performance verbessert.

Direktes Lesen

Da mit einem Aufruf mehrere Leseaufträge abgearbeitet werden können, wird die Performance gegenüber mehreren Einzelaufrufen verbessert.

Beispiel

Das folgende Beispiel zeigt wie Variablen in die TagSet-Auflistung aufgenommen und danach gelesen werden.

```
'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
```

Setzt man den optionalen Parameter "Readmode" auf 1, so werden Prozess-Variablen nicht angemeldet sondern direkt aus dem AS oder dem Kanal gelesen.

```
group.Read 1
```

Refresh (RT Professional)

Beschreibung

Zeichnet alle sichtbaren Bilder neu.

Syntax

```
Ausdruck.Refresh
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "Screens" oder "Screen" zurückgibt.

Parameter

--

Beispiele

Das erste Beispiel erzwingt ein sofortiges Neuzeichnen aller sichtbaren Bilder:

```
'VBS149  
HMIRuntime.Screens.Refresh
```

Das zweite Beispiel erzwingt ein sofortiges Neuzeichnen des Grundbildes:

```
'VBS150  
HMIRuntime.Screens(1).Refresh
```

Siehe auch

Screen (Seite 1320)

HMIRuntime (Seite 1313)

Remove (RT Professional)

Beschreibung für TagSet-Objekt

Entfernt eine Variable aus einer TagSet-Auflistung. Die Variable kann über den Namen oder über eine Referenz auf ein Tag-Objekt entfernt werden.

Syntax

```
Ausdruck.Remove [Tag]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "TagSet" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
Tag	Name einer WinCC Variable oder Referenz auf ein Tag-Objekt, das aus der Auflistung entfernt werden soll.

Beispiel

Das folgende Beispiel zeigt, wie mehrere Variablen in die TagSet-Auflistung aufgenommen werden und eine Variable wieder entfernt wird.

```
'VBS175  
Dim group  
Set group = HMIRuntime.Tags.CreateTagSet  
group.Add "Motor1"  
group.Add "Motor2"  
group.Remove "Motor1"
```

Beschreibung für DataSet-Objekt

Löscht das im Parameter "Name" angegebene Element aus einer Auflistung.

Syntax

```
Ausdruck.Remove [Name]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "DataSet" zurückgibt.

Parameter

VARIANT

Parameter	Beschreibung
Name	Name des Objekts, das aus der Auflistung entfernt werden soll.

Beispiel

Das Beispiel zeigt, wie das Objekt "motor1" aus der Auflistung entfernt wird.

```
'VBS166  
HMIRuntime.DataSet.Remove("motor1")
```

Beschreibung für Objekte Logging, AlarmLogs, DataLogs

Die Methode löscht zuvor eingelagerte Archivsegmente aus dem Runtime-Projekt.

Mit der Methode "Remove" gelöschte Archivsegmente werden aus dem Ordner "Common Archiving" des Projekts entfernt.

Der Aufruf kann in Abhängigkeit von den Archivdaten einen längeren Zeitraum in Anspruch nehmen. Dies kann die Abarbeitung der nachfolgenden Skripte blockieren. Eine Blockade der Aktionen im Bild können Sie vermeiden, indem Sie den Aufruf in einer Aktion im Global Scripting starten, zum Beispiel durch das Starten der Aktion durch eine Triggervariable.

Durch das Trennen / Löschen der Archive wird CPU-Last erzeugt. Dies wirkt sich auf die Performance aus.

Hinweis

Der Aufruf der Methode "Remove" ist derzeit nur am Server möglich. Es existiert aber ein Beispiel, das zeigt, wie die Methode vom Client aus auf dem Server gestartet werden kann.

Syntax

Objekte Logging, AlarmLogs

```
Ausdruck.Remove [TimeFrom] [TimeTo] [TimeOut] [ServerPrefix]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "Logging" oder "AlarmLogs" zurückgibt.

Objekt DataLogs

```
Ausdruck.Remove [TimeFrom] [TimeTo] [TimeOut] [Type] [ServerPrefix]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "DataLogs" zurückgibt.

Parameter

TimeFrom

Zeitpunkt, von dem an die Archive gelöscht werden sollen.

Bei der Angabe des Zeitpunktes ist auch eine Kurzform möglich, wie im Absatz "Format der Zeitangaben" beschrieben.

TimeTo

Zeitpunkt, bis zu dem die Archive Segmente gelöscht werden sollen.

Bei der Angabe des Zeitpunktes ist auch eine Kurzform möglich, wie im Absatz "Format der Zeitangaben" beschrieben.

Timeout

Timeout in Millisekunden.

Wenn Sie als Wert "-1" eingeben, wird für immer gewartet (infinite). Wenn Sie den Wert "0" eingeben, wird nicht gewartet.

Type

Typ des Archivs.

Dieser Parameter kann (optional) nur für das Löschen von Archivsegmenten des Tag Logging verwendet werden.

Folgende Werte können eingegeben werden:

zugeordneter Wert	Typ	Beschreibung
1	hmiDataLogFast	Tag Logging Fast data
2	hmiDataLogSlow	Tag Logging Slow data
3	hmiDataLogAll	Tag Logging Fast and Slow data

ServerPrefix

Reserviert für zukünftige Versionen.

Rückgabewert

Wenn beim Löschen der Archive Segmente ein Fehler aufgetreten ist, gibt die Methode eine Fehlermeldung zurück. Weitere Informationen finden Sie unter dem Thema "Fehlermeldungen aus dem Bereich Datenbanken".

Format der Zeitangaben

Das Format für die Angabe von Zeitpunkten ist wie folgt definiert: YYYY-MM-DD hh:mm:ss, wobei YYYY das Jahr bezeichnet, MM den Monat, DD den Tag, hh die Stunde, mm die Minute und ss die Sekunde. Beispielsweise wird die Uhrzeit 2 Minuten und eine Sekunde nach 11 Uhr am 26. Juli 2004 so dargestellt: 2004-07-26 11:02:01.

Bei den Parametern "TimeFrom" und "TimeTo" ist die Angabe von Datum und Uhrzeit in einer Kurzform möglich. Dabei sind nicht alle Felder des Formats zu belegen. Die Kurzform bedeutet, dass bei der Datum/Zeit-Angabe ein oder mehrere Parameter, mit dem Sekundenwert beginnend, entfallen können. So kann die Angabe z. B. in den Formaten "YYYY-MM" oder "YYYY-MM-DD hh" erfolgen. Mit der Angabe "TimeFrom" = "2004-09" und "TimeTo" = "2004-10-04" werden alle Archivsegmente von September 2004 bis einschließlich 4. Oktober eingelagert.

Beispiel

Im folgenden Beispiel werden nachträglich (wieder) eingelagerte Archivsegmente eines bestimmten Zeitraumes entfernt und der Rückgabewert als Trace ausgegeben.

```
'VBS182
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Remove("2004-08-22","2004-09-22",-1) &
vbNewLine
```

Im folgenden Beispiel werden alle nachträglich (wieder) eingelagerten Archivsegmente entfernt und der Rückgabewert als Trace ausgegeben.

```
'VBS183
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Remove("", "", -1) & vbNewLine
```

Siehe auch

- Logging (Seite 1318)
- DataSet (Auflistung) (Seite 1311)
- DataLogs (Auflistung) (Seite 1309)
- AlarmLogs (Auflistung) (Seite 1306)

RemoveAll (RT Professional)

Beschreibung für TagSet-Objekt

Löscht alle Variablen aus einer TagSet-Auflistung.

Syntax

```
Ausdruck.RemoveAll
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "TagSet" zurückgibt.

Parameter

--

Beispiel

Das folgende Beispiel zeigt, wie mehrere Variablen in die TagSet-Auflistung aufgenommen und alle wieder entfernt werden.

```
'VBS176
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Motor1"
group.Add "Motor2"
group.RemoveAll
```

Beschreibung für DataSet-Objekt

Löscht alle Werte oder Objektreferenzen aus einer DataSet-Auflistung.

Syntax

```
Ausdruck.RemoveAll
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "DataSet" zurückgibt.

Parameter

--

Beispiel

Das Beispiel zeigt, wie alle Objekte aus der Auflistung entfernt werden.

```
'VBS167
HMIRuntime.DataSet.RemoveAll
```

Siehe auch

[DataSet \(Auflistung\) \(Seite 1311\)](#)

Restore (RT Professional)

Beschreibung für Objekte Logging, AlarmLogs, DataLogs

Die Methode fügt ausgelagerte Archivsegmente dem Runtime-Projekt hinzu.

Beim Einlagern der Archivsegmente werden diese in den Ordner "Common Archiving" des Projekts kopiert. Daher muss entsprechend Speicherkapazität vorhanden sein.

Der Aufruf kann in Abhängigkeit von den Archivdaten einen längeren Zeitraum in Anspruch nehmen. Dies kann die Abarbeitung der nachfolgenden Skripte blockieren. Eine Blockade der Aktionen im Bild können Sie vermeiden, indem Sie den Aufruf in einer Aktion im Global Scripting starten, zum Beispiel durch das Starten der Aktion durch eine Triggervariable.

Durch das Verbinden / Kopieren der Archive entsteht CPU-Last weil insbesondere bei eingeschalteter Signaturprüfung der SQL-Server mehr belastet wird. Durch das Kopieren der Archivsegmente wird der Festplattenzugriff verlangsamt.

Bei eingeschalteter Signaturprüfung wird eine Fehlermeldung zurückgegeben, wenn ein nicht signiertes oder verändertes Archiv eingelagert werden soll. Es wird immer nur eine Fehlermeldung zurückgegeben, auch beim Auftreten mehrerer Fehler während eines Einlagerungsvorgangs. Zusätzlich wird für jedes Archivsegment eine WinCC Systemmeldung erzeugt. In der Windows Ereignisanzeige wird unter "Anwendung" ein Eintrag hinzugefügt. Dadurch besteht die Möglichkeit zu überprüfen, welche Archivsegmente den Fehler erzeugen.

- Bei einem nicht signierten Archiv wird der Rückgabewert "0x8004720F" zurückgegeben. In der Ereignisanzeige erfolgt der Eintrag "Validation of database <db_name> failed ! No signature found !".
Das Archiv wird eingelagert.
- Bei einem veränderten Archiv wird der Rückgabewert "0x80047207" zurückgegeben. In der Ereignisanzeige erfolgt der Eintrag "Validation of database <db_name> failed !".
Das Archiv wird nicht eingelagert.

Hinweis

Der Aufruf der Methode "Restore" ist derzeit nur am Server möglich. Es existiert aber ein Beispiel, das zeigt, wie die Methode vom Client aus auf dem Server gestartet werden kann.

Syntax

Objekte Logging, AlarmLogs

```
Ausdruck.Restore [SourcePath] [TimeFrom] [TimeTo] [TimeOut]  
[ServerPrefix]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "Logging" oder "AlarmLogs" zurückgibt.

Objekt DataLogs

```
Ausdruck.Restore [SourcePath] [TimeFrom] [TimeTo] [TimeOut] [Type]  
[ServerPrefix]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "DataLogs" zurückgibt.

Parameter

SourcePath

Pfad zu den Archivdaten.

TimeFrom

Zeitpunkt, von dem an die Archive eingelagert werden sollen.

Bei der Angabe des Zeitpunktes ist auch eine Kurzform möglich, wie im Absatz "Format der Zeitangaben" beschrieben.

TimeTo

Zeitpunkt, bis zu dem die Archivsegmente eingelagert werden sollen.

Bei der Angabe des Zeitpunktes ist auch eine Kurzform möglich, wie im Absatz "Format der Zeitangaben" beschrieben.

Timeout

Timeout in Millisekunden.

Wenn Sie als Wert "-1" eingeben, wird für immer gewartet (infinite). Wenn Sie den Wert "0" eingeben, wird nicht gewartet.

Type

Typ des Archivs.

Dieser Parameter kann (optional) nur für die Einlagerung von Archivsegmenten des Tag Logging verwendet werden.

Folgende Werte können eingegeben werden:

zugeordneter Wert	Typ	Beschreibung
1	hmiDataLogFast	Tag Logging Fast data
2	hmiDataLogSlow	Tag Logging Slow data
3	hmiDataLogAll	Tag Logging Fast and Slow data

ServerPrefix

Reserviert für zukünftige Versionen.

Rückgabewert

Wenn beim Einlagern der Archivsegmente ein Fehler aufgetreten ist, gibt die Methode eine Fehlermeldung zurück. Weitere Informationen finden Sie unter dem Thema "Fehlermeldungen aus dem Bereich Datenbanken".

Format der Zeitangaben

Das Format für die Angabe von Zeitpunkten ist wie folgt definiert: YYYY-MM-DD hh:mm:ss, wobei YYYY das Jahr bezeichnet, MM den Monat, DD den Tag, hh die Stunde, mm die Minute und ss die Sekunde. Beispielsweise wird die Uhrzeit 2 Minuten und eine Sekunde nach 11 Uhr am 26. Juli 2004 so dargestellt: 2004-07-26 11:02:01.

Bei den Parametern "TimeFrom" und "TimeTo" ist die Angabe von Datum und Uhrzeit in einer Kurzform möglich. Dabei sind nicht alle Felder des Formats zu belegen. Die Kurzform bedeutet, dass bei der Datum/Zeit-Angabe ein oder mehrere Parameter, mit dem Sekundenwert beginnend, entfallen können. So kann die Angabe z.B. in den Formaten "YYYY-MM" oder "YYYY-MM-DD hh" erfolgen. Mit der Angabe "TimeFrom" = "2004-09" und "TimeTo" = "2004-10-04" werden alle Archivsegmente von September 2004 bis einschließlich 4. Oktober eingelagert.

Beispiel

Im folgenden Beispiel werden alle Archivsegmente ab dem angegebenen Zeitraum wieder eingelagert und der Rückgabewert als Trace ausgegeben.

```
'VBS184
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.Restore("D:\Folder","2004-09-14","", -1) &
vbNewLine
```

Im folgenden Beispiel werden Tag Logging Slow Archivsegmente des angegebenen Zeitraums wieder eingelagert und der Rückgabewert als Trace ausgegeben.

```
'VBS185
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.DataLogs.Restore("D:\Folder","2004-09-14
12:30:05","2004-09-20 18:30",-1,2) & vbNewLine
```

Im folgenden Beispiel werden Alarm Logging Archivsegmente bis zu dem angegebenen Zeitraum wieder eingelagert und der Rückgabewert als Trace ausgegeben.

```
'VBS186
HMIRuntime.Trace "Ret: " & HMIRuntime.Logging.AlarmLogs.Remove("", "2004-09-20", -1) &
vbNewLine
```

Siehe auch

- Logging (Seite 1318)
- DataLogs (Auflistung) (Seite 1309)
- AlarmLogs (Auflistung) (Seite 1306)

3.5.6.3 Methoden S-Z (Panels, Comfort Panels, RT Advanced, RT Professional)

SelectAll (RT Professional)

Beschreibung

Markiert alle Zeilen in einem tabellenbasierten Control.

Syntax

```
Ausdruck.SelectAll()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

AlarmControl (Seite 1338)

SelectedStatisticAreae (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Statistikbereich festlegen" der Tabellenanzeige aus.

Syntax

```
Ausdruck.SelectedStatisticAreae()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTableControl (Seite 1434)

SelectRow (RT Professional)

Beschreibung

Markiert eine bestimmte Zeile in einem tabellenbasierten Control.

Syntax

```
Ausdruck.SelectRow(ByVal IRow As Long, Optional bExtendSelection As Boolean)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

Parameter	Beschreibung
IRow	Nummer der Zeile, die zu markieren ist.
bExtendSelection	Gibt optional an, ob die aktuelle Markierung erweitert wird. Ist nur relevant, wenn die Mehrfachselektion möglich ist.

Beispiel

- Aktuell ist Zeile 1 markiert. Wenn SelectRow(2, True) aufgerufen wird, sind danach Zeile 1 und Zeile 2 markiert.
- Aktuell ist Zeile 1 markiert. Wenn SelectRow(2, False) oder SelectRow(2) ohne optionalen Parameter aufgerufen wird, ist danach nur Zeile 2 markiert.

Siehe auch

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

AlarmControl (Seite 1338)

ServerExport (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Archiv exportieren" der Rezepturanzeige aus.

Syntax

```
Ausdruck.ServerExport()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

ServerImport (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Archiv importieren" der Rezepturanzeige aus.

Syntax

```
Ausdruck.ServerImport()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

ShowColumnSelection (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Spalten wählen" der Tabellenanzeige aus.

Syntax

```
Ausdruck.ShowColumnSelection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTableControl (Seite 1434)

ShowComment (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Kommentar-Dialog" der Meldeanzeige aus.

Syntax

`Ausdruck.ShowComment()`

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowDisplayOptionsDialog (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Anzeigeoptions-Dialog" der Meldeanzeige aus.

Syntax

`Ausdruck.ShowDisplayOptionsDialog()`

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowEmergencyQuitDialog (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Not-Quittierung" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowEmergencyQuitDialog()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowHelp (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Hilfe" des Controls aus.

Syntax

```
Ausdruck.ShowHelp()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Siehe auch

OnlineTrendControl (Seite 1443)
OnlineTableControl (Seite 1434)
TrendRulerControl (Seite 1508)
UserArchiveControl (Seite 1528)
FunctionTrendControl (Seite 1398)
AlarmControl (Seite 1338)

ShowHideList (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Liste auszublendender Meldungen" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowHideList ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowHitList (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Hitliste" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowHitList ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowInfoText (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Infotext-Dialog" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowInfoText()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowLockDialog (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Sperr-Dialog" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowLockDialog()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowLockList (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Sperrliste" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowLockList()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowLongTermArchiveList (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Langzeitarchivliste" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowLongTermArchiveList()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowMessageList (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Meldeliste" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowMessageList()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowPercentageAxis (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Relative Achse" der f(t)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ShowPercentageAxis()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

ShowPropertyDialog (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Konfigurationsdialog" des Controls aus.

Syntax

```
Ausdruck.ShowPropertyDialog()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

VARIANT

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

FunctionTrendControl (Seite 1398)

AlarmControl (Seite 1338)

ShowSelectArchive (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Datenanbindung wählen" der Rezepturanzeige aus.

Syntax

```
Ausdruck.ShowSelectArchive()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

ShowSelection (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Selektions-Dialog" der Rezepturanzeige aus.

Syntax

```
Ausdruck.ShowSelection ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

ShowSelectionDialog (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Selektions-Dialog" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowSelectionDialog ()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowSelectTimeBase (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Zeitbasis-Dialog" der Rezepturanzeige aus.

Syntax

```
Ausdruck.ShowSelectTimeBase()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

ShowShortTermArchiveList (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Kurzzeitarchivliste" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowShortTermArchiveList()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowSort (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Sortier-Dialog" der Rezepturanzeige aus.

Syntax

```
Ausdruck.ShowSort()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

ShowSortDialog (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Sortier-Dialog" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowSortDialog()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowTagSelection (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Datenanbindung wählen" des Controls aus.

Syntax

```
Ausdruck.ShowTagSelection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

OnlineTableControl (Seite 1434)

FunctionTrendControl (Seite 1398)

ShowTimebaseDialog (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Zeitbasis-Dialog" der Meldeanzeige aus.

Syntax

```
Ausdruck.ShowTimebaseDialog()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

ShowTimeSelection (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Zeitbereich wählen" des Controls aus.

Syntax

```
Ausdruck.ShowTimeSelection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

[OnlineTrendControl \(Seite 1443\)](#)

[OnlineTableControl \(Seite 1434\)](#)

[FunctionTrendControl \(Seite 1398\)](#)

ShowTrendSelection (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Kurven wählen" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ShowTrendSelection()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

- OnlineTrendControl (Seite 1443)
- OnlineTableControl (Seite 1434)
- FunctionTrendControl (Seite 1398)

StartStopUpdate (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Start" oder "Stopp" des Controls aus.

Syntax

```
Ausdruck.StartStopUpdate()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

- OnlineTrendControl (Seite 1443)
- OnlineTableControl (Seite 1434)
- FunctionTrendControl (Seite 1398)

Stop (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Beendet WinCC Runtime.

Syntax

```
Expression.Stop
```

Expression

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "HMIRuntime" zurückgibt.

Parameter

--

Siehe auch

HMIRuntime (Seite 1313)

Trace (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Gibt einen benutzerdefinierten Text über den Betriebssystemkanal für Debug-Meldungen aus.

Die Methode HMIRuntime.Trace funktioniert nur in einer PC basierten Umgebung. Der als Parameter übergebene Text kann mit den Diagnosewerkzeugen "GSC-Diagnose" oder "ApDiag" angezeigt werden. Wenn ein Trace ohne externe Tools notwendig ist, kann die Systemfunktion "ZeigeSystemmeldung" eingesetzt werden.

Syntax

Expression.Trace"STRING"

Expression

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "HMIRuntime" zurückgibt.

Parameter

STRING

Der Text, der als Debug-Meldung ausgegeben wird. Der übergebene Text kann mit den Diagnosewerkzeugen "GSC-Diagnose" oder "ApDiag" angezeigt werden. Wenn ein Trace ohne externe Tools notwendig ist, kann die Systemfunktion "ZeigeSystemmeldung" eingesetzt werden.

Beispiel

Im folgenden Beispiel wird eine Debug-Meldung ausgegeben:

```
'VBS_Example_Trace  
HMIRuntime.Trace "Customized error message"
```

Siehe auch

HMIRuntime (Seite 1313)

UnhideAlarm (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Meldung einblenden" der Meldeanzeige aus.

Syntax

```
Ausdruck.UnhideAlarm()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

UnlockAlarm (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Meldung freigeben" der Meldeanzeige aus.

Syntax

```
Ausdruck.UnlockAlarm()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

AlarmControl (Seite 1338)

UnselectAll (RT Professional)

Beschreibung

Entfernt alle Markierungen in den Zellen eines tabellenbasierten Controls.

Syntax

```
Ausdruck.UnselectAll()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

AlarmControl (Seite 1338)

UnselectRow (RT Professional)

Beschreibung

Entfernt die Markierungen in einer bestimmten Zelle eines tabellenbasierten Controls.

Syntax

```
Ausdruck.UnselectRow(ByVal IRow As Long)
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

Long

Parameter	Beschreibung
IRow	Nummer der Zeile, die zu markieren ist.

Siehe auch

OnlineTableControl (Seite 1434)

TrendRulerControl (Seite 1508)

UserArchiveControl (Seite 1528)

AlarmControl (Seite 1338)

Write (RT Professional)

Beschreibung für Tag-Objekt

Schreibt einen Wert in eine Variable. Über die Eigenschaft "LastError" kann festgestellt werden, ob der Aufruf erfolgreich war.

Wird der Wert der Variablen erfolgreich gesetzt, werden die Eigenschaften des Tag-Objektes mit folgenden Werten belegt:

Eigenschaft	Belegung
Value	Vom Benutzer gesetzter Wert der Variablen (unverändert)
Name	Variablenname (unverändert)
QualityCode	Bad Out of Service
TimeStamp	0
LastError	0
ErrorDescription	" "

Wird der Variablenwert nicht erfolgreich gesetzt, werden die Eigenschaften des Tag-Objektes mit folgenden Werten belegt:

Eigenschaft	Belegung
Value	Vom Benutzer gesetzter Wert der Variablen (unverändert)
Name	Variablenname (unverändert)
QualityCode	Bad Out of Service
TimeStamp	0
LastError	Fehlercode der Schreiboperation
ErrorDescription	Fehlerbeschreibung zu LastError

Syntax

```
Ausdruck.Write [Value],[Writemode]
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Tags-Objekt zurückgibt.

Parameter

Der zu schreibende Wert kann der Methode als Parameter direkt übergeben werden. Wird der Parameter nicht angegeben, wird der Wert der Eigenschaft "Value" verwendet. Mit dem optionalen Parameter "Writemode" können Sie wählen, ob der Variablenwert synchron oder asynchron geschrieben werden soll. Wird der Parameter "Writemode" nicht verwendet, wird als Defaultwert asynchron geschrieben.

Bei einem Schreibvorgang werden keine Informationen zum Zustand der Variablen geliefert.

In der Eigenschaft "Value" steht der Wert, der vor oder bei der Schreiboperation gesetzt wurde, er muss also nicht dem tatsächlichen aktuellen Wert der Variable entsprechen. Wenn Sie die Informationen zu der Variable aktualisieren möchten, verwenden Sie die Read-Methode.

Parameter	Beschreibung
Value (optional)	Wert der Variablen wird angegeben. Der angegebene Wert überschreibt den Wert der Eigenschaft "Value" im Tag-Objekt. Wert der Variablen wird nicht angegeben. Die Variable erhält den aktuellen Wert aus der Eigenschaft "Value" des Tag-Objektes.
Writemode (optional)	0 oder leer: Der Variablenwert wird asynchron geschrieben. 0 ist der Defaultwert. 1: Der Variablenwert wird synchron geschrieben.

Beim asynchronen Schreiben wird sofort in das Variablenabbild geschrieben. Der Anwender erhält keine Rückmeldung darüber, ob der Wert auch in das AS geschrieben wurde.

Beim synchronen Schreiben (direkt in das AS) wird erst geschrieben, wenn das Automatisierungsgerät bereit ist. Der Anwender erhält eine Rückmeldung, wenn das Schreiben nicht erfolgreich war.

Beispiel

Asynchrones Schreiben

```
'VBS104
Dim objTag
Set objTag = HMIRuntime.Tags("Var1")
objTag.Value = 5
objTag.Write
MsgBox objTag.Value
```

oder

```
'VBS105
Dim objTag
Set objTag = HMIRuntime.Tags("Var1")
objTag.Write 5
MsgBox objTag.Value
```

Synchrones Schreiben

```
'VBS106  
Dim objTag  
Set objTag = HMIRuntime.Tags("Var1")  
objTag.Value = 5  
objTag.Write ,1  
MsgBox objTag.Value
```

oder

```
'VBS107  
Dim objTag  
Set objTag = HMIRuntime.Tags("Var1")  
objTag.Write 5, 1  
MsgBox objTag.Value
```

Beschreibung für TagSet-Objekt

Das TagSet-Objekt bietet die Möglichkeit, mehrere Variablen mit einem Aufruf zu schreiben.

Die Funktionsweise ist dabei weitgehend mit der eines Tag-Objektes identisch. Nachfolgend werden nur die Abweichungen beschrieben.

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "TagSet" zurückgibt.

Parameter

Um unterschiedliche Werte zu schreiben muss die Eigenschaft "Value" der einzelnen Tag-Objekte gesetzt und danach Write ohne Parameter "Value" aufgerufen werden. Da die Schreibaufträge zu einem Aufruf zusammengefasst werden ergibt sich eine verbesserte Performance gegenüber den Einzelaufrufen.

Beim TagSet-Objekt ist es nicht möglich mit der "Write"-Methode einen Wert mitzugeben. Die einzelnen Werte müssen über die Eigenschaft "Value" der einzelnen Tag-Objekte gesetzt werden.

Beispiel

Das folgende Beispiel zeigt wie man Variablen in die TagSet Collection aufnimmt, die Variablenwerte setzt und danach schreibt.

```
'VBS173
Dim group
Set group = HMIRuntime.Tags.CreateTagSet
group.Add "Wert1"
group.Add "Wert2"
group("Wert1").Value = 3
group("Wert2").Value = 9
group.Write
```

Setzt man den optionalen Parameter "Writemode" auf 1, so werden Prozess-Variablen synchron (direkt in das AS) geschrieben.

```
group.Write 1
```

WriteTags (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Variablen schreiben" der Rezepturanzeige aus.

Syntax

```
Ausdruck.WriteTags()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

UserArchiveControl (Seite 1528)

ZoomArea (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Zoomen Ausschnitt" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ZoomArea ( )
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ZoomInOut (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Zoomen +/-" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ZoomInOut ( )
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

ZoomInOutTime (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Zeitachse zoomen +/-" der f(t)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ZoomInOutTime()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

ZoomInOutValues (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Wertachse zoomen +/-" der f(t)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ZoomInOutValues()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

ZoomInOutX (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "X-Achse zoomen +/-" der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ZoomInOutX()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

FunctionTrendControl (Seite 1398)

ZoomInOutY (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Y-Achse zoomen +/-" der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ZoomInOutY()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

FunctionTrendControl (Seite 1398)

ZoomMove (RT Professional)

Beschreibung

Führt die Funktion der Schaltfläche "Kurvenbereich verschieben" der f(t)-Kurvenanzeige und der f(x)-Kurvenanzeige aus.

Syntax

```
Ausdruck.ZoomMove()
```

Ausdruck

Erforderlich. Ein Ausdruck, der ein Objekt vom Typ "ScreenItem" zurückgibt.

Parameter

--

Siehe auch

OnlineTrendControl (Seite 1443)

FunctionTrendControl (Seite 1398)

Index

—

_ismbcalnum-Funktion, 178
_ismbcalpha-Funktion, 178
_ismbcdigit-Funktion, 178
_ismbcgraph-Funktion, 178
_ismbclower-Funktion, 178
_ismbcprint-Funktion, 178
_ismbcpunct-Funktion, 178
_ismbcspace-Funktion, 178
_ismbcupper-Funktion, 178
_mbscat-Funktion, 178
_mbschr-Funktion, 178
_mbscmp-Funktion, 178
_mbscpy-Funktion, 178
_mbscspn-Funktion, 178
_mbsdec-Funktion, 178
_mbsinc-Funktion, 178
_mbslen-Funktion, 178
_mbsncat-Funktion, 178
_mbsncmp-Funktion, 178
_mbsncpy-Funktion, 178
_mbspbrk-Funktion, 178
_mbsrchr-Funktion, 178
_mbsspn-Funktion, 178
_mbsstr-Funktion, 178
_mbstok-Funktion, 178
_mbstrlen-Funktion, 178

{

{"ColumnDateFormat-Eigenschaft (VBS)"
{"Eigenschaften (VBS)";"ColumnDateFormat"}, 1618
{"ColumnTimeFormat-Eigenschaft (VBS)"
{"Eigenschaften (VBS)";"ColumnTimeFormat"}, 1622

A

Abmelden, 1193
abort-Funktion, 180
AboveUpperLimitColor-Eigenschaft (VBS), 1544
abs-Funktion, 180
AcceptOnExit-Eigenschaft (VBS), 1545
AcceptOnFull-Eigenschaft (VBS), 1545
AcknowledgeAlarm, 1149
acos-Funktion, 176
ActivateDynamic, 2158

Activate-Eigenschaft (VBS), 2067
Activate-Methode, 2155
ActivateNextScreen, 23
ActivatePreviousScreen, 24, 1149
ActivateScreen, 25, 1150, 1240
ActivateScreenByNumber, 1151
ActivateScreenInScreenWindow, 26, 1241
ActivateStartScreen, 27
ActivateStoredScreen, 28
ActivateSystemDiagnosticsView, 1152
Active-Eigenschaft (VBS), 2090
ActiveScreen-Eigenschaft (VBS), 1547
ActiveScreenItem-Objekt, 1549
ActualPointIndex-Eigenschaft (VBS), 1549
ActualPointLeft-Eigenschaft (VBS), 1550
ActualPointTop-Eigenschaft (VBS), 1550
AdaptBorder-Eigenschaft (VBS), 1551
AdaptFontSizeToLineHeight-Eigenschaft (VBS), 2091
AdaptPicture-Eigenschaft (VBS), 1552
AdaptScreenToWindow-Eigenschaft (VBS), 1552
AdaptWindowtoScreen-Eigenschaft (VBS), 1552
Address-Eigenschaft (VBS), 1553
AddressEnabled-Eigenschaft (VBS), 1553
AdjustBorder3DWithStyle-Eigenschaft (VBS), 2067
AdjustRulerWindow-Eigenschaft (VBS), 2092
AdressPreview-Eigenschaft (VBS), 1553
AktiviereBild, 25, 1150, 1240
AktiviereBildInBildfenster, 26, 1241
AktiviereBildMitNummer, 1151
AktiviereSystemdiagnoseAnzeige, 1152
AktiviereVorherigesBild, 1149
AlarmClasses-Eigenschaft (VBS), 1554
AlarmColor-Eigenschaft (VBS), 1554
AlarmControl-Objekt, 1338
Alarm-Eigenschaft (VBS), 1554
AlarmFilter-Eigenschaft (VBS), 2068
AlarmHigh-Eigenschaft (VBS), 2092
AlarmID-Eigenschaft, 1555
AlarmLog-Eigenschaft (VBS), 1555
Alarmlogs-Objekt, 1306
AlarmLow-Eigenschaft (VBS), 2006
AlarmLowerLimitColor-Eigenschaft (VBS), 1556
AlarmLowerLimit-Eigenschaft (VBS), 1556
AlarmLowerLimitEnabled-Eigenschaft (VBS), 1557
AlarmLowerLimitRelative-Eigenschaft (VBS), 1557
Alarm-Objekt, 1304
Alarms-Objekt (Auflistung), 1305
AlarmSource-Eigenschaft (VBS), 1558

AlarmUpperLimitColor-Eigenschaft (VBS), 1559
AlarmUpperLimit-Eigenschaft (VBS), 1558
AlarmUpperLimitEnabled-Eigenschaft (VBS), 1559
AlarmUpperLimitRelative-Eigenschaft (VBS), 1560
Alignment-Eigenschaft (VBS), 2093
AlignmentLeft-Eigenschaft (VBS), 2093
AlignmentTop-Eigenschaft (VBS), 2007
AllFilters-Eigenschaft (VBS), 1560
AllFiltersForHitlist-Eigenschaft (VBS), 1561
AllowEdit-Eigenschaft (VBS), 1561
AllowMenu-Eigenschaft (VBS), 1561
AllowPersistence-Eigenschaft (VBS), 2094
Analog-Eigenschaft (VBS), 1561
AngleMax-Eigenschaft (VBS), 1561
AngleMin-Eigenschaft (VBS), 1562
Anmelden, 1194
Appearance-Eigenschaft (VBS), 1563
ApplicationWindow-Objekt, 1479
ApplyProjectSettings-Eigenschaft (VBS), 1563
ApplyProjectSettingsForDesignMode-Eigenschaft (VBS), 1563
ArchiveLogFile, 1153
ArchiviereProtokolldatei, 1153
asctime-Funktion, 182
asin-Funktion, 176
AskOperationMotive-Eigenschaft (VBS), 1563
Assignments-Eigenschaft (VBS), 1564
AssociatedS7GraphDBTag-Eigenschaft (VBS), 1564
AssumeOnExit-Eigenschaft (VBS), 2008
AssumeOnFull-Eigenschaft (VBS), 2008
atan2-Funktion, 177
atan-Funktion, 176
atof-Funktion, 180
atoi-Funktion, 180
atol-Funktion, 180
AttachDB, 2161
Auflistungen, 1336
Auflistungen in VBS
 Alarms-Objekt (Auflistung), 1305
Auflistungen:DataSet, 1311
Auflistungen:Layers, 1317
Auflistungen:Tags-Objekt (Auflistung), 1335
Authorization-Eigenschaft (VBS), 1565
AutoCompleteColumns-Eigenschaft (VBS), 1566
AutoCompleteRows-Eigenschaft (VBS), 1567
AutoScroll-Eigenschaft (VBS), 1567
AutoSelectionColors-Eigenschaft (VBS), 1568
AutoSelectionRectColors-Eigenschaft (VBS), 1568
AutoSizing-Eigenschaft (VBS), 1569
AverageLast15Values-Eigenschaft (VBS), 1569
AxisXBunchCount-Eigenschaft (VBS), 1570
AxisXMarkCount-Eigenschaft (VBS), 1570

AxisXShowBunchValues-Eigenschaft (VBS), 1570
AxisY1BunchCount-Eigenschaft (VBS), 1570
AxisY1MarkCount-Eigenschaft (VBS), 1570
AxisY1ShowBunchValues-Eigenschaft (VBS), 1570
AxisY2BunchCount-Eigenschaft (VBS), 1570
AxisY2MarkCount-Eigenschaft (VBS), 1571
AxisY2ShowBunchValues-Eigenschaft (VBS), 1571

B

BackColorBottom-Eigenschaft (VBS), 1573
BackColor-Eigenschaft (VBS), 1571
BackColorTop-Eigenschaft (VBS), 1573
BackFillStyle-Eigenschaft (VBS), 1574
BackFillStyleReadOnlySpecial-Eigenschaft (VBS), 2095
BackFlashingColorOff-Eigenschaft (VBS), 1575
BackFlashingColorOn-Eigenschaft (VBS), 1576
BackFlashingEnabled-Eigenschaft (VBS), 1577
BackFlashingRate-Eigenschaft (VBS), 1578
BackgroundColor-Eigenschaft (VBS), 1579
BackupRAMFileSystem, 1155
BarBackColor-Eigenschaft (VBS), 1579
BarBackFillStyle-Eigenschaft (VBS), 1579
BarBackFlashingColorOff-Eigenschaft (VBS), 1581
BarBackFlashingColorOn-Eigenschaft (VBS), 1581
BarBackFlashingEnabled-Eigenschaft (VBS), 1581
BarBackFlashingRate-Eigenschaft (VBS), 1582
BarColor-Eigenschaft (VBS), 1582
BarEdgeStyle-Eigenschaft (VBS), 1582
Bar-Objekt, 1354
BarStartValue-Eigenschaft (VBS), 2095
BaseScreenName-Eigenschaft (VBS), 1583
Battery-Objekt, 1359
BearbeiteMeldung, 1167
BeginTime(i)-Eigenschaft (VBS), 2096
BelowLowerLimitColor-Eigenschaft (VBS), 1583
BitNumber-Eigenschaft (VBS), 1583
BlinkColor-Eigenschaft (VBS), 1584
BlinkMode-Eigenschaft (VBS), 1585
BlinkSpeed-Eigenschaft (VBS), 1585
Blocks-Eigenschaft (VBS), 1586
BorderBackColor-Eigenschaft (VBS), 1586
BorderBrightColor3D-Eigenschaft (VBS), 1587
BorderColor-Eigenschaft (VBS), 1588
BorderEnabled-Eigenschaft (VBS), 1590
BorderEndStyle-Eigenschaft (VBS), 1590
BorderFlashingColorOff-Eigenschaft (VBS), 1591
BorderFlashingColorOn-Eigenschaft (VBS), 1592
BorderFlashingEnabled-Eigenschaft (VBS), 1593
BorderFlashingRate-Eigenschaft (VBS), 1594
BorderInnerStyle3D-Eigenschaft (VBS), 1595

BorderInnerWidth3D-Eigenschaft (VBS), 1596
 BorderOuterStyle3D-Eigenschaft (VBS), 1596
 BorderOuterWidth3D-Eigenschaft (VBS), 1597
 BorderShadeColor3D-Eigenschaft (VBS), 1597
 BorderStyle3D-Eigenschaft (VBS), 1599
 BorderStyle-Eigenschaft (VBS), 1598
 BorderWidth3D-Eigenschaft (VBS), 1601
 BorderWidth-Eigenschaft (VBS), 1599
 BottomMargin-Eigenschaft (VBS), 1602
 BrowserView-Objekt, 1417
 bsearch-Funktion, 180
 BusyText-Eigenschaft (VBS), 1602
 ButtonBackColor-Eigenschaft (VBS), 1602
 ButtonBackFillStyle-Eigenschaft (VBS), 1602
 ButtonBarElements-Eigenschaft (VBS), 1602
 ButtonBarStyle-Eigenschaft (VBS), 1602
 ButtonBorderBackColor-Eigenschaft (VBS), 1602
 ButtonBorderColor-Eigenschaft (VBS), 1603
 ButtonBorderWidth-Eigenschaft (VBS), 1603
 ButtonCommand-Eigenschaft (VBS), 2011
 ButtonCornerRadius-Eigenschaft (VBS), 1603
 ButtonEdgeStyle-Eigenschaft (VBS), 1603
 ButtonFirstGradientColor-Eigenschaft (VBS), 1603
 ButtonFirstGradientOffset-Eigenschaft (VBS), 1603
 ButtonMiddleGradientColor-Eigenschaft (VBS), 1603
 Button-Objekt, 1360
 ButtonSecondGradientColor-Eigenschaft (VBS), 1604
 ButtonSecondGradientOffset-Eigenschaft (VBS), 1604
 BV_ColumnWidth_Date-Eigenschaft (VBS), 1604
 BV_ColumnWidth_Event-Eigenschaft (VBS), 1604
 BV_ColumnWidth_EventSeverity-Eigenschaft (VBS), 1604
 BV_ColumnWidth_EventState-Eigenschaft (VBS), 1604
 BV_ColumnWidth_Number-Eigenschaft (VBS), 1604
 BV_ColumnWidth_Time-Eigenschaft (VBS), 1605
 BV_ItemText_Date-Eigenschaft (VBS), 1605
 BV_ItemText_Event-Eigenschaft (VBS), 1605
 BV_ItemText_EventSeverity-Eigenschaft (VBS), 1605
 BV_ItemText_EventState-Eigenschaft (VBS), 1605
 BV_ItemText_Number-Eigenschaft (VBS), 1605
 BV_ItemText_Time-Eigenschaft (VBS), 1605
 BV_ShowItem_Date-Eigenschaft (VBS), 1606
 BV_ShowItem_Event-Eigenschaft (VBS), 1606
 BV_ShowItem_EventSeverity-Eigenschaft (VBS), 1606
 BV_ShowItem_EventState-Eigenschaft (VBS), 1606
 BV_ShowItem_Number-Eigenschaft (VBS), 1606
 BV_ShowItem_Time-Eigenschaft (VBS), 1606

C

CalculateStatistic, 2162
 CalibrateTouchScreen, 1155
 calloc-Funktion, 180
 CaptionBackColor-Eigenschaft (VBS), 1607
 CaptionColor-Eigenschaft (VBS), 1608
 Caption-Eigenschaft (VBS), 1607
 CaptionText-Eigenschaft (VBS), 1608
 CaptionTop-Eigenschaft (VBS), 1609
 ceil-Funktion, 177
 CellCut-Eigenschaft (VBS), 1610
 CellSpaceBottom-Eigenschaft (VBS), 1610
 CellSpaceLeft-Eigenschaft (VBS), 1611
 CellSpaceRight-Eigenschaft (VBS), 1611
 CellSpaceTop-Eigenschaft (VBS), 1612
 CenterColor-Eigenschaft (VBS), 1612
 CenterSize-Eigenschaft (VBS), 1613
 CentrePoint-Eigenschaft (VBS), 2097
 CentrePointLeft-Eigenschaft (VBS), 2097
 CentrePointTop-Eigenschaft (VBS), 2098
 C-Funktionen
 _ismbcalnum-Funktion, 178
 _ismbcalpha-Funktion, 178
 _ismbcdigit-Funktion, 178
 _ismbcgraph-Funktion, 178
 _ismbcclower-Funktion, 178
 _ismbcprint-Funktion, 178
 _ismbcpunct-Funktion, 178
 _ismbcspc-Funktion, 178
 _ismbcupper-Funktion, 178
 _mbscat-Funktion, 178
 _mbschr-Funktion, 178
 _mbscmp-Funktion, 178
 _mbscpy-Funktion, 178
 _mbscspn-Funktion, 178
 _mbsdec-Funktion, 178
 _mbsinc-Funktion, 178
 _mbslen-Funktion, 178
 _mbsncat-Funktion, 178
 _mbsncmp-Funktion, 178
 _mbsncpy-Funktion, 178
 _mbspbrk-Funktion, 178
 _mbsrchr-Funktion, 178
 _mbscspn-Funktion, 178
 _mbsstr-Funktion, 178
 _mbstok-Funktion, 178
 _mbstrlen-Funktion, 178
 abort-Funktion, 180
 abs-Funktion, 180
 acos-Funktion, 176

asctime-Funktion, 182
asin-Funktion, 176
atan2-Funktion, 177
atan-Funktion, 176
atof-Funktion, 180
atoi-Funktion, 180
atol-Funktion, 180
bsearch-Funktion, 180
calloc-Funktion, 180
ceil-Funktion, 177
clearerr-Funktion, 180
clock-Funktion, 182
cos-Funktion, 177
cosh-Funktion, 177
ctime-Funktion, 182
difftime-Funktion, 182
div-Funktion, 180
exit-Funktion, 180
exp-Funktion, 177
fabs-Funktion, 177
fclose-Funktion, 179
feof-Funktion, 179
ferror-Funktion, 179
fflush-Funktion, 179
fgetc-Funktion, 179
fgetpos-Funktion, 179
fgets-Funktion, 179
floor-Funktion, 177
fmod-Funktion, 177
fopen-Funktion, 179
fputc-Funktion, 179
fputs-Funktion, 179
fread-Funktion, 180
free-Funktion, 180
freopen-Funktion, 179
frexp-Funktion, 177
fseek-Funktion, 179
fsetpos-Funktion, 179
ftell-Funktion, 179
fwrite-Funktion, 180
getc-Funktion, 179
getenv-Funktion, 180
gmtime-Funktion, 182
isalnum-Funktion, 175
isalpha-Funktion, 175
isdigit-Funktion, 175
isgraph-Funktion, 175
islower-Funktion, 175
isprint-Funktion, 175
ispunct-Funktion, 175
isspace-Funktion, 175
isupper-Funktion, 175
isxdigit-Funktion, 175
labs-Funktion, 180
ldexp-Funktion, 177
ldiv-Funktion, 180
localtime-Funktion, 182
log10-Funktion, 177
log-Funktion, 177
malloc-Funktion, 180
memchr-Funktion, 177
memcmp-Funktion, 177
memcpy-Funktion, 177
memmove-Funktion, 177
memset-Funktion, 177
mktime-Funktion, 182
modf-Funktion, 177
pow-Funktion, 177
putc-Funktion, 179
qsort-Funktion, 181
rand-Funktion, 180
realloc-Funktion, 181
remove-Funktion, 179
rename-Funktion, 179
rewind-Funktion, 180
setbuf-Funktion, 180
setvbuf-Funktion, 179
sin-Funktion, 177
sinh-Funktion, 177
sqrt-Funktion, 177
srand-Funktion, 180
strcat-Funktion, 181
strchr-Funktion, 181
strcmp-Funktion, 181
strcpy-Funktion, 181
strcspn-Funktion, 181
strerror-Funktion, 181
strftime-Funktion, 182
strlen-Funktion, 181
strncat-Funktion, 181
strncmp-Funktion, 181
strncpy-Funktion, 181
strpbrk-Funktion, 181
strrchr-Funktion, 181
strspn-Funktion, 181
strstr-Funktion, 181
strtod-Funktion, 180
strtok-Funktion, 181
strtol-Funktion, 180
strtoul-Funktion, 180
system-Funktion, 180
tan-Funktion, 177
tanh-Funktion, 177
time-Funktion, 182

- tmpfile-Funktion, 179
- tmpnam-Funktion, 179
- tolower-Funktion, 175
- toupper-Funktion, 175
- ungetc-Funktion, 179
- fprintf-Funktion, 179
- vsprintf-Funktion, 179
- ChangeConnection, 1156
- ChangeMouseCursor-Eigenschaft (VBS), 1613
- ChannelDiagnose-Objekt, 1364
- Check-Box-Objekt, 1366
- CheckMarkAlignment-Eigenschaft (VBS), 1614
- CheckMarkCount-Eigenschaft (VBS), 1614
- Circle-Objekt, 1369
- CircleSegment-Objekt, 1372
- CircularArc-Objekt, 1375
- ClearAlarmBuffer, 1158
- ClearAlarmBufferProtooolLegacy, 1159
- clearerr-Funktion, 180
- ClearLog, 1162
- ClearOnError-Eigenschaft (VBS), 1615
- ClearOnFocus-Eigenschaft (VBS), 1615
- clock-Funktion, 182
- clock-Funktionen
 - asctime-Funktion, 182
- Clock-Objekt, 1377
- Closeable-Eigenschaft (VBS), 1616
- CloseAllLogs, 1162
- Codieren, 1167
- CodierenEx, 1168
- ColorChangeHysteresis-Eigenschaft (VBS), 1617
- ColorChangeHysteresisEnabled-Eigenschaft (VBS), 1618
- Color-Eigenschaft (VBS), 1616
- ColumnAlignment-Eigenschaft (VBS), 1618
- ColumnColor(i)-Eigenschaft (VBS), 2099
- ColumnDisplayName(i)-Eigenschaft (VBS), 2100
- ColumnResize-Eigenschaft (VBS), 1619
- ColumnScrollbar-Eigenschaft (VBS), 1620
- Columns-Eigenschaft (VBS), 1620
- ColumnSettingsBufferView-Eigenschaft (VBS), 1621
- ColumnSettings-Eigenschaft (VBS), 1620
- ColumnSizingEnable-Eigenschaft (VBS), 2070
- ColumnsMoveable-Eigenschaft (VBS), 1621
- ColumnTextAckGroup-Eigenschaft (VBS), 1621
- ColumnTextAlarmState-Eigenschaft (VBS), 1621
- ColumnTextAlarmText-Eigenschaft (VBS), 1621
- ColumnTextClassName-Eigenschaft (VBS), 1621
- ColumnTextDevice-Eigenschaft (VBS), 1622
- ColumnTextDiagnosable-Eigenschaft (VBS), 1622
- ColumnTextNumber-Eigenschaft (VBS), 1622
- ColumnTextTime-Eigenschaft (VBS), 1622
- ColumnTitleAlignment-Eigenschaft (VBS), 1623
- ColumnTitles-Eigenschaft (VBS), 1623
- ColumnUpdateEnabled(i)-Eigenschaft (VBS), 2100
- ComboBoxFont-Eigenschaft (VBS), 1624
- Command-Eigenschaft (VBS), 2101
- CommonTimeAxisColor-Eigenschaft (VBS), 2101
- CompatibilityMode-Eigenschaft (VBS), 1624
- ConfigureTimeAxis(i)-Eigenschaft (VBS), 2102
- ConnectionType-Eigenschaft (VBS), 1624
- ConnectOnStart-Eigenschaft (VBS), 1624
- Connector-Objekt, 1379
- ConnectTrendWindows-Eigenschaft (VBS), 1625
- ContinousChange-Eigenschaft (VBS), 1625
- Control Center
 - Aktualisierungsvariablen festlegen, 462
 - Benachrichtigungsfunktion anbinden, 270
 - Benachrichtigungsfunktion entfernen, 294, 302
 - Datentyp einer Variablen ermitteln, 380
 - Grenzwerte einer Variablen ermitteln, 373
 - Informationen zu einer Variablengruppe auflisten, 334
 - Variablen einer strukturierten Variablen auflisten, 433
 - Variablen-Auswahldialog öffnen, 412, 417
 - Variableneigenschaften über Dialog bearbeiten, 407
 - Variablen-ID ermitteln, 362
 - Variablenname ermitteln, 362
 - Variablenwerte ändern, 388
 - Variablenwerte ermitteln, 344
 - Variablenwerte mit Aktualisierung ermitteln, 358
 - Variablenwerte mit Benachrichtigung ändern, 397
 - Variablenwerte mit Benachrichtigung und Meldungsausgabe ändern, 403
 - Variablenwerte mit Meldungsausgabe ändern, 393
 - Verbindungsdaten auflisten, 444
- ControlDesignMode-Eigenschaft (VBS), 1829
- Controls
 - WinCC MediaControl, 1427
- ControlSmartServer, 1163
- ControlWebServer, 1164
- CopyRows, 2163
- CornerRadius-Eigenschaft (VBS), 1626
- CornerStyle-Eigenschaft (VBS), 1626
- cos-Funktion, 177
- cosh-Funktion, 177
- CountDivisions-Eigenschaft (VBS), 1628
- Count-Eigenschaft (VBS), 1627
- CountOfLinesPerAlarms-Eigenschaft (VBS), 1628
- CountOfVisibleAlarms-Eigenschaft (VBS), 1628
- CountSubDivisions-Eigenschaft (VBS), 1628

CountValueColumns-Eigenschaft (VBS), 2017
CountVisibleItems-Eigenschaft (VBS), 1629
CreateTagSet-Methode, 2164
C-Textliste, 707, 708
ctime-Funktion, 182
ctype-Funktionen
 isalnum-Funktion, 175
 isalpha-Funktion, 175
 isdigit-Funktion, 175
 isgraph-Funktion, 175
 islower-Funktion, 175
 isprint-Funktion, 175
 ispunct-Funktion, 175
 isspace-Funktion, 175
 isupper-Funktion, 175
 isxdigit-Funktion, 175
 tolower-Funktion, 175
 toupper-Funktion, 175
CurrentColumnIndex-Eigenschaft (VBS), 2017
CurrentCurveIndex-Eigenschaft (VBS), 2102
CursorControl-Eigenschaft (VBS), 1630
CurveColor(i)-Eigenschaft (VBS), 2103
CurveLineWidth(i)-Eigenschaft (VBS), 2104
CurvesCount-Eigenschaft (VBS), 2104
Curves-Eigenschaft (VBS), 1630
CurveUpdateEnabled(i)-Eigenschaft (VBS), 2105
CutRows, 2164

D

DangerRangeColor-Eigenschaft (VBS), 1631
DangerRangeStart-Eigenschaft (VBS), 1631
DangerRangeVisible-Eigenschaft (VBS), 1632
DataFormat-Eigenschaft (VBS), 1632
DataIndex(i)-Eigenschaft (VBS), 2105
DataItem-Objekt, 1308
DataLogs-Objekt, 1309
DataLogTag-Eigenschaft (VBS), 2106
DataRecordNameCaption-Eigenschaft (VBS), 1633
DataRecordNrCaption-Eigenschaft (VBS), 1634
DataSet-Objekt (Auflistung), 1311
DataX(i)-Eigenschaft (VBS), 2107
DataXY(i)-Eigenschaft (VBS), 2107
DataY(i)-Eigenschaft (VBS), 2108
DateTimeField-Objekt, 1382
DecreaseTag, 29, 1166, 1242
DeleteDataRecord, 1159
DeleteDataRecordMemory, 1161
DeleteDate(i)-Eigenschaft (VBS), 2108
DeleteEnable-Eigenschaft (VBS), 2018
DeleteRows, 2167
DetachDB, 2167
DialColor-Eigenschaft (VBS), 1634
DialFillStyle-Eigenschaft (VBS), 1635
DialSize-Eigenschaft (VBS), 1635
difftime-Funktion, 182
Direction-Eigenschaft (VBS), 2109
DiskSpaceView-Objekt, 1384
DisplayButton2Plc-Eigenschaft (VBS), 1636
DisplayButtonComparison-Eigenschaft (VBS), 1636
DisplayButtonDelete-Eigenschaft (VBS), 1636
DisplayButtonFromPlc-Eigenschaft (VBS), 1636
DisplayButtonHelp-Eigenschaft (VBS), 1636
DisplayButtonNew-Eigenschaft (VBS), 1637
DisplayButtonSaveAs-Eigenschaft (VBS), 1637
DisplayButtonSave-Eigenschaft (VBS), 1637
DisplayComboBox-Eigenschaft (VBS), 1637
DisplayGridLines-Eigenschaft (VBS), 1637
DisplayLabeling-Eigenschaft (VBS), 1637
DisplayName(i)-Eigenschaft (VBS), 2109
DisplayNumbers-Eigenschaft (VBS), 1637
DisplayOptions-Eigenschaft (VBS), 1638
DisplayStatusBar-Eigenschaft (VBS), 1638
DisplayTable-Eigenschaft (VBS), 1638
div-Funktion, 180
DMAddNotify (Control Center), 270
DMEnumConnectionDataExStr (Control Center), 444
DMEnumVarGrpDataExStr (Control Center), 334
DMFireNotifyData (Control Center), 294
DMGetValueExStr (Control Center), 344
DMGetValueWaitExStr (Control Center), 358
DMGetVarInfoExStr (Control Center), 362
DMGetVarLimitsExStr (Control Center), 373
DMGetVarTypeExStr (Control Center), 380
DMRemoveNotify (Control Center), 302
DMSetValueExStr (Control Center), 388
DMSetValueMessageExStr (Control Center), 393
DMSetValueWaitExStr (Control Center), 397
DMSetValueWaitMessageExStr (Control Center), 403
DMShowVarDatabaseExStr (Control Center), 412
DMShowVarDatabaseMultiExStr (Control Center), 417
DMShowVarPropertiesExStr (Control Center), 407
DMStartVarUpdateExStr (Control Center), 462
DoubleClickAction-Eigenschaft (VBS), 1638
DrawAlwaysInsideFrame-Eigenschaft (VBS), 2018
DrawEnhancedHTMLBrowser-Eigenschaft (VBS), 2110
DrawInsideFrame-Eigenschaft (VBS), 1639
DrawStylishButton-Eigenschaft (VBS), 2019
Drive-Eigenschaft (VBS), 1640
DruckeBild, 1203

DruckeProtokoll, 1203

E

EdgeStyle-Eigenschaft (VBS), 1640

Edit, 2168

Editable(i)-Eigenschaft (VBS), 2110

EditAlarm, 1167

EditEnabled-Eigenschaft (VBS), 2111

EditOnFocus-Eigenschaft (VBS), 1642

Eigenschaften (VBS)

 AboveUpperLimitColor, 1544

 AcceptOnExit, 1545

 AcceptOnFull, 1545

 AccessPath, 1546

 ActiveProject, 1546

 ActiveScreen, 1547, 1548

 ActualPointIndex, 1549

 ActualPointLeft, 1550

 ActualPointTop, 1550

 AdaptBorder, 1551

 AdaptFontSizeToLineHeight, 2091

 AdaptPicture, 1552

 AdaptScreenToWindow, 1552

 AdaptWindowtoScreen, 1552

 Address, 1553

 AddressEnabled, 1553

 AdjustBorder3DWithStyle, 2067

 AdjustRulerWindow, 2092

 AdressPreview, 1553

 Alarm, 1554

 AlarmClasses, 1554

 AlarmColor, 1554

 AlarmFilter, 2068

 AlarmHigh, 2092

 AlarmLog, 1555

 AlarmLow, 2006

 AlarmLowerLimit, 1556

 AlarmLowerLimitColor, 1556

 AlarmLowerLimitEnabled, 1557

 AlarmLowerLimitRelative, 1557

 AlarmSource, 1558

 AlarmUpperLimit, 1558

 AlarmUpperLimitColor, 1559

 AlarmUpperLimitEnabled, 1559

 AlarmUpperLimitRelative, 1560

 Alignment, 2093

 AlignmentLeft, 2093

 AlignmentTop, 2007

 AllFilters, 1560

 AllFiltersForHitlist, 1561

 AllowEdit, 1561

 AllowMenu, 1561

 AllowPersistence, 2094

 Analog, 1561

 AngleMax, 1561

 AngleMin, 1562

 Appearance, 1563

 ApplyProjectSettings, 1563

 ApplyProjectSettingsForDesignMode, 1563

 AskOperationMotive, 1563

 Assignments, 1564

 AssociatedS7GraphDBTag, 1564

 AssumeOnExit, 2008

 AssumeOnFull, 2008

 Authorization, 1565

 AutoCompleteColumns, 1566

 AutoCompleteRows, 1567

 AutoScroll, 1567

 AutoSelectionColors, 1568

 AutoSelectionRectColors, 1568

 AutoSizing, 1569

 AverageLast15Values, 1569

 AxisXBunchCount, 1570

 AxisXMarkCount, 1570

 AxisXShowBunchValues, 1570

 AxisY1BunchCount, 1570

 AxisY1MarkCount, 1570

 AxisY1ShowBunchValues, 1570

 AxisY2BunchCount, 1570

 AxisY2MarkCount, 1571

 AxisY2ShowBunchValues, 1571

 BackColor, 1571

 BackColorBottom, 1573

 BackColorTop, 1573

 BackFillStyle, 1574

 BackFillStyleReadOnlySpecial, 2095

 BackFlashingColorOff, 1575

 BackFlashingColorOn, 1576

 BackFlashingEnabled, 1577

 BackFlashingRate, 1578

 BackgroundColor, 1579

 BackPictureName, 2040

 BarBackColor, 1579

 BarBackFillStyle, 1579

 BarBackFlashingColorOff, 1581

 BarBackFlashingColorOn, 1581

 BarBackFlashingEnabled, 1581

 BarBackFlashingRate, 1582

 BarColor, 1582

 BarEdgeStyle, 1582

 BarStartValue, 2095

 BaseScreenName, 1583

 BeginTime(i), 2096

BelowLowerLimitColor, 1583
BitNumber, 1583
BlinkColor, 1584
BlinkMode, 1585
BlinkSpeed, 1585
Blocks, 1586
BorderBackColor, 1586
BorderBrightColor3D, 1587
BorderColor, 1588
BorderEnabled, 1590
BorderEndStyle, 1590
BorderFlashingColorOff, 1591
BorderFlashingColorOn, 1592
BorderFlashingEnabled, 1593
BorderFlashingRate, 1594
BorderInnerStyle3D, 1595
BorderInnerWidth3D, 1596
BorderOuterStyle3D, 1596
BorderOuterWidth3D, 1597
BorderShadeColor3D, 1597
BorderStyle, 1598
BorderStyle3D, 1599
BorderWidth, 1599
BorderWidth3D, 1601
BottomMargin, 1602
BusyText, 1602
ButtonBackColor, 1602
ButtonBackFillStyle, 1602
ButtonBarElements, 1602
ButtonBarStyle, 1602
ButtonBorderBackColor, 1602
ButtonBorderColor, 1603
ButtonBorderWidth, 1603
ButtonCommand, 2011
ButtonCornerRadius, 1603
ButtonEdgeStyle, 1603
ButtonFirstGradientColor, 1603
ButtonFirstGradientOffset, 1603
ButtonMiddleGradientColor, 1603
ButtonSecondGradientColor, 1604
ButtonSecondGradientOffset, 1604
BV_ColumnWidth_Date, 1604
BV_ColumnWidth_Event, 1604
BV_ColumnWidth_EventSeverity, 1604
BV_ColumnWidth_EventState, 1604
BV_ColumnWidth_Number, 1604
BV_ColumnWidth_Time, 1605
BV_ItemText_Date, 1605
BV_ItemText_Event, 1605
BV_ItemText_EventSeverity, 1605
BV_ItemText_EventState, 1605
BV_ItemText_Number, 1605
BV_ItemText_Time, 1605
BV_ShowItem_Date, 1606
BV_ShowItem_Event, 1606
BV_ShowItem_EventSeverity, 1606
BV_ShowItem_EventState, 1606
BV_ShowItem_Number, 1606
BV_ShowItem_Time, 1606
Caption, 1607
CaptionBackColor, 1607
CaptionColor, 1608
CaptionText, 1608
CaptionTop, 1609
CellCut, 1610
CellSpaceBottom, 1610
CellSpaceLeft, 1611
CellSpaceRight, 1611
CellSpaceTop, 1612
CenterColor, 1612
CenterSize, 1613
CentrePoint, 2097
CentrePointLeft, 2097
CentrePointTop, 2098
ChangeMouseCursor, 1613
CheckMarkAlignment, 1614
CheckMarkCount, 1614
ClearOnError, 1615
ClearOnFocus, 1615
Closeable, 1616
Color, 1616
ColorChangeHysteresis, 1617
ColorChangeHysteresisEnabled, 1618
ColumnAlignment, 1618
ColumnColor(i), 2099
ColumnDisplayName(i), 2100
ColumnResize, 1619
Columns, 1620
ColumnScrollbar, 1620
ColumnSettings, 1620
ColumnSettingsBufferView, 1621
ColumnSizingEnable, 2070
ColumnsMoveable, 1621
ColumnTextAckGroup, 1621
ColumnTextAlarmState, 1621
ColumnTextAlarmText, 1621
ColumnTextClassName, 1621
ColumnTextDate, 1621
ColumnTextDevice, 1622
ColumnTextDiagnosable, 1622
ColumnTextNumber, 1622
ColumnTextTime, 1622
ColumnTitleAlignment, 1623
ColumnTitles, 1623

ColumnUpdateEnabled(i), 2100
ComboboxFont, 1624
Command, 2101
CommonTimeAxisColor, 2101
CompatibilityMode, 1624
ConfigureTimeAxis(i), 2102
ConnectionType, 1624
ConnectOnStart, 1624
ConnectTrendWindows, 1625
ContinousChange, 1625
ControlDesignMode, 1829
CornerRadius, 1626
CornerStyle, 1626
Count, 1627
CountDivisions, 1628
CountOfLinesPerAlarms, 1628
CountOfVisibleAlarms, 1628
CountSubDivisions, 1628
CountValueColumns, 2017
CountVisibleItems, 1629
CurrentColumnIndex, 2017
CurrentCurveIndex, 2102
CursorControl, 1630
CurveColor(i), 2103
CurveLineWidth(i), 2104
Curves, 1630
CurvesCount, 2104
CurveUpdateEnabled(i), 2105
DangerRangeColor, 1631
DangerRangeStart, 1631
DangerRangeVisible, 1632
DataFormat, 1632
DataIndex(i), 2105
DataLogTag, 2106
DataRecordNameCaption, 1633
DataRecordNrCaption, 1634
DataX(i), 2107
DataXY(i), 2107
DataY(i), 2108
DeleteDate(i), 2108
DeleteEnable, 2018
DialColor, 1634
DialFillStyle, 1635
DialSize, 1635
Direction, 2109
DisplayButton2Plc, 1636
DisplayButtonComparison, 1636
DisplayButtonDelete, 1636
DisplayButtonFromPlc, 1636
DisplayButtonHelp, 1636
DisplayButtonNew, 1637
DisplayButtonSave, 1637
DisplayButtonSaveAs, 1637
DisplayComboBox, 1637
DisplayGridLines, 1637
DisplayLabeling, 1637
DisplayName(i), 2109
DisplayNumbers, 1637
DisplayOptions, 1638
DisplayStatusBar, 1638
DisplayTable, 1638
DoubleClickAction, 1638
DrawAlwaysInsideFrame, 2018
DrawEnhancedHTMLBrowser, 2110
DrawInsideFrame, 1639
DrawStylishButton, 2019
Drive, 1640
EdgeStyle, 1640
Editable(i), 2110
EditEnabled, 2111
EditOnFocus, 1642
Enabled, 1643
EnableEdit, 1644
EnableNavigateKeys, 1645
EndAngle, 1645
EndLeft, 1646
EndPoint, 2111
EndPointLeft, 2112
EndPointTop, 2112
EndStyle, 1646
EndTime(i), 2113
EndTop, 1647
EntryNameCaption, 1647
EvenRowBackColor, 1648
ExportDelimiter, 1648
ExportDirectoryChangeable, 1649
ExportDirectoryname, 1649
ExportFileExtension, 1650
ExportFilename, 1650
ExportFilenameChangeable, 1651
ExportFormatGuid, 1652
ExportFormatName, 1652
ExportParameters, 1653
ExportSelection, 1653
ExportShowDialog, 1654
ExtraSpaceForLabelDisplay, 2114
FieldLength, 1655, 1656
FillColorMode, 1656
FillPatternColor, 1656
FillStyle, 1658
Filter, 1658
FilterTag, 1658
FilterText, 1659
FirstConnectedObjectIndex, 1659

FirstConnectedObjectName, 1659
FitToLargest, 1660
FitToSize, 1660
FixedAspectRatio, 1660
FlashingColorOff, 1660
FlashingColorOn, 1662
FlashingEnabled, 1663
FlashingOnLimitViolation, 1664
FlashingRate, 1664
FlashTransparentColor, 1665
Flip, 1666
FocusColor, 1666
FocusWidth, 1667
Font, 1668
FontBold, 1669
FontItalic, 1670
FontName, 1671
FontSize, 1671
FontUnderline, 1672
ForeColor, 1673
ForeColorTransparency, 1674
FormatPattern, 1674
FormatPatternReadOnlySpecial, 2116
FormatType, 2022
Free, 1674
FreePercent, 1675
FreezeProviderConnections, 2117
GetSelectionText, 2117
Gradation, 1675
GraphDirection, 1676
GridBackColor, 2023
GridLineColor, 1676
GridlineFillColor, 1677
GridlineStyle, 1677
GridLineWidth, 1677
HeaderFont, 1678
Height, 1678
HelpText, 1683
HiddenInput, 1684
HideTagNames, 2117
HighLimitColor, 1684
Hitlist, 1685
HitlistColumnAdd, 1685
HitlistColumnCount, 1685
HitlistColumnIndex, 1686
HitlistColumnName, 1686
HitlistColumnRemove, 1687
HitlistColumnRepos, 1687
HitlistColumnSort, 1688
HitlistColumnSortIndex, 1688
HitlistColumnVisible, 1689
HitlistDefaultSort, 1689
HitlistMaxSourceItems, 1690
HitlistMaxSourceItemsWarn, 1690
HitlistRelTime, 1692
HitlistRelTimeFactor, 1691
HitlistRelTimeFactorType, 1691
HorizontalAlignment, 1692
HorizontalGridLines, 1693
HorizontalScrollBarPosition, 1694
Hotkey, 1694
HourNeedleHeight, 1694
HourNeedleWidth, 1695
IconSpace, 1695
Index, 1696
InnerBackColorOff, 1696
InnerBackColorOn, 1697
InputValue, 1698
InsertData(i), 2071
InsertEnable, 2024
IntegerDigits, 1698
Interval, 1699
ItemBorderStyle, 1699
JumpToLimitsAfterMouseClicked, 1700
LabelColor, 1700
Language, 1701
LargeTicksBold, 1702
LargeTicksSize, 1702
LastConnectedObjectIndex, 1702
LastConnectedObjectName, 1703
LeaveMarginForBorder, 2118
LeaveMarginForMarkers, 2119
Left, 1709
LeftOffset, 1709
Limit4LowerLimit, 1710
Limit4LowerLimitEnabled, 1711
Limit4LowerLimitRelative, 1712
Limit4LowerLimitColor, 1710
Limit4UpperLimit, 1712
Limit4UpperLimitColor, 1713
Limit4UpperLimitEnabled, 1714
Limit4UpperLimitRelative, 1714
Limit5LowerLimit, 1715
Limit5LowerLimitColor, 1715
Limit5LowerLimitEnabled, 1716
Limit5LowerLimitRelative, 1717
Limit5UpperLimit, 1717
Limit5UpperLimitColor, 1718
Limit5UpperLimitEnabled, 1718
Limit5UpperLimitRelative, 1719
LineColor, 1719
LineEdShapeStyle, 1720
LineWidth, 1721
LoadDataImmediately, 1722

LocalCursor, 1722
Localizable, 2079
LockSquaredExtent, 1723
LogOperation, 1723
LongTermArchiveConsistency, 1724
LowerLimit, 1725
LowerLimitColor(i), 2028
LowerLimitValue(i), 2028, 2029
LowLimitColor, 1725
MachineName, 1726
MarginToBorder, 1726
MaximumValue, 1727
MenuToolBarConfig, 1727
MessageBlockAlignment, 1728
MessageBlockAutoPrecisions, 1728
MessageBlockCaption, 1729
MessageBlockCount, 1729, 1732
MessageBlockDateFormat, 1730
MessageBlockExponentialFormat, 1730
MessageBlockFlashOn, 1731
MessageBlockHideText, 1731
MessageBlockHideTitleText, 1732
MessageBlockID, 1732
MessageBlockLeadingZeros, 1733
MessageBlockLength, 1733
MessageBlockName, 1734
MessageBlockPrecisions, 1734
MessageBlockSelected, 1735
MessageBlockShowDate, 1735
MessageBlockShowIcon, 1736
MessageBlockShowTitleIcon, 1736
MessageBlockTextID, 1737
MessageBlockTimeFormat, 1737
MessageBlockType, 1738
MessageColumnAdd, 1738
MessageColumnCount, 1739
MessageColumnIndex, 1739
MessageColumnName, 1740
MessageColumnRemove, 1740
MessageColumnRepos, 1741
MessageColumnSort, 1741
MessageColumnSortIndex, 1742
MessageColumnVisible, 1742
MessageListType, 1743
MinimumValue, 1743
MinuteNeedleHeight, 1744
MinuteNeedleWidth, 1744
Mode, 1745
Moveable, 1746
MsgFilterSQL, 1746
NeedleBorderColor, 1749
NeedleColor, 1750
NeedleFillStyle, 1751
NormalRangeColor, 1751
NormalRangeVisible, 1752
NumberOfValues(i), 2119
ObjectName, 2034
OCXState, 1755
Online, 1755
OperationSteps, 1756
OperatorMessageID, 1756
OperatorMessageIndex, 1757
OperatorMessageName, 1758
OperatorMessageNumber, 1758
OperatorMessageSelected, 1759
OperatorMessageSource1, 1759
OperatorMessageSource10, 1764
OperatorMessageSource2, 1760
OperatorMessageSource3, 1760
OperatorMessageSource4, 1761
OperatorMessageSource5, 1761
OperatorMessageSource6, 1762
OperatorMessageSource7, 1762
OperatorMessageSource8, 1763
OperatorMessageSource9, 1764
OperatorMessageType1, 1765
OperatorMessageType10, 1769
OperatorMessageType2, 1765
OperatorMessageType3, 1766
OperatorMessageType4, 1766
OperatorMessageType5, 1767
OperatorMessageType6, 1767
OperatorMessageType7, 1768
OperatorMessageType8, 1768
OperatorMessageType9, 1769
Orientation, 2036
PageMode, 1770
PageModeMessageNumber, 1770
Parent, 2120
Password, 1771
PersistentRTAuthorization, 2037
PersistentRTCSAuthorization, 2079
PersistentRuntime, 2038
PersistentRuntimeAuthorization, 2038
PersistentToDownload, 2080
PersistentToDownloadAuthorization, 2081
Picture, 1772
PictureAlignment, 1772
PictureDeactivated, 1773
PictureName, 2040
PictureOff, 1773
PictureOn, 1774
PointerColor, 1775
PointsCount, 1775

Precision, 1776
Pressed, 1776
PrintConfiguration, 2082
PrintJobName, 2042
PrintVisibleColumnsOnly, 2042
ProcessValue, 1777
ProjectPath, 2083
Radius, 1779
RadiusHeight, 1780
RadiusWidth, 1780
ReadOnly, 2043
RecipeName, 1781
RecipeNumber, 1781
RecordName, 1782
RecordNumber, 1782
RelativeFillLevel, 1782
Rotation, 1783
RotationAngle, 1784
RotationCenterLeft, 1784
RotationCenterTop, 1785
RoundCornerHeight, 1786
RoundCornerWidth, 1786
RowNumber, 2122
RowScrollbar, 1787
RowSizingEnable, 2083
RowTitles, 1820
RTPersistence, 1788
RTPersistencePasswordLevel, 2044
RTPersistenceType, 1788
RulerColor, 1789
RulerPrecision(i), 2122
RulerXPrecision(i), 2123
RulerYPrecision(i), 2124
ScaleColor, 1790
ScaleDenominator, 1795
ScaleGradation, 1790
ScaleLabelColor, 1791
ScaleNumerator, 1792
ScalePosition, 1792
ScaleTickColor, 1792
ScaleTickLabelPosition, 1793
ScaleTickLength, 1794
ScaleTickPosition, 1794
Scaling, 1795
ScalingType, 1795
ScreenItems, 1796
ScreenName, 1797
Screens, 1797
Scrollable, 2084
SecondNeedleHeight, 1798
SecondNeedleWidth, 1798
SegmentColoring, 1799
SelectBackColor, 1799
SelectedCellColor, 1800
SelectedCellForeColor, 1801
SelectedIndex, 1801
SelectedRowColor, 1802
SelectedRowForeColor, 1803
SelectedTitleColor, 1803
SelectedTitleForeColor, 1804
SelectForeColor, 1805
SelectionBackColor, 1806
SelectionColoring, 1806
SelectionForeColor, 1807
SelectionMode, 2045, 2046
SelectionRect, 1808
SelectionRectColor, 1808
SelectionRectWidth, 1809
SelectionType, 1810
SeparatorBackColor, 1810
SeparatorColor, 1811
SeparatorStyle, 1812
SeparatorWidth, 1812
SeperatorCornerStyle, 1812
ServerNames, 1813
ServerPrefix, 1814
ServerScale, 1813
SetpointTrendArchiveStartId(i), 2124
SetpointTrendColor(i), 2125
SetpointTrendNumberOfValues(i), 2126
Shared, 1814
ShareTimeColumn, 2127
ShareValueAxis, 2127
ShareXAxis, 2128
ShareYAxis, 2128
ShiftDecimalPoint, 1814
ShortenCellText, 2047
ShortenColumnHeaderText, 2047
ShowAlarmsFromDate, 1814
ShowBadTagState, 1815
ShowBar, 1816
ShowBorder, 2129
ShowCaption, 1816
ShowColumn(i), 2129
ShowCurve(i), 2130
ShowDecimalPoint, 1817
ShowFillLevel, 1817
ShowFocusRectangle, 1818
ShowInputControls, 2130
ShowLargeTicksOnly, 1818, 2027
ShowLimitMarkers, 1819
ShowMainFrame, 2131
ShowMessagesAtDate, 2084
ShowOverflowIndicator, 2131

ShowPeakValuePointer, 1819
 ShowPosition, 1820
 ShowRuler, 1821
 ShowRulerInAxis, 1821
 ShowScale, 1822
 ShowScrollbars, 1822
 ShowSetpointTrend(i), 2132
 ShowSortButton, 1823
 ShowSortIcon, 1823, 1824
 ShowStatusBar, 1824
 ShowTableGridlines, 1825
 ShowThumb, 1825
 ShowTickLabels, 1826
 ShowTicks, 1826
 ShowTitle, 1827
 ShowToolBar, 1827
 ShowTrendIcon, 1828
 ShowTrendIndicator, 1828
 ShowVerticalGridlines, 2048
 ShowXValuesExponential(i), 2048
 ShowYValuesExponential(i), 2049
 SmartTags, 1830
 Sort, 2085
 SortByTimeDirection, 1830
 SortByTimeEnable, 2050
 SortByTimeEnabled, 1830
 SortOnColumnHeaderClick, 2050
 SortSequence, 1831
 SortTimeAscending, 2051
 SortTimeEnable, 2051
 StartAngle, 1831
 StartLeft, 2052
 StartStyle, 1832
 StartTop, 1833
 StartValue, 1833
 StatusBarBackColor, 1834
 StatusBarElementAdd, 1834
 StatusBarElementAutoSize, 1835
 StatusBarElementCount, 1836
 StatusBarElementIconId, 1836
 StatusBarElementID, 1837
 StatusBarElementIndex, 1838
 StatusBarElementName, 1838
 StatusBarElementRemove, 1839
 StatusBarElementRename, 1840
 StatusBarElements, 1840
 StatusBarElementTooltipText, 1840
 StatusBarElementUserDefined, 1841
 StatusBarElementVisible, 1842
 StatusBarElementWidth, 1842
 StatusBarFontColor, 1843
 StatusBarShowArchiveName, 2132
 StatusBarShowColumn, 2133
 StatusBarShowRecord, 2133
 StatusBarShowRow, 2134
 StatusBarShowText, 2134
 StatusBarShowTooltips, 1844
 StatusBarText, 1844
 StatusBarUseBackColor, 1845
 StatusBarVisible, 1846
 Style, 1846
 StyleSettings, 1847, 2065
 SwapDimensionsWithOrientation, 2134
 SwapFirstWithLastConnection, 1847
 TableBackColor, 1848
 TableColor, 1849
 TableColor2, 1849
 TableFocusOnButtonCommand, 2085
 TableForeColor, 1850
 TableForeColor2, 1851
 TableGridLineColor, 1852
 TableHeaderBackColor, 1852
 TableHeaderForeColor, 1853
 TagPrefix, 1854
 Template, 1856
 Text, 1857
 TextList, 1857
 TextOff, 1858
 TextOn, 1858
 TextOrientation, 1859
 ThumbBackColor, 1860
 TicksColor, 1860
 TickStyle, 1861
 TimeAxisBeginTime(i), 1862
 TimeAxisEndTime, 1862
 TimeAxisLabel(i), 1863
 TimeAxisRange, 1863
 TimeAxisShowGridLines(i), 2052
 TimeAxisShowLargeIncrements(i), 2135
 TimeAxisShowSmallIncrements(i), 2135
 TimeAxisTimeFormat(i), 1863
 TimeBase, 1864
 TimeColumnActualize, 1865
 TimeColumnAdd, 1865
 TimeColumnAlignment, 1866
 TimeColumnAlignment(i), 1866
 TimeColumnBackColor, 1867
 TimeColumnBeginTime, 1867
 TimeColumnCaption, 1868
 TimeColumnCount, 1868
 TimeColumnDateFormat, 1869
 TimeColumnEndTime, 1869
 TimeColumnForeColor, 1870
 TimeColumnFormat(i), 2053

TimeColumnHideText, 1870
TimeColumnHideTitleText, 1871
TimeColumnIndex, 1871
TimeColumnLength, 1872
TimeColumnMeasurePoints, 1872
TimeColumnName, 1873
TimeColumnRangeType, 1873
TimeColumnRemove, 1874
TimeColumnRename, 1874
TimeColumnRepos, 1875
TimeColumnShowDate, 1875
TimeColumnShowIcon, 1876
TimeColumnShowTitleIcon, 1876
TimeColumnSort, 1877
TimeColumnSortIndex, 1877
TimeColumnTimeFormat, 1878
TimeColumnTimeRangeBase, 1878
TimeColumnTimeRangeFactor, 1879
TimeColumnUseValueColumnColors, 1879
TimeColumnVisible, 1880
TimeJumpColor(i), 2136
TimeJumpEnabled(i), 2137
TimeOverlapColor(i), 2137
TimeOverlapEnabled(i), 2138
TimeRangeBase(i), 2086
TimeRangeFactor(i), 2087
TitleColor, 2054
TitleCut, 1882
TitleDarkShadowColor, 1882
TitleForeColor, 1883
TitleGridLineColor, 1884
TitleLightShadowColor, 1884
TitleSort, 1885
TitleStyle, 1886
Toggle, 1886
Tolerance, 1887
ToleranceColor, 1887
ToleranceLowerLimit, 1888
ToleranceLowerLimitColor, 1889
ToleranceLowerLimitEnabled, 1889
ToleranceLowerLimitRelative, 1890
ToleranceUpperLimit, 1890
ToleranceUpperLimitColor, 1891
ToleranceUpperLimitEnabled, 1891
ToleranceUpperLimitRelative, 1892
ToolbarAlignment, 1893
ToolbarBackColor, 1893
ToolbarButtonActive, 1894
ToolbarButtonAdd, 1895
ToolbarButtonAuthorization, 1900
ToolbarButtonBeginGroup, 1895
ToolbarButtonCount, 1896
ToolbarButtonEnabled, 1897
ToolbarButtonHotKey, 1897
ToolbarButtonID, 1898
ToolbarButtonIndex, 1898
ToolbarButtonLocked, 1899
ToolbarButtonName, 1900
ToolbarButtonRemove, 1901
ToolbarButtonRename, 1901
ToolbarButtonRepos, 1902
ToolbarButtonTooltipText, 1903
ToolbarButtonUserDefined, 1903
ToolbarShowTooltips, 1904
ToolbarUseBackColor, 1904
ToolbarUseHotKeys, 1905
ToolbarVisible, 1906
ToolTipText, 1906
Top, 1907
TopOffset, 1910
Total, 1910
Transparency, 1911
TransparentColor, 1912
TransparentColorDeactivatedPicture, 1913
TransparentColorPictureOff, 1913
TransparentColorPictureOn, 1914
TrendActualize, 1915
TrendAdd, 1915
TrendBeginTime, 1916
TrendColor, 1916
TrendCount, 1917
TrendEndTime, 1917
TrendExtendedColorSet, 1918
TrendFill, 1918
TrendFillColor, 1919
TrendIndex, 1920
TrendIndicatorColor, 1920
TrendLabel, 1921
TrendLineStyle, 1921
TrendLineType, 1922
TrendLineWidth, 1923
TrendLowerLimit, 1923
TrendLowerLimitColor, 1924
TrendLowerLimitColoring, 1924
TrendMeasurePoints, 1925
TrendName, 1925
TrendPointColor, 1926
TrendPointStyle, 1926
TrendPointWidth, 1927
TrendProvider, 1927
TrendRangeType, 1928
TrendRemove, 1929
TrendRename, 1929
TrendRepos, 1930

TrendSelectTagNameX, 1930
 TrendSelectTagNameY, 1931
 TrendTag, 2055
 TrendTagNameX, 1931
 TrendTagNameY, 1932
 TrendTimeRangeBase, 1932
 TrendTimeRangeFactor, 1933
 TrendTrendWindow, 1933
 TrendUncertainColor, 1934
 TrendUncertainColoring, 1934
 TrendUpperLimit, 1935
 TrendUpperLimitColor, 1935
 TrendUpperLimitColoring, 1936
 TrendVisible, 1937
 TrendWindowAdd, 1937
 TrendWindowCoarseGrid, 1938
 TrendWindowCoarseGridColor, 1938
 TrendWindowFineGrid, 1939
 TrendWindowFineGridColor, 1939
 TrendWindowForegroundTrendGrid, 1940
 TrendWindowGridInTrendColor, 1940
 TrendWindowHorizontalGrid, 1941
 TrendWindowIndex, 1941
 TrendWindowName, 1942
 TrendWindowRemove, 1942
 TrendWindowRename, 1943
 TrendWindowRepos, 1943
 TrendWindowRulerColor, 1944
 TrendWindowRulerLayer, 1945
 TrendWindowRulerStyle, 1945
 TrendWindowRulerWidth, 1946
 TrendWindowSpacePortion, 1946
 TrendWindowVerticalGrid, 1947
 TrendWindowVisible, 1947
 TrendXAxis, 1948
 TrendYAxis, 1948
 Type, 2138
 UncertainStateColor(i), 2057
 UncertainStateEnabled(i), 2058
 Unit, 1949
 UnitColor, 1949
 UnitText, 1950
 UnitTop, 1950
 UpdateEnable, 2059
 UpperLimit, 1951
 UpperLimitColor(i), 2059
 UpperLimitEnabled(i), 2060
 UpperLimitValue(i), 2060
 UseAllServers, 2061
 UseBarBorderConstraints, 2139
 Used, 1951
 UseDesignColorSchema, 1952
 UseDesignShadowSettings, 1953
 UsedPercent, 1955
 UseEffectiveProcessValue, 2140
 UseExponentialFormat, 1955
 UseFlashTransparentColor, 1956
 UseGDI, 2140
 UseMeasurePoints(i), 2141
 UseMessageColor, 1956
 UseMultipleLimits, 2141
 UserArchiveNumberOfValues(i), 2144
 UserArchiveStartId(i), 2144
 UseScaleConstraints, 2142
 UseScaledBarBorder, 2142
 UseSelectedTitleColor, 1957
 UseSeparateDiagrams, 2143
 UseSimplePrecisionOffset, 2143
 UseTableColor2, 1957
 UseTagLimitColors, 1958
 UseTimeRange(i), 2087
 UseTransparentColor, 1958
 UseTransparentColorDeactivatedPicture, 1959
 UseTransparentColorPictureOff, 1959
 UseTransparentColorPictureOn, 1960
 UseTrendNameAsLabel, 1960
 UV_ColumnWidth_AKZ, 1961
 UV_ColumnWidth_Descriptor, 1961
 UV_ColumnWidth_InstallationDate, 1961
 UV_ColumnWidth_LADDR, 1961
 UV_ColumnWidth_Name, 1961
 UV_ColumnWidth_OKZ, 1961
 UV_ColumnWidth_OperationState, 1962
 UV_ColumnWidth_OrderID, 1962
 UV_ColumnWidth_ProfileID, 1962
 UV_ColumnWidth_Rack, 1962
 UV_ColumnWidth_Slot, 1962
 UV_ColumnWidth_SoftwareRevision, 1962
 UV_ColumnWidth_SpecificProfileData, 1962
 UV_ColumnWidth_State, 1963
 UV_ColumnWidth_Station, 1963
 UV_ColumnWidth_SubAddress, 1963
 UV_ColumnWidth_SubSlot, 1963
 UV_ColumnWidth_SubSystem, 1963
 UV_ColumnWidth_Type, 1963
 UV_ShowItem_AKZ, 1963
 UV_ShowItem_Descriptor, 1964
 UV_ShowItem_InstallationDate, 1964
 UV_ShowItem_LADDR, 1964
 UV_ShowItem_Name, 1964
 UV_ShowItem_OKZ, 1964
 UV_ShowItem_OperationState, 1964
 UV_ShowItem_OrderID, 1964
 UV_ShowItem_ProfileID, 1965

UV_ShowItem_Rack, 1965
UV_ShowItem_Slot, 1965
UV_ShowItem_SoftwareRevision, 1965
UV_ShowItem_SpecificProfileData, 1965
UV_ShowItem_State, 1965
UV_ShowItem_Station, 1965
UV_ShowItem_SubAddress, 1966
UV_ShowItem_SubSlot, 1966
UV_ShowItem_SubSystem, 1966
UV_ShowItem_Type, 1966
ValidateFormatPattern, 2088
ValueAxisAutorange(i), 1967
ValueAxisBegin(i), 2089
ValueAxisDecimalPrecision(i), 2062
ValueAxisEnd(i), 2089
ValueAxisGridLineInterval(i), 2146
ValueAxisLabel(i), 1968
ValueAxisLargeIncrementsSize(i), 2146
ValueAxisScalingType(i), 1968
ValueAxisShowGridLines(i), 2063
ValueAxisShowLargeIncrements(i), 2147
ValueAxisShowSmallIncrements(i), 2147
ValueAxisSmallIncrementSize(i), 2148
ValueColumnAlignment(i), 1969
ValueColumnPrecision(i), 2064
VerticalAlignment, 1969
VerticalGridLines, 1970
VerticalScrollBarPosition, 1971
ViewOnly, 1971
Visible, 1972
Warning, 1974
WarningColor, 1974
WarningLowerLimit, 1975
WarningLowerLimitColor, 1976
WarningLowerLimitEnabled, 1976
WarningLowerLimitRelative, 1977
WarningRangeColor, 1977
WarningRangeStart, 1978
WarningRangeVisible, 1979
WarningUpperLimit, 1979
WarningUpperLimitColor, 1980
WarningUpperLimitEnabled, 1980
WarningUpperLimitRelative, 1981
Width, 1981
WindowCloseEnabled, 1984
WindowMaximizeEnabled, 1984
WindowMovingEnabled, 1985
WindowOnTop, 1985
WindowSizingEnabled, 1986
WindowsStyle, 1986
XAxisAdd, 1987
XAxisAlignment, 1987
XAxisAutoPrecisions, 1988
XAxisAutoRange(i), 1988
XAxisBegin(i), 2065
XAxisBeginValue, 1989
XAxisColor, 1989
XAxisCount, 1990
XAxisDecimalPrecision(i), 2149
XAxisEnd(i), 2066
XAxisEndValue, 1990
XAxisExponentialFormat, 1991
XAxisGridLineInterval(i), 2149
XAxisIndex, 1991
XAxisInTrendColor, 1992
XAxisLabel(i), 1992
XAxisLargeIncrementSize(i), 2150
XAxisMode(i), 2150
XAxisName, 1993
XAxisPrecisions, 1993
XAxisRemove, 1994
XAxisRepos, 1994
XAxisScalingType(i), 1994
XAxisShowGridLines(i), 2066
XAxisShowLargeIncrements(i), 2151
XAxisShowSmallIncrements(i), 2151
XAxisSmallIncrementSize(i), 2152
XAxisTrendWindow, 1995
XAxisVisible, 1995
XDataLogTag(i), 2152
XOnlineTag(i), 2153
YAxisAdd, 1996
YAxisAlignment, 1996
YAxisAutoPrecisions, 1997
YAxisAutoRange, 1997
YAxisBeginValue, 1998
YAxisColor, 1998
YAxisCount, 1999
YAxisEndValue, 1999
YAxisExponentialFormat, 2000
YAxisIndex, 2000
YAxisInTrendColor, 2001
YAxisLabel, 2001
YAxisName, 2002
YAxisPrecisions, 2002
YAxisRemove, 2002
YAxisRename, 2003
YAxisRepos, 2003
YAxisScalingType, 2004
YAxisTrendWindow, 2004
YAxisVisible, 2005
YDataLogTag(i), 2154
YOnlineTag(i), 2154
ZeroPoint, 2005

Zoomable, 2090
 ZoomFactor, 2006
 Eigenschaften (VBS); "Name, 1747
 Eigenschaften (VBS); "Object, 1752
 Eigenschaften in VBS
 Comment, 2101
 ComputerName, 1624
 Context, 1625
 Instance, 1698
 SelText, 1799
 State, 1833
 UserName, 1961
 Eigenschaften in VBS/ScrollPositionX, 2045
 Eigenschaften in VBS/ScrollPositionY, 2045
 Eigenschaften in VBS:AngleAlpha, 2068
 Eigenschaften in VBS:AngleBeta, 2069
 Eigenschaften in VBS:Application, 1986, 2007
 Eigenschaften in VBS:Axe, 2069
 Eigenschaften in VBS:AxisSection, 2069
 Eigenschaften in VBS:BackBorderWidth, 2009
 Eigenschaften in VBS:BackColor2, 2094
 Eigenschaften in VBS:BackColor3, 2009
 Eigenschaften in VBS:BackFlashColorOff, 2009
 Eigenschaften in VBS:BackFlashColorOn, 2009
 Eigenschaften in VBS:Background, 2069
 Eigenschaften in VBS:BarDepth, 2069
 Eigenschaften in VBS:BarHeight, 2069
 Eigenschaften in VBS:BarWidth, 2096
 Eigenschaften in VBS:BaseX, 2070
 Eigenschaften in VBS:BaseY, 2070
 Eigenschaften in VBS:BorderColorBottom, 2096
 Eigenschaften in VBS:BorderColorTop, 2097
 Eigenschaften in VBS:BorderFlashColorOff, 2009
 Eigenschaften in VBS:BorderFlashColorOn, 2010
 Eigenschaften in
 VBS:BottomConnectedConnectionPointIndex, 2010
 Eigenschaften in
 VBS:BottomConnectedObjectName, 2010
 Eigenschaften in VBS:BoxAlignment, 2010
 Eigenschaften in VBS:BoxCount, 2010
 Eigenschaften in VBS:BoxType, 2010
 Eigenschaften in VBS:Button1Width, 2011
 Eigenschaften in VBS:Button2Width, 2011
 Eigenschaften in VBS:Button3Width, 2011
 Eigenschaften in VBS:Button4Width, 2011
 Eigenschaften in VBS:ButtonColor, 2011
 Eigenschaften in VBS:CheckAlarmHigh, 2013
 Eigenschaften in VBS:CheckAlarmLow, 2013
 Eigenschaften in VBS:CheckLimitHigh4, 2013
 Eigenschaften in VBS:CheckLimitHigh5, 2013
 Eigenschaften in VBS:CheckLimitLow4, 2013
 Eigenschaften in VBS:CheckLimitLow5, 2014
 Eigenschaften in VBS:CheckToleranceHigh, 2014
 Eigenschaften in VBS:CheckToleranceLow, 2014
 Eigenschaften in VBS:CheckWarningHigh, 2014
 Eigenschaften in VBS:CheckWarningLow, 2014
 Eigenschaften in VBS:ClearOnNew, 2015
 Eigenschaften in VBS:CloseButton, 2070
 Eigenschaften in VBS:ColorAlarmHigh, 2015
 Eigenschaften in VBS:ColorAlarmLow, 2015
 Eigenschaften in VBS:ColorBottom, 2099
 Eigenschaften in VBS:ColorChangeType, 2099
 Eigenschaften in VBS:ColorLimitHigh4, 2015
 Eigenschaften in VBS:ColorLimitHigh5, 2015
 Eigenschaften in VBS:ColorLimitLow4, 2016
 Eigenschaften in VBS:ColorLimitLow5, 2016
 Eigenschaften in VBS:ColorToleranceHigh, 2016
 Eigenschaften in VBS:ColorToleranceLow, 2016
 Eigenschaften in VBS:ColorTop, 2099
 Eigenschaften in VBS:ColorWarningHigh, 2016
 Eigenschaften in VBS:ColorWarningLow, 2017
 Eigenschaften in VBS:CurrentContext, 1629
 Eigenschaften in VBS>EditAtOnce, 2019
 Eigenschaften in VBS:Exponent, 2020
 Eigenschaften in VBS:ExtendedOperation, 2114
 Eigenschaften in
 VBS:ExtendedZoomingEnable, 1654
 Eigenschaften in VBS:FillColor, 2020
 Eigenschaften in VBS:Filling, 2021
 Eigenschaften in VBS:FillingIndex, 2021
 Eigenschaften in VBS:FillStyle2, 2021
 Eigenschaften in VBS:FlashBackColor, 2021
 Eigenschaften in VBS:FlashBorderColor, 2021
 Eigenschaften in VBS:FlashFlashPicture, 2115
 Eigenschaften in VBS:FlashForeColor, 2115
 Eigenschaften in VBS:FlashPicReferenced, 2115
 Eigenschaften in VBS:FlashPicture, 2115
 Eigenschaften in VBS:FlashRate, 2021
 Eigenschaften in VBS:FlashRateBackColor, 2022
 Eigenschaften in VBS:FlashRateBorderColor, 2115
 Eigenschaften in VBS:FlashRateFlashPic, 2116
 Eigenschaften in VBS:FlashRateForeColor, 2116
 Eigenschaften in VBS:ForeFlashColorOff, 2071
 Eigenschaften in VBS:ForeFlashColorOn, 2071
 Eigenschaften in VBS:Hysteresis, 2023
 Eigenschaften in VBS:HysteresisRange, 2024
 Eigenschaften in VBS:ItemBorderBackColor, 2024
 Eigenschaften in VBS:ItemBorderColor, 2024
 Eigenschaften in VBS:ItemBorderWidth, 2025
 Eigenschaften in VBS:LanguageSwitch, 2118
 Eigenschaften in VBS:Layer, 1705
 Eigenschaften in VBS:Layer00Checked, 2072
 Eigenschaften in VBS:Layer00Color, 2072
 Eigenschaften in VBS:Layer00Value, 2072

Eigenschaften in VBS:Layer01Checked, 2072
Eigenschaften in VBS:Layer01Color, 2072
Eigenschaften in VBS:Layer01Value, 2073
Eigenschaften in VBS:Layer02Checked, 2073
Eigenschaften in VBS:Layer02Color, 2073
Eigenschaften in VBS:Layer02Value, 2073
Eigenschaften in VBS:Layer03Checked, 2073
Eigenschaften in VBS:Layer03Color, 2074
Eigenschaften in VBS:Layer03Value, 2074
Eigenschaften in VBS:Layer04Checked, 2074
Eigenschaften in VBS:Layer04Color, 2074
Eigenschaften in VBS:Layer04Value, 2074
Eigenschaften in VBS:Layer05Checked, 2075
Eigenschaften in VBS:Layer05Color, 2075
Eigenschaften in VBS:Layer05Value, 2075
Eigenschaften in VBS:Layer06Checked, 2075
Eigenschaften in VBS:Layer06Color, 2075
Eigenschaften in VBS:Layer06Value, 2076
Eigenschaften in VBS:Layer07Checked, 2076
Eigenschaften in VBS:Layer07Color, 2076
Eigenschaften in VBS:Layer07Value, 2076
Eigenschaften in VBS:Layer08Checked, 2076
Eigenschaften in VBS:Layer08Color, 2077
Eigenschaften in VBS:Layer08Value, 2077
Eigenschaften in VBS:Layer09Checked, 2077
Eigenschaften in VBS:Layer09Color, 2077
Eigenschaften in VBS:Layer09Value, 2077
Eigenschaften in VBS:Layer10Checked, 2078
Eigenschaften in VBS:Layer10Color, 2078
Eigenschaften in VBS:Layer10Value, 2078
Eigenschaften in
VBS:LayerDeclutteringEnable, 1708
Eigenschaften in VBS:Layers, 1708
Eigenschaften in VBS:LeftComma, 2025
Eigenschaften in VBS:LightEffect, 2078
Eigenschaften in VBS:LimitHigh4, 2025
Eigenschaften in VBS:LimitHigh5, 2025
Eigenschaften in VBS:LimitLow4, 2025
Eigenschaften in VBS:LimitLow5, 2026
Eigenschaften in VBS:LimitMax, 2026
Eigenschaften in VBS:LimitMin, 2026
Eigenschaften in VBS:ListType, 2078
Eigenschaften in VBS:LockStatus, 2026
Eigenschaften in VBS:LockText, 2026
Eigenschaften in VBS:LockTextColor, 2027
Eigenschaften in VBS:Logging, 1723
Eigenschaften in VBS:LongStrokesBold, 2027
Eigenschaften in VBS:LongStrokesOnly, 2027
Eigenschaften in VBS:LongStrokesTextEach, 2028
Eigenschaften in VBS:Marker, 2030
Eigenschaften in VBS:Max, 2079
Eigenschaften in VBS:MaximizeButton, 2030
Eigenschaften in VBS:MCGUBackColorOff, 2030
Eigenschaften in VBS:MCGUBackColorOn, 2030
Eigenschaften in VBS:MCGUBackFlash, 2030
Eigenschaften in VBS:MCGUTextColorOff, 2031
Eigenschaften in VBS:MCGUTextColorOn, 2031
Eigenschaften in VBS:MCGUTextFlash, 2031
Eigenschaften in VBS:MCKOBackColorOff, 2031
Eigenschaften in VBS:MCKOBackColorOn, 2031
Eigenschaften in VBS:MCKOBackFlash, 2031
Eigenschaften in VBS:MCKOTextColorOff, 2032
Eigenschaften in VBS:MCKOTextColorOn, 2032
Eigenschaften in VBS:MCKOTextFlash, 2032
Eigenschaften in VBS:MCKQBackColorOff, 2032
Eigenschaften in VBS:MCKQBackColorOn, 2032
Eigenschaften in VBS:MCKQBackFlash, 2032
Eigenschaften in VBS:MCKQTextColorOff, 2033
Eigenschaften in VBS:MCKQTextColorOn, 2033
Eigenschaften in VBS:MCKQTextFlash, 2033
Eigenschaften in VBS:MCText, 2033
Eigenschaften in VBS:MessageClass, 2033
Eigenschaften in VBS:Min, 2119
Eigenschaften in VBS:NumberLines, 2033
Eigenschaften in
VBS:ObjectSizeDeclutteringEnable, 1753
Eigenschaften in
VBS:ObjectSizeDeclutteringMax, 1754
Eigenschaften in
VBS:ObjectSizeDeclutteringMin, 1755
Eigenschaften in VBS:OnTop, 2035
Eigenschaften in VBS:OperationMessage, 2035
Eigenschaften in VBS:OperationReport, 2036
Eigenschaften in VBS:OutputFormat, 2037
Eigenschaften in VBS:OutputValue, 2037
Eigenschaften in VBS:Path, 1771
Eigenschaften in VBS:PicDeactReferenced, 2039
Eigenschaften in VBS:PicDeactTransparent, 2039
Eigenschaften in
VBS:PicDeactUseTransColor, 2039
Eigenschaften in VBS:PicDownReferenced, 2081
Eigenschaften in VBS:PicDownTransparent, 2040
Eigenschaften in
VBS:PicDownUseTransColor, 2040
Eigenschaften in VBS:PicReferenced, 2081
Eigenschaften in VBS:PicTransColor, 2040
Eigenschaften in VBS:PictureUp, 2040
Eigenschaften in VBS:PicUpReferenced, 2122
Eigenschaften in VBS:PicUpTransparent, 2041
Eigenschaften in VBS:PicUpUseTransColor, 2041
Eigenschaften in VBS:PicUseTransColor, 2041
Eigenschaften in VBS:PointCount, 2041
Eigenschaften in VBS:Position, 2082
Eigenschaften in VBS:PredefinedAngles, 2082

- Eigenschaften in VBS:Process, 2042
 - Eigenschaften in VBS:QualityCode, 1778
 - Eigenschaften in VBS:ReferenceRotationLeft, 2043
 - Eigenschaften in VBS:ReferenceRotationTop, 2043
 - Eigenschaften in VBS:Relevant, 2043
 - Eigenschaften in VBS:RightComma, 2044
 - Eigenschaften in VBS:SameSize, 2044
 - Eigenschaften in VBS:ScaleTicks, 2044
 - Eigenschaften in VBS:ScrollBars, 2045
 - Eigenschaften in VBS:SelBGColor, 2045
 - Eigenschaften in VBS:SelTextColor, 2047
 - Eigenschaften in VBS:SignificantMask, 2050
 - Eigenschaften in VBS:Sizeable, 1829
 - Eigenschaften in VBS:SmallChange, 2132
 - Eigenschaften in VBS:TagName, 2052
 - Eigenschaften in VBS:Tags, 1855
 - Eigenschaften in VBS:TimeStamp, 1880
 - Eigenschaften in VBS:Titleline, 2054
 - Eigenschaften in VBS:ToleranceHigh, 2054
 - Eigenschaften in VBS:ToleranceLow, 2054
 - Eigenschaften in
 - VBS:TopConnectedConnectionPointIndex, 2054
 - Eigenschaften in
 - VBS:TopConnectedObjectName, 2055
 - Eigenschaften in VBS:Trend, 2055
 - Eigenschaften in VBS:TypeAlarmHigh, 2056
 - Eigenschaften in VBS:TypeAlarmLow, 2056
 - Eigenschaften in VBS:TypeLimitHigh4, 2056
 - Eigenschaften in VBS:TypeLimitHigh5, 2056
 - Eigenschaften in VBS:TypeLimitLow4, 2056
 - Eigenschaften in VBS:TypeLimitLow5, 2056
 - Eigenschaften in VBS:TypeToleranceHigh, 2057
 - Eigenschaften in VBS:TypeToleranceLow, 2057
 - Eigenschaften in VBS:TypeWarningHigh, 2057
 - Eigenschaften in VBS:TypeWarningLow, 2057
 - Eigenschaften in VBS:UnselBGColor, 2139
 - Eigenschaften in VBS:UnselTextColor, 2139
 - Eigenschaften in VBS:UpdateCycle, 2139
 - Eigenschaften in VBS:UserValue1, 2062
 - Eigenschaften in VBS:UserValue2, 2062
 - Eigenschaften in VBS:UserValue3, 2062
 - Eigenschaften in VBS:UserValue4, 2062
 - Eigenschaften in VBS:Value, 1966
 - Eigenschaften in VBS:WarningHigh, 2064
 - Eigenschaften in VBS:WarningLow, 2064
 - Eigenschaften in VBS:WindowBorder, 2065
 - Eigenschaften in VBS:ZeroPointValue, 2090
 - Eigenschaften(VBS)
 - AlarmID, 1555
 - Ellipse-Objekt, 1386
 - EllipseSegment-Objekt, 1389
 - EllipticalArc-Objekt, 1391
 - Enabled-Eigenschaft (VBS), 1643
 - EnableEdit-Eigenschaft (VBS), 1644
 - EnableNavigateKeys-Eigenschaft (VBS), 1645
 - EncodeEx, 1168
 - EndAngle-Eigenschaft (VBS), 1645
 - EndLeft-Eigenschaft (VBS), 1646
 - EndPoint-Eigenschaft (VBS), 2111
 - EndPointLeft-Eigenschaft (VBS), 2112
 - EndPointTop-Eigenschaft (VBS), 2112
 - EndStyle-Eigenschaft (VBS), 1646
 - EndTime(i)-Eigenschaft (VBS), 2113
 - EndTop-Eigenschaft (VBS), 1647
 - EntryNameCaption-Eigenschaft (VBS), 1647
 - ErfasseBenutzeraktion, 1195
 - ErhöheVariable, 71, 1187, 1244
 - EstablishPROFIsafe, 1169
 - EvenRowBackColor-Eigenschaft (VBS), 1648
 - exit-Funktion, 180
 - exp-Funktion, 177
 - Export, 2168
 - Rezeptur, 1172
 - ExportDataRecords, 1170
 - ExportDataRecordsWithChecksum, 1172
 - ExportDelimiter-Eigenschaft (VBS), 1648
 - ExportDirectoryChangeable-Eigenschaft (VBS), 1649
 - ExportDirectoryname-Eigenschaft (VBS), 1649
 - ExportFileExtension-Eigenschaft (VBS), 1650
 - ExportFilenameChangeable-Eigenschaft (VBS), 1651
 - ExportFilename-Eigenschaft (VBS), 1650
 - ExportFormatGuid-Eigenschaft (VBS), 1652
 - ExportFormatName-Eigenschaft (VBS), 1652
 - ExportiereDatensätze, 1170
 - ExportiereDatensätzeMitPrüfsumme, 1172
 - ExportiereImportiereBenutzerverwaltung, 1175, 1243
 - ExportImportUserAdministration, 1175, 1243
 - ExportParameters-Eigenschaft (VBS), 1653
 - ExportSelection-Eigenschaft (VBS), 1653
 - ExportShowDialog-Eigenschaft (VBS), 1654
 - ExtraSpaceForLabelDisplay-Eigenschaft (VBS), 2114
- F**
- fabs-Funktion, 177
 - fclose-Funktion, 179
 - feof-Funktion, 179
 - ferror-Funktion, 179
 - fflush-Funktion, 179
 - fgetc-Funktion, 179

fgetpos-Funktion, 179
fgets-Funktion, 179
FieldLength-Eigenschaft (VBS), 1655, 1656
FillColorMode-Eigenschaft (VBS), 1656
FillPatternColor-Eigenschaft (VBS), 1656
FillStyle-Eigenschaft (VBS), 1658
Filter-Eigenschaft (VBS), 1658
FilterTag-Eigenschaft (VBS), 1658
FilterText-Eigenschaft (VBS), 1659
FirstConnectedObjectIndex-Eigenschaft (VBS), 1659
FirstConnectedObjectName-Eigenschaft (VBS), 1659
FitToLargest-Eigenschaft (VBS), 1660
FitToSize-Eigenschaft (VBS), 1660
FixedAspectRatio-Eigenschaft (VBS), 1660
FlashingColorOff-Eigenschaft (VBS), 1660
FlashingColorOn-Eigenschaft (VBS), 1662
FlashingEnabled-Eigenschaft (VBS), 1663
FlashingOnLimitViolation-Eigenschaft (VBS), 1664
FlashingRate-Eigenschaft (VBS), 1664
FlashTransparentColor-Eigenschaft (VBS), 1665
Flip-Eigenschaft (VBS), 1666
floor-Funktion, 177
fmod-Funktion, 177
FocusColor-Eigenschaft (VBS), 1666
FocusWidth-Eigenschaft (VBS), 1667
FontBold-Eigenschaft (VBS), 1669
Font-Eigenschaft (VBS), 1668
FontItalic-Eigenschaft (VBS), 1670
FontName-Eigenschaft (VBS), 1671
FontSize-Eigenschaft (VBS), 1671
FontUnderline-Eigenschaft (VBS), 1672
fopen-Funktion, 179
ForeColor-Eigenschaft (VBS), 1673
ForeColorTransparency-Eigenschaft (VBS), 1674
FormatPattern-Eigenschaft (VBS), 1674
FormatPatternReadOnlySpecial-Eigenschaft (VBS), 2116
FormatType-Eigenschaft (VBS), 2022
fputc-Funktion, 179
fputs-Funktion, 179
fread-Funktion, 180
Free-Eigenschaft (VBS), 1674
free-Funktion, 180
FreePercent-Eigenschaft (VBS), 1675
FreezeProviderConnections-Eigenschaft (VBS), 2117
freopen-Funktion, 179
frexp-Funktion, 177
fseek-Funktion, 179
fsetpos-Funktion, 179

ftell-Funktion, 179
FunctionTrendControl-Objekt, 1398
fwrite-Funktion, 180

G

GAPIEnumTypeMembersExStr (Control Center), 433
Gauge-Objekt, 1406
GeheZumAnfang, 1183
GeheZumEnde, 1182
GetBrightness, 1176
getc-Funktion, 179
GetColumn, 2169, 2188, 2189, 2249, 2250, 2266, 2267
GetColumnCollection, 2170
GetDataRecordFromPLC, 1176
GetDataRecordName, 1178
GetDataRecordTagsFromPLC, 1180
getenv-Funktion, 180
GetGroupNumber, 1180
GetHitlistColumn, 2172
GetHitlisteColumnCollection, 2171
GetLanguageByLocaleID, 34
GetLinkedTag, 33
GetLocalScreen, 31
GetMessageBlock, 2173
GetMessageBlockCollection, 2174
GetMessageColumn, 2175
GetMessageColumnCollection, 2177, 2181
GetOperatorMessage, 2176
GetOperatorMessageCollection, 2179
GetParentScreen, 36
GetParentScreenWindow, 37
GetPassword, 1181
GetPropBOOL, 39
GetPropChar, 41
GetPropDouble, 42
GetPropLong, 43
GetRow, 2180
GetRulerBlock, 2182
GetRulerBlockCollection, 2183
GetRulerColumn, 2185
GetRulerColumnCollection, 2186
GetRulerData, 2187
GetSelectionText-Eigenschaft (VBS), 2117
GetStatisticAreaColumn, 2191
GetStatisticAreaColumnCollection, 2192
GetStatisticResultColumn, 2193
GetStatisticResultColumnCollection, 2194
GetStatusBarElement, 2195
GetStatusBarElementCollection, 2196

GetTagBit, 47
GetTagBitState, 54
GetTagBitStateQC, 56
GetTagBitStateQCWait, 59
GetTagBitStateWait, 61
GetTagBitWait-Funktion, 69
GetTagByte, 47
GetTagByteState, 54
GetTagByteStateQC, 56
GetTagByteStateQCWait, 59
GetTagByteStateWait, 61
GetTagByteWait-Funktion, 69
GetTagChar, 47
GetTagCharState, 54
GetTagCharStateQC, 56
GetTagCharStateQCWait, 59
GetTagCharStateWait, 61
GetTagCharWait-Funktion, 69
GetTagDateTime, 47, 48
GetTagDouble, 47
GetTagDoubleState, 54
GetTagDoubleStateQC, 56
GetTagDoubleStateQCWait, 59
GetTagDoubleStateWait, 61
GetTagDoubleWait-Funktion, 69
GetTagDWord-Funktion, 47
GetTagDWordState, 54
GetTagDWordStateQC, 56
GetTagDWordStateQCWait, 59
GetTagDWordStateWait, 62
GetTagDWordWait-Funktion, 69
GetTagFloat, 47
GetTagFloatState, 54
GetTagFloatStateQC, 56
GetTagFloatStateQCWait, 59
GetTagFloatStateWait, 62
GetTagFloatWait-Funktion, 69
GetTagMultiStateQCWait-Funktion, 49
GetTagMultiStateWait, 51
GetTagMultiWait, 52
GetTagRaw, 47
GetTagRawState, 54
GetTagRawStateQC, 56
GetTagRawStateQCWait, 59
GetTagRawStateWait, 62
GetTagRawWait-Funktion, 69
GetTagSByte, 47
GetTagSByteState, 54
GetTagSByteStateQC, 56
GetTagSByteStateQCWait, 59
GetTagSByteStateWait, 62
GetTagSByteWait-Funktion, 69
GetTagSDWord-Funktion, 47
GetTagSDWordState, 54
GetTagSDWordStateQC, 56
GetTagSDWordStateQCWait, 59
GetTagSDWordStateWait, 62
GetTagSDWordWait-Funktion, 70
GetTagSWord, 47
GetTagSWordState, 54
GetTagSWordStateQC, 57
GetTagSWordStateQCWait, 59
GetTagSWordStateWait, 62
GetTagSWordWait-Funktion, 70
GetTagValue, 63
GetTagValueStateQC, 65
GetTagValueStateQCWait, 66
GetTagValueWait, 68
GetTagWord, 47
GetTagWordState, 54
GetTagWordStateQC, 57
GetTagWordStateQCWait, 59
GetTagWordStateWait, 62
GetTagWordWait-Funktion, 70
GetTimeAxis, 2198
GetTimeAxisCollection, 2199
GetTimeColumn, 2201
GetTimeColumnCollection, 2202
GetToolBarButton, 2203
GetToolBarButtonCollection, 2205
GetTrend, 2206
GetTrendCollection, 2207
GetTrendWindow, 2208
GetTrendWindowCollection, 2209
GetUserName, 1182
GetValueAxis, 2210
GetValueAxisCollection, 2212
GetValueColumn, 2213
GetValueColumnCollection, 2214
GetXAxis, 2216
GetXAxisCollection, 2217
GetYAxis, 2218
GetYAxisCollection, 2219
gmtime-Funktion, 182
GoToEnd, 1182
GoToHome, 1183
Gradation-Eigenschaft (VBS), 1675
GraphDirection-Eigenschaft (VBS), 1676
GraphicIOField-Objekt, 1409
GraphicView-Objekt, 1412
GridBackColor-Eigenschaft (VBS), 2023
GridLineColor-Eigenschaft (VBS), 1676
GridlineFillColor-Eigenschaft (VBS), 1677
GridLineStyle-Eigenschaft (VBS), 1677

GridLineWidth-Eigenschaft (VBS), 1677
Group-Objekt, 1415

H

HardwareSicherEntfernen, 1206
HeaderFont-Eigenschaft (VBS), 1678
Height-Eigenschaft (VBS), 1678
HelpText-Eigenschaft (VBS), 1683
HiddenInput-Eigenschaft (VBS), 1684
HideAlarm, 2221
HideTagName-Eigenschaft (VBS), 2117
HighLimitColor-Eigenschaft (VBS), 1684
HitlistColumnAdd-Eigenschaft (VBS), 1685
HitlistColumnCount-Eigenschaft (VBS), 1685
HitlistColumnIndex-Eigenschaft (VBS), 1686
HitlistColumnName-Eigenschaft (VBS), 1686
HitlistColumnRemove-Eigenschaft (VBS), 1687
HitlistColumnRepos-Eigenschaft (VBS), 1687
HitlistColumnSort-Eigenschaft (VBS), 1688
HitlistColumnSortIndex-Eigenschaft (VBS), 1688
HitlistColumnVisible-Eigenschaft (VBS), 1689
HitlistDefaultSort-Eigenschaft (VBS), 1689
Hitlist-Eigenschaft (VBS), 1685
HitlistMaxSourceItems-Eigenschaft (VBS), 1690
HitlistMaxSourceItemsWarn-Eigenschaft (VBS), 1690
HitlistRelTime-Eigenschaft (VBS), 1692
HitlistRelTimeFactor-Eigenschaft (VBS), 1691
HitlistRelTimeFactorType-Eigenschaft (VBS), 1691
HMIRuntime, 1295
HMIRuntime-Objekt, 1295, 1313
HorizontalAlignment-Eigenschaft (VBS), 1692
HorizontalGridLines-Eigenschaft (VBS), 1693
HorizontalScrollBarPosition-Eigenschaft (VBS), 1694
Hotkey-Eigenschaft (VBS), 1694
HourNeedleHeight-Eigenschaft (VBS), 1694
HourNeedleWidth-Eigenschaft (VBS), 1695

I

IconSpace-Eigenschaft (VBS), 1695
ImportDataRecords, 1183
ImportDataRecordsWithChecksum, 1185
ImportiereDatenätzeMitPrüfsumme, 1185
ImportiereDatensätze, 1183
IncreaseTag, 71, 1187, 1244
Index-Eigenschaft (VBS), 1696
InnerBackColorOff-Eigenschaft (VBS), 1696
InnerBackColorOn-Eigenschaft (VBS), 1697

InputValue-Eigenschaft (VBS), 1698
InquireLanguage, 72
InsertData(i)-Eigenschaft (VBS), 2071
InsertEnable-Eigenschaft (VBS), 2024
IntegerDigits-Eigenschaft (VBS), 1698
Interval-Eigenschaft (VBS), 1699
InverseLinearScaling, 74, 1187, 1245
InvertBit, 76, 1189, 1247
InvertBitInTag, 77, 1190, 1248
InvertiereBit, 76, 1189, 1247
InvertiereBitInVariable, 77, 1190, 1248
InvertiereLineareSkalierung, 74, 1187, 1245
IOField-Objekt, 1418
isalnum-Funktion, 175
isalpha-Funktion, 175
isdigit-Funktion, 175
isgraph-Funktion, 175
islower-Funktion, 175
isprint-Funktion, 175
ispunct-Funktion, 175
isspace-Funktion, 175
isupper-Funktion, 175
IsUserAuthorized, 78
isxdigit-Funktion, 175
ItemBorderStyle-Eigenschaft (VBS), 1699
Item-Methode, 2222
Item-Objekt, 1314

J

JumpToLimitsAfterMouseClicked-Eigenschaft (VBS), 1700

K

KalibriereTouchscreen, 1155

L

LabelColor-Eigenschaft (VBS), 1700
labs-Funktion, 180
LadeDatensatz, 1192
Language-Eigenschaft (VBS), 1701
LargeTicksBold-Eigenschaft (VBS), 1702
LargeTicksSize-Eigenschaft (VBS), 1702
LastConnectedObjectIndex-Eigenschaft (VBS), 1702
LastConnectedObjectName-Eigenschaft (VBS), 1703
Layer-Objekt, 1315
Layers-Objekt (Auflistung), 1317
Ldexp-Funktion, 177

Idiv-Funktion, 180
 LeaveMarginForBorder-Eigenschaft (VBS), 2118
 LeaveMarginForMarkers-Eigenschaft (VBS), 2119
 Left-Eigenschaft (VBS), 1709
 LeftOffset-Eigenschaft (VBS), 1709
 LeseBenutzername, 1182
 LeseDatensatzAusSteuerung, 1176
 LeseDatensatzname, 1178
 LeseDatensatzvariablenAusSteuerung, 1180
 LeseGruppennummer, 1180
 LeseHelligkeit, 1176
 LeseKennwort, 1181
 Limit4LowerLimitColor-Eigenschaft (VBS), 1710
 Limit4LowerLimit-Eigenschaft (VBS), 1710
 Limit4LowerLimitEnabled-Eigenschaft (VBS), 1711
 Limit4LowerLimitRelative-Eigenschaft (VBS), 1712
 Limit4UpperLimitColor-Eigenschaft (VBS), 1713
 Limit4UpperLimit-Eigenschaft (VBS), 1712
 Limit4UpperLimitEnabled-Eigenschaft (VBS), 1714
 Limit4UpperLimitRelative-Eigenschaft (VBS), 1714
 Limit5LowerLimitColor-Eigenschaft (VBS), 1715
 Limit5LowerLimit-Eigenschaft (VBS), 1715
 Limit5LowerLimitEnabled-Eigenschaft (VBS), 1716
 Limit5LowerLimitRelative-Eigenschaft (VBS), 1717
 Limit5UpperLimitColor-Eigenschaft (VBS), 1718
 Limit5UpperLimit-Eigenschaft (VBS), 1717
 Limit5UpperLimitEnabled-Eigenschaft (VBS), 1718
 Limit5UpperLimitRelative-Eigenschaft (VBS), 1719
 LineareSkalierung, 79, 1191, 1249
 LinearScaling, 79, 1191, 1249
 LineColor-Eigenschaft (VBS), 1719
 LineEndShapeStyle-Eigenschaft (VBS), 1720
 Line-Objekt, 1422
 LineWidth-Eigenschaft (VBS), 1721
 Listbox, 1424
 LoadDataImmediately-Eigenschaft (VBS), 1722
 LoadDataRecord, 1192
 LocalCursor-Eigenschaft (VBS), 1722
 Localizable-Eigenschaft (VBS), 2079
 localtime-Funktion, 182
 LockAlarm, 2223
 LockSquaredExtent-Eigenschaft (VBS), 1723
 LoescheArchiv, 1162
 log10-Funktion, 177
 log-Funktion, 177
 Logging-Objekt, 1318
 Logoff, 1193
 Logon, 1194
 LogOperation-Eigenschaft (VBS), 1723
 LongTermArchiveConsistency-Eigenschaft (VBS), 1724
 LoopInAlarm, 2224

LöscheDatensatz, 1159
 LöscheDatensatzSpeicher, 1161
 LöscheMeldepuffer, 1158
 LöscheMeldepufferProTool, 1159
 LowerLimitColor(i)-Eigenschaft (VBS), 2028
 LowerLimit-Eigenschaft (VBS), 1725
 LowerLimitValue(i)-Eigenschaft (VBS), 2028, 2029
 LowLimitColor-Eigenschaft (VBS), 1725

M

MachineName-Eigenschaft (VBS), 1726
 malloc-Funktion, 180
 MarginToBorder-Eigenschaft (VBS), 1726
 math-Funktionen

- acos-Funktion, 176, 177
- asin-Funktion, 176
- atan-Funktion, 176
- ceil-Funktion, 177
- cos-Funktion, 177
- cosh-Funktion, 177
- exp-Funktion, 177
- fabs-Funktion, 177
- floor-Funktion, 177
- fmod-Funktion, 177
- frexp-Funktion, 177
- ldexp-Funktion, 177
- log10-Funktion, 177
- log-Funktion, 177
- modf-Funktion, 177
- pow-Funktion, 177
- sin-Funktion, 177
- sinh-Funktion, 177
- sqrt-Funktion, 177
- tan-Funktion, 177
- tanh-Funktion, 177

 MaximumValue-Eigenschaft (VBS), 1727
 MediaControl, 1427
 memchr-Funktion, 177
 memcmp-Funktion, 177
 memcpy-Funktion, 177
 memmove-Funktion, 177
 memory-Funktionen

- memchr-Funktion, 177
- memcmp-Funktion, 177
- memcpy-Funktion, 177
- memmove-Funktion, 177
- memset-Funktion, 177

 memset-Funktion, 177
 MenuToolBarConfig-Eigenschaft (VBS), 1727
 MessageBlockAlignment-Eigenschaft (VBS), 1728

MessageBlockAutoPrecisions-Eigenschaft (VBS), 1728
MessageBlockCaption-Eigenschaft (VBS), 1729
MessageBlockCount -Eigenschaft (VBS), 1732
MessageBlockCount-Eigenschaft (VBS), 1729
MessageBlockDateFormat-Eigenschaft (VBS), 1730
MessageBlockExponentialFormat-Eigenschaft (VBS), 1730
MessageBlockFlashOn-Eigenschaft (VBS), 1731
MessageBlockHideText-Eigenschaft (VBS), 1731
MessageBlockHideTitleText-Eigenschaft (VBS), 1732
MessageBlockID-Eigenschaft (VBS), 1732
MessageBlockLeadingZeros-Eigenschaft (VBS), 1733
MessageBlockLength-Eigenschaft (VBS), 1733
MessageBlockName-Eigenschaft (VBS), 1734
MessageBlockPrecisions-Eigenschaft (VBS), 1734
MessageBlockSelected-Eigenschaft (VBS), 1735
MessageBlockShowDate-Eigenschaft (VBS), 1735
MessageBlockShowIcon-Eigenschaft (VBS), 1736
MessageBlockShowTitleIcon-Eigenschaft (VBS), 1736
MessageBlockTextId-Eigenschaft (VBS), 1737
MessageBlockTimeFormat-Eigenschaft (VBS), 1737
MessageBlockType-Eigenschaft (VBS), 1738
MessageColumnAdd-Eigenschaft (VBS), 1738
MessageColumnCount-Eigenschaft (VBS), 1739
MessageColumnIndex-Eigenschaft (VBS), 1739
MessageColumnName-Eigenschaft (VBS), 1740
MessageColumnRemove-Eigenschaft (VBS), 1740
MessageColumnRepos-Eigenschaft (VBS), 1741
MessageColumnSort-Eigenschaft (VBS), 1741
MessageColumnSortIndex-Eigenschaft (VBS), 1742
MessageColumnVisible-Eigenschaft (VBS), 1742
MessageListType-Eigenschaft (VBS), 1743
MessageView-Objekt, 1350
Methoden, 2158, 2161, 2163, 2164, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2179, 2180, 2181, 2182, 2183, 2185, 2186, 2187, 2188, 2189, 2191, 2192, 2193, 2194, 2195, 2196, 2198, 2199, 2201, 2202, 2203, 2205, 2206, 2207, 2208, 2209, 2210, 2212, 2213, 2214, 2216, 2217, 2218, 2219, 2221, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2271, 2272, 2273, 2274
Methoden (VBS), 2155, 2164
 Activate, 2155
 CreateTagSet, 2164
Item, 2222
Stop, 2264
Trace, 2265
Methoden in VBS
 Create, 2163
 DeactivateDynamic, 2165
Methoden:Add, 2160
Methoden:Read, 2236
Methoden:Refresh, 2240
Methoden:Remove, 2241
Methoden:RemoveAll, 2244
Methoden:Restore, 2246
Methoden:Write, 2268
MinimumValue-Eigenschaft (VBS), 1743
MinuteNeedleHeight-Eigenschaft (VBS), 1744
MinuteNeedleWidth-Eigenschaft (VBS), 1744
mktime-Funktion, 182
Mode-Eigenschaft (VBS), 1745
modf-Funktion, 177
Moveable-Eigenschaft (VBS), 1746
MoveAxis, 2224
MoveToFirst, 2225
MoveToFirstLine, 2225
MoveToFirstPage, 2226
MoveToLast, 2226
MoveToLastLine, 2227
MoveToLastPage, 2227
MoveToNext, 2228
MoveToNextPage, 2229
MoveToPrevious, 2229
MoveToPreviousLine, 2230
MoveToPreviousPage, 2230
MsgFilterSQL-Eigenschaft (VBS), 1746
multibyte-Funktionen
 _ismbcalnum-Funktion, 178
 _ismbcalpha-Funktion, 178
 _ismbcdigit-Funktion, 178
 _ismbcgraph-Funktion, 178
 _ismbcclower-Funktion, 178
 _ismbcprint-Funktion, 178
 _ismbcpunct-Funktion, 178
 _ismbcspace-Funktion, 178
 _ismbcupper-Funktion, 178
 _mbscat-Funktion, 178
 _mbschr-Funktion, 178
 _mbscmp-Funktion, 178
 _mbscpy-Funktion, 178
 _mbscspn-Funktion, 178
 _mbsdec-Funktion, 178
 _mbsinc-Funktion, 178
 _mbslen-Funktion, 178
 _mbsncat-Funktion, 178

_mbsncmp-Funktion, 178
 _mbsncpy-Funktion, 178
 _mbspbrk-Funktion, 178
 _mbsrchr-Funktion, 178
 _mbsspn-Funktion, 178
 _mbsstr-Funktion, 178
 _mbstok-Funktion, 178
 _mbstrlen-Funktion, 178
 MultiLineEdit, 1429

N

Name-Eigenschaft (VBS), 1747
 NColumnTextDate-Eigenschaft (VBS), 1621
 NeedleBorderColor-Eigenschaft (VBS), 1749
 NeedleColor-Eigenschaft (VBS), 1750
 NeedleFillStyle-Eigenschaft (VBS), 1751
 NextColumn, 2231
 NextTrend, 2231
 NormalRangeColor-Eigenschaft (VBS), 1751
 NormalRangeVisible-Eigenschaft (VBS), 1752
 NotifyUserAction, 1195
 NumberOfValues(i)-Eigenschaft (VBS), 2119

O

Object-Eigenschaft (VBS), 1752
 ObjectName-Eigenschaft (VBS), 2034
 Objekte (VBS)

- ActiveScreenItem, 1548, 1549
- AlarmControl-Objekt, 1338
- ApplicationWindow, 1479
- Bar, 1354
- Battery, 1359
- BrowserView, 1417
- Button, 1360
- ChannelDiagnose, 1364
- Check-Box, 1366
- Circle, 1369
- CircleSegment, 1372
- CircularArc, 1375
- Clock, 1377
- Connector, 1379
- DateTimeField, 1382
- DiskSpaceView, 1384
- Ellipse-Objekt, 1386
- EllipseSegment, 1389
- EllipticalArc, 1391
- FunctionTrendControl, 1398
- Gauge, 1406
- GraphicIOField, 1409

GraphicView, 1412
 Group-Objekt, 1415
 HMIRuntime, 1295
 IOField-Objekt, 1418
 Line, 1422
 Listbox, 1424
 MessageView, 1350
 MultiLineEdit, 1429
 OLEView-Objekt, 1432
 OnlineTrend Control, 1443
 OptionGroup, 1453
 Polygon, 1456
 Polyline, 1459
 ProjectName-Objekt, 1461
 ProtectedAreaName, 1463
 RangeLabelView, 1464
 RangeQualityView, 1466
 Rectangle, 1473
 RoundButton, 1474
 Screen, 1297, 1321
 ScreenItem, 1298, 1323
 ScreenItems (Auflistung), 1300, 1325
 Screens (Auflistung), 1296
 ScreenWindow, 1477
 Slider, 1483
 SmartClientView-Objekt, 1484
 SmartTag, 1302, 1329
 SmartTags (Auflistung), 1301, 1331
 StatusForce, 1486
 Switch, 1488
 SymbolicIOField, 1491
 SymbolLibrary, 1496
 SystemDiagnoseView, 1498
 SystemDiagnoseWindow, 1501
 TableView-Objekt, 1434
 TextField, 1505
 TrendRulerControl, 1508
 TrendView, 1516
 TubeArcObject, 1520
 TubeDoubleTeeObject, 1522
 TubePolyline, 1524
 TubeTeeObject, 1526
 UserArchiveControl-Objekt, 1528
 UserView, 1536
 WindowSlider, 1539
 WLanQualityView, 1541
 ZoneLabelView, 1542
 ZoneQualityView, 1543
 Objekte in VBS

- Alarm-Objekt, 1304

 Objekte:Alarmlogs, 1306
 Objekte:DataItem, 1308

Objekte:DataLogs, 1309
Objekte:HMIRuntime, 1313
Objekte:Item, 1314
Objekte:Layer, 1315
Objekte:Logging, 1318
Objekte:Tag, 1332
OCXState-Eigenschaft (VBS), 1755
ÖffneAlleArchive, 1197
ÖffneBildschirmtastatur, 1200
ÖffneDialogSystemsteuerung, 1198
ÖffneEingabeaufforderung, 1198
ÖffneInternetExplorer, 1199
ÖffneTaskManager, 1201
OLEView-Objekt, 1432
OneToOneView, 2232
Online-Eigenschaft (VBS), 1755
OnlineTrendControl, 1443
OpenAllLogs, 1197
OpenCommandPrompt, 1198
OpenControlPanelDialog, 1198
OpenInternetExplorer, 1199
OpenScreenKeyboard, 1200
OpenTaskManager, 1201
OperationSteps-Eigenschaft (VBS), 1756
OperatorMessageID-Eigenschaft (VBS), 1756
OperatorMessageIndex-Eigenschaft (VBS), 1757
OperatorMessageName-Eigenschaft (VBS), 1758
OperatorMessageNumber-Eigenschaft (VBS), 1758
OperatorMessageSelected-Eigenschaft (VBS), 1759
OperatorMessageSource10-Eigenschaft (VBS), 1764
OperatorMessageSource1-Eigenschaft (VBS), 1759
OperatorMessageSource2-Eigenschaft (VBS), 1760
OperatorMessageSource3-Eigenschaft (VBS), 1760
OperatorMessageSource4-Eigenschaft (VBS), 1761
OperatorMessageSource5-Eigenschaft (VBS), 1761
OperatorMessageSource6-Eigenschaft (VBS), 1762
OperatorMessageSource7-Eigenschaft (VBS), 1762
OperatorMessageSource8-Eigenschaft (VBS), 1763
OperatorMessageSource9-Eigenschaft (VBS), 1764
OperatorMessageType10-Eigenschaft (VBS), 1769
OperatorMessageType1-Eigenschaft (VBS), 1765
OperatorMessageType2-Eigenschaft (VBS), 1765
OperatorMessageType3-Eigenschaft (VBS), 1766
OperatorMessageType4-Eigenschaft (VBS), 1766
OperatorMessageType5-Eigenschaft (VBS), 1767
OperatorMessageType6-Eigenschaft (VBS), 1767
OperatorMessageType7-Eigenschaft (VBS), 1768
OperatorMessageType8-Eigenschaft (VBS), 1768
OperatorMessageType9-Eigenschaft (VBS), 1769
OptionGroup-Objekt, 1453
Orientation-Eigenschaft (VBS), 2036

P

PageDown, 1201
PageMode-Eigenschaft (VBS), 1770
PageModeMessageNumber-Eigenschaft (VBS), 1770
PageUp, 1202
Parent-Eigenschaft (VBS), 2120
Password-Eigenschaft (VBS), 1771
PasteRows, 2232
PersistentRTAuthorization-Eigenschaft (VBS), 2037
PersistentRTCSAuthorization-Eigenschaft (VBS), 2079
PersistentRuntimeAuthorization-Eigenschaft (VBS), 2038
PersistentRuntime-Eigenschaft (VBS), 2038
PersistentToDownloadAuthorization-Eigenschaft (VBS), 2081
PersistentToDownload-Eigenschaft (VBS), 2080
PictureAlignment-Eigenschaft (VBS), 1772
PictureDeactivated-Eigenschaft (VBS), 1773
Picture-Eigenschaft (VBS), 1772
PictureOff-Eigenschaft (VBS), 1773
PictureOn-Eigenschaft (VBS), 1774
PLC-Code-Anzeige
 OpenViewerIECPLByAssignment, 1132
 OpenViewerIECPLByCall, 1126
 OpenViewerIECPLByFCCall, 1129
 OpenViewerS7GraphByBlock, 1123
PointerColor-Eigenschaft (VBS), 1775
PointsCount-Eigenschaft (VBS), 1775
Polygon-Objekt, 1456
Polyline-Objekt, 1459
pow-Funktion, 177
Precision-Eigenschaft (VBS), 1776
Pressed-Eigenschaft (VBS), 1776
PreviousColumn, 2233
PreviousTrend, 2233
Print, 2234
PrintConfiguration-Eigenschaft (VBS), 2082
PrintJobName-Eigenschaft (VBS), 2042
PrintReport, 1203
PrintScreen, 1203
PrintVisibleColumnsOnly-Eigenschaft (VBS), 2042
ProcessValue-Eigenschaft (VBS), 1777
PROFIsafeTrennen, 1237
PROFIsafeVerbinden, 1169
ProjectName-Object, 1461
ProjectPath-Eigenschaft (VBS), 2083
ProtectedAreaName-Objekt, 1463
putc-Funktion, 179

Q

qsort-Funktion, 181
 Quality Codes von Variablen, 198
 QuitHorn, 2234
 QuitSelected, 2235
 QuittiereMeldung, 1149
 QuitVisible, 2235

R

Radius-Eigenschaft (VBS), 1779
 RadiusHeight-Eigenschaft (VBS), 1780
 RadiusWidth-Eigenschaft (VBS), 1780
 rand-Funktion, 180
 RangeLabelView-Objekt, 1464
 RangeQualityView-Objekt, 1466
 ReadOnly-Eigenschaft (VBS), 2043
 ReadTags, 2236
 realloc-Funktion, 181
 RecipeName-Eigenschaft (VBS), 1781
 RecipeNumber-Eigenschaft (VBS), 1781
 RecordName-Eigenschaft (VBS), 1782
 RecordNumber-Eigenschaft (VBS), 1782
 Rectangle-Objekt, 1473
 RelativeFillLevel-Eigenschaft (VBS), 1782
 remove-Funktion, 179
 rename-Funktion, 179
 ReportJob, 81
 ResetBit, 82, 1204, 1250
 ResetBitInTag, 83, 1205, 1251
 rewind-Funktion, 180
 Rezeptur
 Exportformat, 1172
 Rezepturdatensatz
 Exportformat, 1172
 RotationAngle-Eigenschaft (VBS), 1784
 RotationCenterLeft-Eigenschaft (VBS), 1784
 RotationCenterTop-Eigenschaft (VBS), 1785
 Rotation-Eigenschaft (VBS), 1783
 RoundButton-Objekt, 1474
 RoundCornerHeight-Eigenschaft (VBS), 1786
 RoundCornerWidth-Eigenschaft (VBS), 1786
 RowNumber-Eigenschaft (VBS), 2122
 RowScrollbar-Eigenschaft (VBS), 1787
 RowSizingEnable-Eigenschaft (VBS), 2083
 RowTitles-Eigenschaft (VBS), 1820
 RTPersistence-Eigenschaft (VBS), 1788
 RTPersistencePasswordLevel-Eigenschaft (VBS), 2044

RTPersistenceType-Eigenschaft (VBS), 1788
 RücksetzeBit, 82, 1204, 1250
 RücksetzeBitInVariable, 83, 1205, 1251
 RulerColor-Eigenschaft (VBS), 1789
 RulerPrecision(i)-Eigenschaft (VBS), 2122
 RulerXPrecision(i)-Eigenschaft (VBS), 2123
 RulerYPrecision(i)-Eigenschaft (VBS), 2124

S

SafelyRemoveHardware, 1206
 SaveDataRecord, 1207
 ScaleColor-Eigenschaft (VBS), 1790
 ScaleDenominator-Eigenschaft (VBS), 1795
 ScaleGradation-Eigenschaft (VBS), 1790
 ScaleLabelColor-Eigenschaft (VBS), 1791
 ScaleNumerator-Eigenschaft (VBS), 1792
 ScalePosition-Eigenschaft (VBS), 1792
 ScaleTickColor-Eigenschaft (VBS), 1792
 ScaleTickLabelPosition-Eigenschaft (VBS), 1793
 ScaleTickLength-Eigenschaft (VBS), 1794
 ScaleTickPosition-Eigenschaft (VBS), 1794
 Scaling-Eigenschaft (VBS), 1795
 ScalingType-Eigenschaft (VBS), 1795
 SchlieÙeAlleArchive, 1162
 SchreibeDatensatzInSteuerung, 1217
 SchreibeDatensatzvariablenInSteuerung, 1216
 ScreenItem, 1427
 Control, 1393
 ScreenItem-Objekt, 1298, 1323
 ScreenItems-Eigenschaft (VBS), 1796
 ScreenItems-Objekt (Auflistung), 1300, 1325
 ScreenName-Eigenschaft (VBS), 1797
 Screen-Objekt, 1297, 1321
 Screens-Eigenschaft (VBS), 1797
 Screens-Objekt (Auflistung), 1296
 ScreenWindow-Objekt, 1477
 Scrollable-Eigenschaft (VBS), 2084
 SecondNeedleHeight-Eigenschaft (VBS), 1798
 SecondNeedleWidth-Eigenschaft (VBS), 1798
 SegmentColoring-Eigenschaft (VBS), 1799
 SeiteAbwärts, 1201
 SeiteAufwärts, 1202
 SelectBackColor-Eigenschaft (VBS), 1799
 SelectedCellColor-Eigenschaft (VBS), 1800
 SelectedCellForeColor-Eigenschaft (VBS), 1801
 SelectedIndex-Eigenschaft (VBS), 1801
 SelectedRowColor-Eigenschaft (VBS), 1802
 SelectedRowForeColor-Eigenschaft (VBS), 1803
 SelectedStatisticArea, 2249
 SelectedTitleColor-Eigenschaft (VBS), 1803
 SelectedTitleForeColor-Eigenschaft (VBS), 1804

SelectForeColor-Eigenschaft (VBS), 1805
SelectionBackColor-Eigenschaft (VBS), 1806
SelectionColoring-Eigenschaft (VBS), 1806
SelectionForeColor-Eigenschaft (VBS), 1807
SelectionMode-Eigenschaft (VBS), 2045, 2046
SelectionRectColor-Eigenschaft (VBS), 1808
SelectionRect-Eigenschaft (VBS), 1808
SelectionRectWidth-Eigenschaft (VBS), 1809
SelectionType-Eigenschaft (VBS), 1810
SendeEMail, 1208
SendEMail, 1208
SeparatorBackColor-Eigenschaft (VBS), 1810
SeparatorColor-Eigenschaft (VBS), 1811
SeparatorStyle-Eigenschaft (VBS), 1812
SeparatorWidth-Eigenschaft (VBS), 1812
SeperatorCornerStyle-Eigenschaft (VBS), 1812
ServerExport, 2250
ServerImport, 2251
ServerNames-Eigenschaft (VBS), 1813
ServerPrefix-Eigenschaft (VBS), 1814
ServerScale-Eigenschaft (VBS), 1813
SetAcousticSignal, 1209
SetAndGetBrightness, 1211
SetBit, 84, 1211, 1252
SetBitInTag, 85, 1212, 1253
SetBrightness, 1214
setbuf-Funktion, 180
SetConnectionMode, 1215
SetDataRecordTagsToPLC, 1216
SetDataRecordToPLC, 1217
SetDeviceMode, 1219
SetDisplayMode, 1220
SetLanguage, 87, 1220
SetMessageLogMode, 1210
SetPLCMode, 1221
SetpointTrendArchiveStartId(i)-Eigenschaft (VBS), 2124
SetpointTrendColor(i)-Eigenschaft (VBS), 2125
SetpointTrendNumberOfValues(i)-Eigenschaft (VBS), 2126
SetPropBOOL, 89
SetPropChar, 90
SetPropDouble, 92
SetPropertyByConstant, 95, 1255
SetPropertyByProperty, 96, 1256
SetPropertyByTag, 98, 1258
SetPropertyByTagIndirect, 100, 1260
SetPropLong, 93
SetRecipeTags, 1222
SetScreenKeyboardMode, 1223
SetTagBit, 101
SetTagBitState, 108
SetTagBitStateWait, 111
SetTagBitWait, 116
SetTagByProperty, 119, 1262
SetTagByte, 101
SetTagByteState, 108
SetTagByteStateWait, 111
SetTagByteWait, 116
SetTagChar, 101
SetTagCharState, 108
SetTagCharStateWait, 111
SetTagCharWait, 116
SetTagDateTime, 101, 103
SetTagDouble, 101
SetTagDoubleState, 108
SetTagDoubleStateWait, 111
SetTagDoubleWait, 117
SetTagDWord, 101
SetTagDWordState, 108
SetTagDWordStateWait, 111
SetTagDWordWait, 117
SetTagFloat, 101
SetTagFloatState, 108
SetTagFloatStateWait, 111
SetTagFloatWait, 117
SetTagIndirect, 122, 1264
SetTagIndirectByProperty, 121, 122, 1263, 1265
SetTagIndirectByTagIndirect, 124, 1266
SetTagIndirectWithOperatorInputAlarm, 125, 1267, 1268
SetTagMultiStateWait-, 104
SetTagMultiWait, 105
SetTagRaw, 101
SetTagRawState, 108
SetTagRawStateWait, 111
SetTagRawWait, 117
SetTagSByte, 101
SetTagSByteState, 109
SetTagSByteStateWait, 111
SetTagSByteWait, 117
SetTagSDWord, 101
SetTagSDWordState, 109
SetTagSDWordStateWait, 111
SetTagSDWordWait, 117
SetTagSWord, 101
SetTagSWordState, 109
SetTagSWordStateWait, 111
SetTagSWordWait, 117
SetTagValue, 113
SetTagValueWait, 115
SetTagWord, 101
SetTagWordState, 109
SetTagWordStateWait, 111

SetTagWordWait, 117
setvbuf-Funktion, 179
SetzeAkustischesSignal, 1209
SetzeAnlaufverhalten, 1238
SetzeAnzeigemodus, 1220
SetzeBetriebsartGerät, 1219
SetzeBit, 84, 1211, 1252
SetzeBitInVariable, 85, 1212, 1253
SetzeEigenschaftDurchEigenschaft, 96, 1256
SetzeEigenschaftDurchKonstante, 95, 1255
SetzeEigenschaftDurchVariable, 98, 1258
SetzeEigenschaftDurchVariableIndirekt, 100, 1260
SetzeHelligkeit, 1214
SetzeModusBildschirmtastatur, 1223
SetzePLCModus, 1221
SetzeProtokolliermodusMeldung, 1210
SetzeRezepturvariablen, 1222
SetzeSommerzeit, 1218
SetzeSprache, 1220
SetzeundLeseHelligkeit, 1211
SetzeVariable, 118, 1224, 1261
SetzeVariableIndirekt, 122, 1264
SetzeVariableIndirektDurchEigenschaft, 121, 122, 1263, 1265
SetzeVariableIndirektDurchVariableIndirekt, 124, 1266
SetzeVariableIndirektMitBedienprotokollierung, 125, 1267, 1268
SetzeVerbindungsmodus, 1215
SetzteVariableDurchEigenschaft, 119, 1262
Shared-Eigenschaft (VBS), 1814
ShareTimeColumn-Eigenschaft (VBS), 2127
ShareValueAxis-Eigenschaft (VBS), 2127
ShareXAxis-Eigenschaft (VBS), 2128
ShareYAxis-Eigenschaft (VBS), 2128
ShiftDecimalPoint-Eigenschaft (VBS), 1814
ShortenCellText-Eigenschaft (VBS), 2047
ShortenColumnHeaderText-Eigenschaft (VBS), 2047
ShowAlarmsFromDate-Eigenschaft (VBS), 1814
ShowAlarmWindow, 1227, 1232
ShowBadTagState-Eigenschaft (VBS), 1815
ShowBar-Eigenschaft (VBS), 1816
ShowBorder-Eigenschaft (VBS), 2129
ShowCaption-Eigenschaft (VBS), 1816
ShowColumn(i)-Eigenschaft (VBS), 2129
ShowColumnSelection, 2251
ShowComment, 2252
ShowCurve(i)-Eigenschaft (VBS), 2130
ShowDecimalPoint-Eigenschaft (VBS), 1817
ShowDisplayOptionsDialog, 2252
ShowEmergencyQuitDialog, 2253
ShowFillLevel-Eigenschaft (VBS), 1817
ShowFocusRectangle-Eigenschaft (VBS), 1818
ShowHelp, 2253
ShowHideList, 2254
ShowHitList, 2254
ShowInputControls-Eigenschaft (VBS), 2130
ShowLargeTicksOnly-Eigenschaft (VBS), 1818, 2027
ShowLimitMarkers-Eigenschaft (VBS), 1819
ShowLockDialog, 2255
ShowLockList, 2256
ShowLogonDialog, 1269
ShowLongTermArchiveList, 2256
ShowMainFrame-Eigenschaft (VBS), 2131
ShowMessagesAtDate-Eigenschaft (VBS), 2084
ShowOverflowIndicator-Eigenschaft (VBS), 2131
ShowPeakValuePointer-Eigenschaft (VBS), 1819
ShowPercentageAxis, 2257
ShowPosition-Eigenschaft (VBS), 1820
ShowPropertyDialog, 2258
ShowRuler-Eigenschaft (VBS), 1821
ShowRulerInAxis-Eigenschaft (VBS), 1821
ShowScale-Eigenschaft (VBS), 1822
ShowScrollbars-Eigenschaft (VBS), 1822
ShowSelectArchive, 2258
ShowSelection, 2259
ShowSelectionDialog, 2259
ShowSelectTimeBase, 2260
ShowSetpointTrend(i)-Eigenschaft (VBS), 2132
ShowShortTermArchiveList, 2260
ShowSoftwareVersion, 1230
ShowSort, 2261
ShowSortButton-Eigenschaft (VBS), 1823
ShowSortDialog, 2261
ShowSortIcon-Eigenschaft (VBS), 1823, 1824
ShowStatusBar-Eigenschaft (VBS), 1824
ShowSystemAlarm, 1231
ShowTableGridlines-Eigenschaft (VBS), 1825
ShowTagSelection, 2262
ShowThumb-Eigenschaft (VBS), 1825
ShowTickLabels-Eigenschaft (VBS), 1826
ShowTicks-Eigenschaft (VBS), 1826
ShowTimebaseDialog, 2262
ShowTimeSelection, 2263
ShowTitle-Eigenschaft (VBS), 1827
ShowToolBar-Eigenschaft (VBS), 1827
ShowTrendIcon-Eigenschaft (VBS), 1828
ShowTrendIndicator-Eigenschaft (VBS), 1828
ShowTrendSelection, 2263
ShowVerticalGridlines-Eigenschaft (VBS), 2048
ShowXValuesExponential(i)-Eigenschaft (VBS), 2048

- ShowYValuesExponential(i)-Eigenschaft (VBS), 2049
- SichereRAMDateisystem, 1155
- sin-Funktion, 177
- sinh-Funktion, 177
- Slider-Objekt, 1483
- SmartClientView-Objekt, 1484
- Smartobjekte
 - Control, 1393
- SmartServer, 1163
- SmartTag-Objekt, 1302, 1329
- SmartTags-Eigenschaft (VBS), 1830
- SmartTags-Objekt (Auflistung), 1301, 1331
- SortByTimeDirection-Eigenschaft (VBS), 1830
- SortByTimeEnabled-Eigenschaft (VBS), 1830
- SortByTimeEnable-Eigenschaft (VBS), 2050
- Sort-Eigenschaft (VBS), 2085
- SortOnColumnHeaderClick-Eigenschaft (VBS), 2050
- SortSequence-Eigenschaft (VBS), 1831
- SortTimeAscending-Eigenschaft (VBS), 2051
- SortTimeEnable-Eigenschaft (VBS), 2051
- SpeichereDatensatz, 1207
- sqrt-Funktion, 177
- srand-Funktion, 180
- StartAngle-Eigenschaft (VBS), 1831
- StarteArchivierung, 1232
- StarteNächstesArchiv, 1233
- StarteProgramm, 1234
- StartLeft-Eigenschaft (VBS), 2052
- StartLogging, 1232
- StartProgram, 126, 1234
- StartSequenceLog, 1233
- StartStopUpdate, 2264
- StartStyle-Eigenschaft (VBS), 1832
- StartTop-Eigenschaft (VBS), 1833
- StartValue-Eigenschaft (VBS), 1833
- StatusbarBackColor-Eigenschaft (VBS), 1834
- StatusbarElementAdd-Eigenschaft (VBS), 1834
- StatusbarElementAutoSize-Eigenschaft (VBS), 1835
- StatusbarElementCount-Eigenschaft (VBS), 1836
- StatusbarElementIconId-Eigenschaft (VBS), 1836
- StatusbarElementID-Eigenschaft (VBS), 1837
- StatusbarElementIndex-Eigenschaft (VBS), 1838
- StatusbarElementName-Eigenschaft (VBS), 1838
- StatusbarElementRemove-Eigenschaft (VBS), 1839
- StatusbarElementRename-Eigenschaft (VBS), 1840
- StatusbarElements-Eigenschaft (VBS), 1840
- StatusbarElementTooltipText-Eigenschaft (VBS), 1840
- StatusbarElementUserDefined-Eigenschaft (VBS), 1841
- StatusbarElementVisible-Eigenschaft (VBS), 1842
- StatusbarElementWidth-Eigenschaft (VBS), 1842
- StatusbarFontColor-Eigenschaft (VBS), 1843
- StatusbarShowArchiveName-Eigenschaft (VBS), 2132
- StatusbarShowColumn-Eigenschaft (VBS), 2133
- StatusbarShowRecord-Eigenschaft (VBS), 2133
- StatusbarShowRow-Eigenschaft (VBS), 2134
- StatusbarShowText-Eigenschaft (VBS), 2134
- StatusbarShowTooltips-Eigenschaft (VBS), 1844
- StatusbarText-Eigenschaft (VBS), 1844
- StatusbarUseBackColor-Eigenschaft (VBS), 1845
- StatusbarVisible-Eigenschaft (VBS), 1846
- StatusForce-Objekt, 1486
- stdio-Funktionen
 - clearerr-Funktion, 180
 - fclose-Funktion, 179
 - feof-Funktion, 179
 - ferror-Funktion, 179
 - fflush-Funktion, 179
 - fgetc-Funktion, 179
 - fgetpos-Funktion, 179
 - fgets-Funktion, 179
 - fopen-Funktion, 179
 - fputc-Funktion, 179
 - fputs-Funktion, 179
 - fread-Funktion, 180
 - freopen-Funktion, 179
 - fseek-Funktion, 179
 - fsetpos-Funktion, 179
 - ftell-Funktion, 179
 - fwrite-Funktion, 180
 - getc-Funktion, 179
 - putc-Funktion, 179
 - remove-Funktion, 179
 - rename-Funktion, 179
 - rewind-Funktion, 180
 - setbuf-Funktion, 180
 - setvbuf-Funktion, 179
 - tmpfile-Funktion, 179
 - tmpnam-Funktion, 179
 - ungetc-Funktion, 179
 - vfprintf-Funktion, 179
 - vsprintf-Funktion, 179
- stdlib-Funktionen
 - abort-Funktion, 180
 - abs-Funktion, 180
 - atof-Funktion, 180
 - atoi-Funktion, 180
 - atol-Funktion, 180
 - bsearch-Funktion, 180
 - calloc-Funktion, 180
 - div-Funktion, 180

- exit-Funktion, 180
- free-Funktion, 180
- getenv-Funktion, 180
- labs-Funktion, 180
- ldiv-Funktion, 180
- malloc-Funktion, 180
- qsort-Funktion, 181
- rand-Funktion, 180
- realloc-Funktion, 181
- srand-Funktion, 180
- strtod-Funktion, 180
- strtol-Funktion, 180
- strtoul-Funktion, 180
- system-Funktion, 180
- SteuereWebServer, 1164
- StopLogging, 1235
- Stop-Methode, 2264
- StoppeArchivierung, 1235
- StoppeRuntime, 127, 1236, 1269
- StopRuntime, 127, 1236, 1269
- strcat-Funktion, 181
- strchr-Funktion, 181
- strcmp-Funktion, 181
- strcpy-Funktion, 181
- strcspn-Funktion, 181
- strerror-Funktion, 181
- strftime-Funktion, 182
- String-Funktionen
 - strcat-Funktion, 181
 - strchr-Funktion, 181
 - strcmp-Funktion, 181
 - strcpy-Funktion, 181
 - strcspn-Funktion, 181
 - strerror-Funktion, 181
 - strlen-Funktion, 181
 - strncat-Funktion, 181
 - strncmp-Funktion, 181
 - strncpy-Funktion, 181
 - strpbrk-Funktion, 181
 - strrchr-Funktion, 181
 - strspn-Funktion, 181
 - strstr-Funktion, 181
 - strtok-Funktion, 181
- strlen-Funktion, 181
- strncat-Funktion, 181
- strncmp-Funktion, 181
- strncpy-Funktion, 181
- strpbrk-Funktion, 181
- strrchr-Funktion, 181
- strspn-Funktion, 181
- strstr-Funktion, 181
- strtod-Funktion, 180
- strtok-Funktion, 181
- strtol-Funktion, 180
- strtoul-Funktion, 180
- Style-Eigenschaft (VBS), 1846
- StyleSettings-Eigenschaft (VBS), 1847, 2065
- SwapDimensionsWithOrientation-Eigenschaft (VBS), 2134
- SwapFirstWithLastConnection-Eigenschaft (VBS), 1847
- Switch-Objekt, 1488
- SymbolicIOField-Objekt, 1491
- SymbolLibrary-Objekt, 1496
- SystemDiagnoseView-Objekt, 1498
- SystemDiagnoseWindow-Objekt, 1501
- Systemfunktion
 - HardwareSicherEntfernen, 1206
 - SetzeAnlaufverhalten, 1238
 - WinACMPLeeseVersion, 1238
 - WinACMPSetzeStartModus, 1239
- system-Funktion, 180
- Systemfunktionen
 - Abmelden, 1193
 - AktiviereBild, 25, 1150, 1240
 - AktiviereBildInBildfenster, 26, 1241
 - AktiviereBildMitNummer, 1151
 - AktiviereSystemdiagnoseAnzeige, 1152
 - AktiviereVorherigesBild, 1149
 - Anmelden, 1194
 - AnzeigenSlideInBild, 1230
 - ArchiviereProtokolldatei, 1153
 - BearbeiteMeldung, 1167
 - Codieren, 1167
 - CodierenEx, 1168
 - DruckeBild, 1203
 - DruckeProtokoll, 1203
 - ErfasseBenutzeraktion, 1195
 - ErhöheVariable, 71, 1187, 1244
 - ExportiereDatensätze, 1170
 - ExportiereDatensätzeMitPrüfsumme, 1172
 - ExportiereImportiereBenutzerverwaltung, 1175, 1243
 - GeheZumAnfang, 1183
 - GeheZumEnde, 1182
 - ImportiereDatensätze, 1183
 - ImportiereDatensätzeMitPrüfsumme, 1185
 - InvertiereBit, 76, 1189, 1247
 - InvertiereBitInVariable, 77, 1190, 1248
 - InvertiereLineareSkalierung, 74, 1187, 1245
 - KalibriereTouchscreen, 1155
 - LadeDatensatz, 1192
 - LeseBenutzername, 1182
 - LeseDatensatzAusSteuerung, 1176

LeseDatensatzname, 1178
LeseDatensatzvariablenAusSteuerung, 1180
LeseGruppennummer, 1180
LeseHelligkeit, 1176
LeseKennwort, 1181
LineareSkalierung, 79, 1191, 1249
LoescheArchiv, 1162
LöscheDatensatz, 1159
LöscheDatensatzSpeicher, 1161
LöscheMeldepuffer, 1158
LöscheMeldepufferProTool, 1159
ÖffneAlleArchive, 1197
ÖffneBildschirmtastatur, 1200
ÖffneDialogSystemsteuerung, 1198
ÖffneEingabeaufforderung, 1198
ÖffneInternetExplorer, 1199
ÖffneTaskManager, 1201
PROFIsafeTrennen, 1237
PROFIsafeVerbinden, 1169
QuittiereMeldung, 1149
RücksetzeBit, 82, 1204, 1250
RücksetzeBitInVariable, 83, 1205, 1251
SchließeAlleArchive, 1162
SchreibeDatensatzInSteuerung, 1217
SchreibeDatensatzvariablenInSteuerung, 1216
SeiteAbwärts, 1201
SeiteAufwärts, 1202
SendeEMail, 1208
SetzeAkustischesSignal, 1209
SetzeAnzeigemodus, 1220
SetzeBetriebsartGerät, 1219
SetzeBit, 84, 1211, 1252
SetzeBitInVariable, 85, 1212, 1253
SetzeEigenschaftDurchEigenschaft, 96, 1256
SetzeEigenschaftDurchKonstante, 95, 1255
SetzeEigenschaftDurchVariable, 98, 1258
SetzeEigenschaftDurchVariableIndirekt, 100, 1260
SetzeHelligkeit, 1214
SetzeModusBildschirmtastatur, 1223
SetzeProtokolliermodusMeldung, 1210
SetzeRezepturvariablen, 1222
SetzeSommerzeit, 1218
SetzeSprache, 1220
SetzeUndLeseHelligkeit, 1211
SetzeVariable, 118, 1224, 1261
SetzeVariableIndirekt, 122, 1264
SetzeVariableIndirektDurchEigenschaft, 121, 122, 1263, 1265
SetzeVariableIndirektDurchVariableIndirekt, 124, 1266

SetzeVariableIndirektMitBedienprotokollierung, 125, 1267
SetzeVariableMitBedienprotokollierung, 125, 1268
SetzeVerbindungsmodus, 1215
SetzteVariableDurchEigenschaft, 119, 1262
SichereRAMDateisystem, 1155
SpeichereDatensatz, 1207
StarteArchivierung, 1232
StarteNächstesArchiv, 1233
StarteProgramm, 1234
SteuereSmartServer, 1163
SteuereWebServer, 1164
StoppeArchivierung, 1235
StoppeRuntime, 127, 1236, 1269
TextAbrufen, 1194
VerringereVariable, 29, 1166, 1242
WechseleVerbindung, 1156
ZeigeAnmeldedialog, 1269
ZeigeHilfetext, 1228
ZeigeMeldefenster, 1227
ZeigePopupBild, 1229
ZeigeSoftwareVersion, 1230
ZeigeSystemDiagnosefenster, 1232
ZeigeSystemmeldung, 1231

T

TableBackColor-Eigenschaft (VBS), 1848
TableColor2-Eigenschaft (VBS), 1849
TableColor-Eigenschaft (VBS), 1849
TableFocusOnButtonCommand-Eigenschaft (VBS), 2085
TableForeColor2-Eigenschaft (VBS), 1851
TableForeColor-Eigenschaft (VBS), 1850
TableGridLineColor-Eigenschaft (VBS), 1852
TableHeaderBackColor-Eigenschaft (VBS), 1852
TableHeaderForeColor-Eigenschaft (VBS), 1853
TableView-Objekt, 1434
Tag-Objekt, 1332
TagPrefix-Eigenschaft (VBS), 1854
Tags, 1335, 1336
Tags-Objekt (Auflistung), 1336
tan-Funktion, 177
tanh-Funktion, 177
Template-Eigenschaft (VBS), 1856
TerminatePROFIsafe, 1237
TextAbrufen, 1194
Text-Eigenschaft (VBS), 1857
TextField-Objekt, 1505
TextList-Eigenschaft (VBS), 1857
TextOff-Eigenschaft (VBS), 1858

- TextOn-Eigenschaft (VBS), 1858
 TextOrientation-Eigenschaft (VBS), 1859
 ThumbBackColor-Eigenschaft (VBS), 1860
 TicksColor-Eigenschaft (VBS), 1860
 TickStyle-Eigenschaft (VBS), 1861
 TimeAxisBeginTime(i)-Eigenschaft (VBS), 1862
 TimeAxisEndTime-Eigenschaft (VBS), 1862
 TimeAxisLabel(i)-Eigenschaft (VBS), 1863
 TimeAxisRange-Eigenschaft (VBS), 1863
 TimeAxisShowGridLines(i)-Eigenschaft (VBS), 2052
 TimeAxisShowLargeIncrements(i)-Eigenschaft (VBS), 2135
 TimeAxisShowSmallIncrements(i)-Eigenschaft (VBS), 2135
 TimeAxisTimeFormat(i)-Eigenschaft (VBS), 1863
 TimeBase-Eigenschaft (VBS), 1864
 TimeColumnActualize-Eigenschaft (VBS), 1865
 TimeColumnAdd-Eigenschaft (VBS), 1865
 TimeColumnAlignment(i)-Eigenschaft (VBS), 1866
 TimeColumnAlignment-Eigenschaft (VBS), 1866
 TimeColumnBackColor-Eigenschaft (VBS), 1867
 TimeColumnBeginTime-Eigenschaft (VBS), 1867
 TimeColumnCaption-Eigenschaft (VBS), 1868
 TimeColumnCount-Eigenschaft (VBS), 1868
 TimeColumnDateFormat-Eigenschaft (VBS), 1869
 TimeColumnEndTime-Eigenschaft (VBS), 1869
 TimeColumnForeColor-Eigenschaft (VBS), 1870
 TimeColumnFormat(i)-Eigenschaft (VBS), 2053
 TimeColumnHideText-Eigenschaft (VBS), 1870
 TimeColumnHideTitleText-Eigenschaft (VBS), 1871
 TimeColumnIndex-Eigenschaft (VBS), 1871
 TimeColumnLength-Eigenschaft (VBS), 1872
 TimeColumnMeasurePoints-Eigenschaft (VBS), 1872
 TimeColumnName-Eigenschaft (VBS), 1873
 TimeColumnRangeType-Eigenschaft (VBS), 1873
 TimeColumnRemove-Eigenschaft (VBS), 1874
 TimeColumnRename -Eigenschaft (VBS), 1874
 TimeColumnRepos-Eigenschaft (VBS), 1875
 TimeColumnShowDate-Eigenschaft (VBS), 1875
 TimeColumnShowIcon-Eigenschaft (VBS), 1876
 TimeColumnShowTitleIcon-Eigenschaft (VBS), 1876
 TimeColumnSort-Eigenschaft (VBS), 1877
 TimeColumnSortIndex-Eigenschaft (VBS), 1877
 TimeColumnTimeFormat-Eigenschaft (VBS), 1878
 TimeColumnTimeRangeBase-Eigenschaft (VBS), 1878
 TimeColumnTimeRangeFactor-Eigenschaft (VBS), 1879
 TimeColumnUseValueColumnColors-Eigenschaft (VBS), 1879
 TimeColumnVisible-Eigenschaft (VBS), 1880
 time-Funktion, 182
 time-Funktionen
 asctime-Funktion, 182
 ctime-Funktion, 182
 difftime-Funktion, 182
 gmtime-Funktion, 182
 localtime-Funktion, 182
 mktime-Funktion, 182
 strftime-Funktion, 182
 time-Funktion, 182
 TimeJumpColor(i)-Eigenschaft (VBS), 2136
 TimeJumpEnabled(i)-Eigenschaft (VBS), 2137
 TimeOverlapColor(i)-Eigenschaft (VBS), 2137
 TimeOverlapEnabled(i)-Eigenschaft (VBS), 2138
 TimeRangeBase(i)-Eigenschaft (VBS), 2086
 TimeRangeFactor(i)-Eigenschaft (VBS), 2087
 TitleColor-Eigenschaft (VBS), 2054
 TitleCut-Eigenschaft (VBS), 1882
 TitleDarkShadowColor-Eigenschaft (VBS), 1882
 TitleForeColor-Eigenschaft (VBS), 1883
 TitleGridLineColor-Eigenschaft (VBS), 1884
 TitleLightShadowColor-Eigenschaft (VBS), 1884
 TitleSort-Eigenschaft (VBS), 1885
 TitleStyle-Eigenschaft (VBS), 1886
 tmpfile-Funktion, 179
 tmpnam-Funktion, 179
 Toggle-Eigenschaft (VBS), 1886
 ToleranceColor-Eigenschaft (VBS), 1887
 Tolerance-Eigenschaft (VBS), 1887
 ToleranceLowerLimitColor-Eigenschaft (VBS), 1889
 ToleranceLowerLimit-Eigenschaft (VBS), 1888
 ToleranceLowerLimitEnabled-Eigenschaft (VBS), 1889
 ToleranceLowerLimitRelative-Eigenschaft (VBS), 1890
 ToleranceUpperLimitColor-Eigenschaft (VBS), 1891
 ToleranceUpperLimit-Eigenschaft (VBS), 1890
 ToleranceUpperLimitEnabled-Eigenschaft (VBS), 1891
 ToleranceUpperLimitRelative-Eigenschaft (VBS), 1892
 tolower-Funktion, 175
 ToolbarAlignment-Eigenschaft (VBS), 1893
 ToolbarBackColor-Eigenschaft (VBS), 1893
 ToolbarButtonActive-Eigenschaft (VBS), 1894
 ToolbarButtonAdd-Eigenschaft (VBS), 1895
 ToolbarButtonAuthorization-Eigenschaft (VBS), 1900
 ToolbarButtonBeginGroup-Eigenschaft (VBS), 1895
 ToolbarButtonCount-Eigenschaft (VBS), 1896
 ToolbarButtonEnabled-Eigenschaft (VBS), 1897
 ToolbarButtonHotKey-Eigenschaft (VBS), 1897

- ToolBarButtonID-Eigenschaft (VBS), 1898
- ToolBarButtonIndex-Eigenschaft (VBS), 1898
- ToolBarButtonLocked-Eigenschaft (VBS), 1899
- ToolBarButtonName-Eigenschaft (VBS), 1900
- ToolBarButtonRemove-Eigenschaft (VBS), 1901
- ToolBarButtonRename-Eigenschaft (VBS), 1901
- ToolBarButtonRepos-Eigenschaft (VBS), 1902
- ToolBarButtonTooltipText-Eigenschaft (VBS), 1903
- ToolBarButtonUserDefined-Eigenschaft (VBS), 1903
- ToolBarShowTooltips-Eigenschaft (VBS), 1904
- ToolBarUseBackColor-Eigenschaft (VBS), 1904
- ToolBarUseHotKeys-Eigenschaft (VBS), 1905
- ToolBarVisible-Eigenschaft (VBS), 1906
- ToolTipText-Eigenschaft (VBS), 1906
- Top-Eigenschaft (VBS), 1907
- TopOffset-Eigenschaft (VBS), 1910
- Total-Eigenschaft (VBS), 1910
- toupper-Funktion, 175
- Trace-Methode, 2265
- Transparency-Eigenschaft (VBS), 1911
- TransparentColorDeactivatedPicture-Eigenschaft (VBS), 1913
- TransparentColor-Eigenschaft (VBS), 1912
- TransparentColorPictureOff-Eigenschaft (VBS), 1913
- TransparentColorPictureOn-Eigenschaft (VBS), 1914
- TrendActualize-Eigenschaft (VBS), 1915
- TrendAdd-Eigenschaft (VBS), 1915
- TrendBeginTime-Eigenschaft (VBS), 1916
- TrendColor-Eigenschaft (VBS), 1916
- TrendCount-Eigenschaft (VBS), 1917
- TrendEndTime-Eigenschaft (VBS), 1917
- TrendExtendedColorSet-Eigenschaft (VBS), 1918
- TrendFill-Eigenschaft (VBS), 1918
- TrendIndex-Eigenschaft (VBS), 1920
- TrendIndicatorColor-Eigenschaft (VBS), 1920
- TrendLabel-Eigenschaft (VBS), 1921
- TrendLineStyle-Eigenschaft (VBS), 1921
- TrendLineType-Eigenschaft (VBS), 1922
- TrendLineWidth-Eigenschaft (VBS), 1923
- TrendLowerLimitColor-Eigenschaft (VBS), 1924
- TrendLowerLimitColoring-Eigenschaft (VBS), 1924
- TrendLowerLimit-Eigenschaft (VBS), 1923
- TrendMeasurePoints-Eigenschaft (VBS), 1925
- TrendName-Eigenschaft (VBS), 1925
- TrendPointColor-Eigenschaft (VBS), 1926
- TrendPointStyle-Eigenschaft (VBS), 1926
- TrendPointWidth-Eigenschaft (VBS), 1927
- TrendProvider-Eigenschaft (VBS), 1927
- TrendRangeType-Eigenschaft (VBS), 1928
- TrendRemove-Eigenschaft (VBS), 1929
- TrendRename-Eigenschaft (VBS), 1929
- TrendRepos-Eigenschaft (VBS), 1930
- TrendRulerControl, 1508
- TrendSelectTagNameX-Eigenschaft (VBS), 1930
- TrendSelectTagNameY-Eigenschaft (VBS), 1931
- TrendTag-Eigenschaft (VBS), 2055
- TrendTagNameX-Eigenschaft (VBS), 1931
- TrendTagNameY-Eigenschaft (VBS), 1932
- TrendTimeRangeBase-Eigenschaft (VBS), 1932
- TrendTimeRangeFactor-Eigenschaft (VBS), 1933
- TrendTrendWindow-Eigenschaft (VBS), 1933
- TrendUncertainColor-Eigenschaft (VBS), 1934
- TrendUncertainColoring-Eigenschaft (VBS), 1934
- TrendUpperLimitColor-Eigenschaft (VBS), 1935
- TrendUpperLimitColoring-Eigenschaft (VBS), 1936
- TrendUpperLimit-Eigenschaft (VBS), 1935
- TrendView-Objekt, 1516
- TrendVisible-Eigenschaft (VBS), 1937
- TrendWindowAdd-Eigenschaft (VBS), 1937
- TrendWindowCoarseGridColor-Eigenschaft (VBS), 1938
- TrendWindowCoarseGrid-Eigenschaft (VBS), 1938
- TrendWindowFineGridColor-Eigenschaft (VBS), 1939
- TrendWindowFineGrid-Eigenschaft (VBS), 1939
- TrendWindowForegroundTrendGrid-Eigenschaft (VBS), 1940
- TrendWindowGridInTrendColor-Eigenschaft (VBS), 1940
- TrendWindowHorizontalGrid-Eigenschaft (VBS), 1941
- TrendWindowIndex-Eigenschaft (VBS), 1941
- TrendWindowName-Eigenschaft (VBS), 1942
- TrendWindowRemove-Eigenschaft (VBS), 1942
- TrendWindowRename-Eigenschaft (VBS), 1943
- TrendWindowRepos-Eigenschaft (VBS), 1943
- TrendWindowRulerColor-Eigenschaft (VBS), 1944
- TrendWindowRulerLayer-Eigenschaft (VBS), 1945
- TrendWindowRulerStyle-Eigenschaft (VBS), 1945
- TrendWindowRulerWidth-Eigenschaft (VBS), 1946
- TrendWindowSpacePortion-Eigenschaft (VBS), 1946
- TrendWindowVerticalGrid-Eigenschaft (VBS), 1947
- TrendWindowVisible-Eigenschaft (VBS), 1947
- TrendXAxis-Eigenschaft (VBS), 1948
- TrendYAxis-Eigenschaft (VBS), 1948
- TrenFillColor-Eigenschaft (VBS), 1919
- TriggerOperatorEvent, 129
- TubeArcObject, 1520
- TubeDoubleTeeObject, 1522
- TubePolyline, 1524
- TubeTeeObject, 1526

Type-Eigenschaft (VBS), 2138

U

- uaAddArchive, 130
- uaAddField, 131
- uaArchiveClose, 132
- uaArchiveDelete, 132
- uaArchiveExport, 133
- uaArchiveGetCount, 134
- uaArchiveGetFieldLength, 135
- uaArchiveGetFieldName, 135
- uaArchiveGetFields, 136
- uaArchiveGetFieldType, 137
- uaArchiveGetFieldValueDate, 137
- uaArchiveGetFieldValueDouble, 138
- uaArchiveGetFieldValueFloat, 139
- uaArchiveGetFieldValueLong, 140
- uaArchiveGetFieldValueString, 141
- uaArchiveGetFilter, 141
- uaArchiveGetID, 142
- uaArchiveGetName, 143
- uaArchiveGetSort, 143
- uaArchiveImport, 144, 965
- uaArchiveInsert, 145
- uaArchiveMoveFirst, 146
- uaArchiveMoveLast, 146
- uaArchiveMoveNext, 147
- uaArchiveMovePrevious, 147
- uaArchiveOpen, 148
- uaArchiveReadTagValues, 149
- uaArchiveReadTagValuesByName, 150
- uaArchiveRequery, 151
- uaArchiveSetFieldValueDate, 152
- uaArchiveSetFieldValueDouble, 152
- uaArchiveSetFieldValueFloat, 153
- uaArchiveSetFieldValueLong, 154
- uaArchiveSetFieldValueString, 155
- uaArchiveSetFilter, 156
- uaArchiveSetSort, 156
- uaArchiveUpdate, 157, 968
- uaArchiveWriteTagValues, 158
- uaArchiveWriteTagValuesByName, 159
- uaConnect, 159, 948
- uaDisconnect, 160, 949
- uaGetArchive, 161
- uaGetField, 161
- uaGetLastError, 162
- uaGetLastHResult, 164
- uaGetNumArchives, 165
- uaGetNumFields, 165
- uaQueryArchive, 166, 950
- uaQueryArchiveByName, 167, 952
- uaQueryConfiguration, 168, 954
- uaReleaseArchive, 169, 955
- uaReleaseConfiguration, 169
- uaRemoveAllArchives, 170
- uaRemoveAllFields, 171
- uaRemoveArchive, 172
- uaRemoveField, 172
- uaSetArchive, 173
- uaSetField, 174
- UncertainStateColor(i)-Eigenschaft (VBS), 2057
- UncertainStateEnabled(i)-Eigenschaft (VBS), 2058
- ungetc-Funktion, 179
- UnhideAlarm, 2265
- UnitColor-Eigenschaft (VBS), 1949
- Unit-Eigenschaft (VBS), 1949
- UnitText-Eigenschaft (VBS), 1950
- UnitTop-Eigenschaft (VBS), 1950
- UnlockAlarm, 2266
- UpdateEnable-Eigenschaft (VBS), 2059
- UpperLimitColor(i)-Eigenschaft (VBS), 2059
- UpperLimit-Eigenschaft (VBS), 1951
- UpperLimitEnabled(i)-Eigenschaft (VBS), 2060
- UpperLimitValue(i)-Eigenschaft (VBS), 2060
- UseAllServers-Eigenschaft (VBS), 2061
- UseBarBorderConstraints-Eigenschaft (VBS), 2139
- Used-Eigenschaft (VBS), 1951
- UseDesignColorSchema-Eigenschaft (VBS), 1952
- UseDesignShadowSettings-Eigenschaft (VBS), 1953
- UsedPercent-Eigenschaft (VBS), 1955
- UseEffectiveProcessValue-Eigenschaft (VBS), 2140
- UseExponentialFormat-Eigenschaft (VBS), 1955
- UseFlashTransparentColor-Eigenschaft (VBS), 1956
- UseGDI-Eigenschaft (VBS), 2140
- UseMeasurePoints(i)-Eigenschaft (VBS), 2141
- UseMessageColor-Eigenschaft (VBS), 1956
- UseMultipleLimits-Eigenschaft (VBS), 2141
- UserArchiveControl-Objekt, 1528
- UserArchiveNumberOfValues(i)-Eigenschaft (VBS), 2144
- UserArchiveStartId(i)-Eigenschaft (VBS), 2144
- UserView-Objekt, 1536
- UseScaleConstraints-Eigenschaft (VBS), 2142
- UseScaledBarBorder-Eigenschaft (VBS), 2142
- UseSelectedTitleColor-Eigenschaft (VBS), 1957
- UseSeparateDiagrams-Eigenschaft (VBS), 2143
- UseSimplePrecisionOffset-Eigenschaft (VBS), 2143
- UseTableColor2-Eigenschaft (VBS), 1957
- UseTagLimitColors-Eigenschaft (VBS), 1958
- UseTimeRange(i)-Eigenschaft (VBS), 2087

UseTransparentColorDeactivatedPicture-Eigenschaft (VBS), 1959
UseTransparentColor-Eigenschaft (VBS), 1958
UseTransparentColorPictureOff-Eigenschaft (VBS), 1959
UseTransparentColorPictureOn-Eigenschaft (VBS), 1960
UseTrendNameAsLabel-Eigenschaft (VBS), 1960
UV_ColumnWidth_AKZ-Eigenschaft (VBS), 1961
UV_ColumnWidth_Descriptor-Eigenschaft (VBS), 1961
UV_ColumnWidth_InstallationDate-Eigenschaft (VBS), 1961
UV_ColumnWidth_LADDR-Eigenschaft (VBS), 1961
UV_ColumnWidth_Name-Eigenschaft (VBS), 1961
UV_ColumnWidth_OKZ-Eigenschaft (VBS), 1961
UV_ColumnWidth_OperationState-Eigenschaft (VBS), 1962
UV_ColumnWidth_OrderID-Eigenschaft (VBS), 1962
UV_ColumnWidth_ProfileID-Eigenschaft (VBS), 1962
UV_ColumnWidth_Rack-Eigenschaft (VBS), 1962
UV_ColumnWidth_Slot-Eigenschaft (VBS), 1962
UV_ColumnWidth_SoftwareRevision-Eigenschaft (VBS), 1962
UV_ColumnWidth_SpecificProfileData-Eigenschaft (VBS), 1962
UV_ColumnWidth_State-Eigenschaft (VBS), 1963
UV_ColumnWidth_Station-Eigenschaft (VBS), 1963
UV_ColumnWidth_SubAddress-Eigenschaft (VBS), 1963
UV_ColumnWidth_SubSlot-Eigenschaft (VBS), 1963
UV_ColumnWidth_SubSystem-Eigenschaft (VBS), 1963
UV_ColumnWidth_Type-Eigenschaft (VBS), 1963
UV_ShowItem_AKZ-Eigenschaft (VBS), 1963
UV_ShowItem_Descriptor-Eigenschaft (VBS), 1964
UV_ShowItem_InstallationDate-Eigenschaft (VBS), 1964
UV_ShowItem_LADDR-Eigenschaft (VBS), 1964
UV_ShowItem_Name-Eigenschaft (VBS), 1964
UV_ShowItem_OKZ-Eigenschaft (VBS), 1964
UV_ShowItem_OperationState-Eigenschaft (VBS), 1964
UV_ShowItem_OrderID-Eigenschaft (VBS), 1964
UV_ShowItem_ProfileID-Eigenschaft (VBS), 1965
UV_ShowItem_Rack-Eigenschaft (VBS), 1965
UV_ShowItem_Slot-Eigenschaft (VBS), 1965
UV_ShowItem_SoftwareRevision-Eigenschaft (VBS), 1965
UV_ShowItem_SpecificProfileData-Eigenschaft (VBS), 1965

UV_ShowItem_State-Eigenschaft (VBS), 1965
UV_ShowItem_Station-Eigenschaft (VBS), 1965
UV_ShowItem_SubAddress-Eigenschaft (VBS), 1966
UV_ShowItem_SubSlot-Eigenschaft (VBS), 1966
UV_ShowItem_SubSystem-Eigenschaft (VBS), 1966
UV_ShowItem_Type-Eigenschaft (VBS), 1966

V

ValidateFormatPattern-Eigenschaft (VBS), 2088
ValueAxisAutorange(i)-Eigenschaft (VBS), 1967
ValueAxisBegin(i)-Eigenschaft (VBS), 2089
ValueAxisDecimalPrecision(i)-Eigenschaft (VBS), 2062
ValueAxisEnd(i)-Eigenschaft (VBS), 2089
ValueAxisGridLineInterval(i)-Eigenschaft (VBS), 2146
ValueAxisLabel(i)-Eigenschaft (VBS), 1968
ValueAxisLargeIncrementSize(i)-Eigenschaft (VBS), 2146
ValueAxisScalingType(i)-Eigenschaft (VBS), 1968
ValueAxisShowGridLines(i)-Eigenschaft (VBS), 2063
ValueAxisShowLargeIncrements(i)-Eigenschaft (VBS), 2147
ValueAxisShowSmallIncrements(i)-Eigenschaft (VBS), 2147
ValueAxisSmallIncrementSize(i)-Eigenschaft (VBS), 2148
ValueColumnAlignment(i)-Eigenschaft (VBS), 1969
ValueColumnPrecision(i)-Eigenschaft (VBS), 2064
Variable
 Quality Code, 198
VBS
 Objektmodell, 1292
 Referenz, 1292
VerringereVariable, 29, 1166, 1242
VerticalAlignment-Eigenschaft (VBS), 1969
VerticalGridLines-Eigenschaft (VBS), 1970
VerticalScrollBarPosition-Eigenschaft (VBS), 1971
vfprintf-Funktion, 179
ViewOnly-Eigenschaft (VBS), 1971
Visible-Eigenschaft (VBS), 1972
vsprintf-Funktion, 179

W

WarningColor-Eigenschaft (VBS), 1974
Warning-Eigenschaft (VBS), 1974
WarningLowerLimitColor-Eigenschaft (VBS), 1976

WarningLowerLimit-Eigenschaft (VBS), 1975
 WarningLowerLimitEnabled-Eigenschaft (VBS), 1976
 WarningLowerLimitRelative-Eigenschaft (VBS), 1977
 WarningRangeColor-Eigenschaft (VBS), 1977
 WarningRangeStart-Eigenschaft (VBS), 1978
 WarningRangeVisible-Eigenschaft (VBS), 1979
 WarningUpperLimitColor-Eigenschaft (VBS), 1980
 WarningUpperLimit-Eigenschaft (VBS), 1979
 WarningUpperLimitEnabled-Eigenschaft (VBS), 1980
 WarningUpperLimitRelative-Eigenschaft (VBS), 1981
 WechseleVerbindung, 1156
 Width-Eigenschaft (VBS), 1981
 WinACMPGetVersion, 1238
 WinACMPLeseVersion, 1238
 WinACMPSetStartAtBoot, 1238
 WinACMPSetzeStartModus, 1239
 WinACSetStartMode, 1239
 WinCC
 WinCC MediaControl, 1427
 WindowCloseEnabled-Eigenschaft (VBS), 1984
 WindowMaximizeEnabled-Eigenschaft (VBS), 1984
 WindowMovingEnabled-Eigenschaft (VBS), 1985
 WindowOnTop-Eigenschaft (VBS), 1985
 WindowSizingEnabled-Eigenschaft (VBS), 1986
 WindowSlider-Objekt, 1539
 WindowsStyle-Eigenschaft (VBS), 1986
 WLanQualityView-Objekt, 1541
 WriteTag, 2271

X

XAxisAdd-Eigenschaft (VBS), 1987
 XAxisAlignment-Eigenschaft (VBS), 1987
 XAxisAutoPrecisions-Eigenschaft (VBS), 1988
 XAxisAutoRange(i)-Eigenschaft (VBS), 1988
 XAxisBegin(i)-Eigenschaft (VBS), 2065
 XAxisBeginValue-Eigenschaft (VBS), 1989
 XAxisColor-Eigenschaft (VBS), 1989
 XAxisCount-Eigenschaft (VBS), 1990
 XAxisDecimalPrecision(i)-Eigenschaft (VBS), 2149
 XAxisEnd(i)-Eigenschaft (VBS), 2066
 XAxisEndValue-Eigenschaft (VBS), 1990
 XAxisExponentialFormat-Eigenschaft (VBS), 1991
 XAxisGridLineInterval(i)-Eigenschaft (VBS), 2149
 XAxisIndex-Eigenschaft (VBS), 1991
 XAxisInTrendColor-Eigenschaft (VBS), 1992
 XAxisLabel(i)-Eigenschaft (VBS), 1992

XAxisLargeIncrementSize(i)-Eigenschaft (VBS), 2150
 XAxisMode(i)-Eigenschaft (VBS), 2150
 XAxisName-Eigenschaft (VBS), 1993
 XAxisPrecisions-Eigenschaft (VBS), 1993
 XAxisRemove-Eigenschaft (VBS), 1994
 XAxisRepos-Eigenschaft (VBS), 1994
 XAxisScalingType(i)-Eigenschaft (VBS), 1994
 XAxisShowGridLines(i)-Eigenschaft (VBS), 2066
 XAxisShowLargeIncrements(i)-Eigenschaft (VBS), 2151
 XAxisShowSmallIncrements(i)-Eigenschaft (VBS), 2151
 XAxisSmallIncrementSize(i)-Eigenschaft (VBS), 2152
 XAxisTrendWindow-Eigenschaft (VBS), 1995
 XAxisVisible-Eigenschaft (VBS), 1995
 XDataLogTag(i)-Eigenschaft (VBS), 2152
 XOnlineTag(i)-Eigenschaft (VBS), 2153

Y

YAxisAdd-Eigenschaft (VBS), 1996
 YAxisAlignment-Eigenschaft (VBS), 1996
 YAxisAutoPrecisions-Eigenschaft (VBS), 1997
 YAxisAutoRange-Eigenschaft (VBS), 1997
 YAxisBeginValue-Eigenschaft (VBS), 1998
 YAxisColor-Eigenschaft (VBS), 1998
 YAxisCount-Eigenschaft (VBS), 1999
 YAxisEndValue-Eigenschaft (VBS), 1999
 YAxisExponentialFormat-Eigenschaft (VBS), 2000
 YAxisIndex-Eigenschaft (VBS), 2000
 YAxisInTrendColor-Eigenschaft (VBS), 2001
 YAxisLabel-Eigenschaft (VBS), 2001
 YAxisName-Eigenschaft (VBS), 2002
 YAxisPrecisions-Eigenschaft (VBS), 2002
 YAxisRemove-Eigenschaft (VBS), 2002
 YAxisRename-Eigenschaft (VBS), 2003
 YAxisRepos-Eigenschaft (VBS), 2003
 YAxisScalingType-Eigenschaft (VBS), 2004
 YAxisTrendWindow-Eigenschaft (VBS), 2004
 YAxisVisible-Eigenschaft (VBS), 2005
 YDataLogTag(i)-Eigenschaft (VBS), 2154
 YOnlineTag(i)-Eigenschaft (VBS), 2154

Z

ZeigeAnmeldedialog, 1269
 ZeigeHilfetext, 1228
 ZeigeMeldefenster, 1227
 ZeigePopupBild, 1229

ZeigeSlideInBild, 1230
ZeigeSoftwareVersion, 1230
ZeigeSystemDiagnosefenster, 1232
ZeigeSystemmeldung, 1231
ZeroPoint-Eigenschaft (VBS), 2005
ZoneLabelView-Objekt, 1542
ZoneQualityView-Objekt, 1543
Zoomable-Eigenschaft (VBS), 2090
ZoomArea, 2272
ZoomFactor-Eigenschaft (VBS), 2006
ZoomInOut, 2272
ZoomInOutTime, 2273
ZoomInOutValues, 2273
ZoomInOutX, 2274
ZoomInOutY, 2274
ZoomMove, 2275