

SIMATIC Visualization Architect

Systemhandbuch

<u>Security-Hinweise</u>	1
<u>Datenschutz</u>	2
<u>Grundlagen</u>	3
<u>Installation</u>	4
<u>Elemente und Grundeinstellungen</u>	5
<u>Arbeiten mit SiVArc</u>	6
<u>SiVArc Openness</u>	7
<u>Referenz</u>	8
<u>Tooltipps</u>	9

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	Security-Hinweise	9
2	Datenschutz	11
3	Grundlagen	13
3.1	Einführung	13
3.2	Einsatzmöglichkeiten	14
3.3	Beispiel: Einsatz von SiVArc zum Generieren der Visualisierung	15
3.4	Beispiel: Einsatz von SiVArc zum Generieren von Variablen	18
3.5	Bedien- und Beobachtungslösung mit SiVArc projektieren	19
4	Installation	23
4.1	SiVArc installieren	23
5	Elemente und Grundeinstellungen	25
5.1	SiVArc-Einstellungen	25
5.2	User Management Control (UMAC) in SiVArc	26
5.3	SiVArc-Editoren	27
5.3.1	Editor "Bildregeln"	27
5.3.2	Editor "Variablenregeln"	29
5.3.3	Editor "Textlistenregeln"	30
5.3.4	Editor "Melderegeln"	31
5.3.5	Editor "SiVArc-Ausdrücke"	33
5.3.6	Editor "Kopierregeln"	34
5.3.7	Editor "Generiermatrix"	36
5.3.8	Expression Resolver	39
5.3.9	Generierübersicht	43
5.3.10	Energy Suite Generierung	45
5.4	SiVArc in HMI-Editoren	46
5.4.1	Eigenschaftskonfigurator	46
5.4.2	Registerkarte "SiVArc-Eigenschaften"	49
5.4.3	Registerkarte "SiVArc-Ereignisse"	52
5.4.4	Registerkarte "SiVArc-Animationen"	54
5.4.5	Registerkarte "Generierübersicht"	55
5.5	SiVArc in PLC-Editoren	56
5.5.1	Unterstützung für Software-Einheiten	56
5.5.2	SiVArc-Unterstützung für Programmbausteine mit Hilfe von Structured Control Language	57
6	Arbeiten mit SiVArc	59
6.1	Layout planen	59
6.1.1	Positionierungsschemata	59
6.1.1.1	Übersicht zur Positionierung von generierten Objekten	59
6.1.1.2	Positionierung nach definierten Schemata	61

6.1.1.3	Feste Positionierung des generierten Objekts	71
6.1.1.4	Freie Positionierung	72
6.1.1.5	Tiefenstaffelung	73
6.1.2	Überlaufmechanismen	75
6.1.3	Unterstützte Geräte	79
6.1.4	Planen eines Layouts	79
6.1.5	Benutzerdefiniertes Positionierungsschema anlegen	81
6.1.6	Überlaufbilder ohne Bildobjekte projektieren	84
6.1.7	Beispiel: Einsatz eines Layouts mit freier Positionierung	85
6.1.8	Beispiel: Einsatz von Überlaufmechanismen	86
6.1.9	Beispiel: Einsatz eines dynamischen Layouts	88
6.1.10	Beispiel: Einsatz eines kombinierten Layouts	91
6.1.11	Beispiel: Einsatz generierter Bildnavigation	92
6.2	Erstellen von Generiervorlagen	93
6.2.1	Generiervorlagen in SiVArc	93
6.2.2	Unterstützte HMI-Objekte	99
6.2.3	Quellen für Texte	101
6.2.3.1	Übersicht zu Textquellen im SiVArc-Projekt	101
6.2.3.2	SiVArc-Texte	104
6.2.4	Unterstützte Objekte im Anwenderprogramm	106
6.2.5	SiVArc-Scripting	107
6.2.6	SiVArc-Ausdruck	109
6.2.6.1	Übersicht zu SiVArc-Ausdrücken	109
6.2.6.2	SiVArc-Variablen	110
6.2.7	Anforderungen an eine Generiervorlage	112
6.2.8	Parametrierungskonzepte	115
6.2.8.1	Beispiel: Unified-Bildbausteine generieren	115
6.2.8.2	Beispiel: Erstellen eines Parametrierungskonzepts	115
6.2.8.3	Zuordnung Baustein und Generiervorlage	116
6.2.8.4	Aufbau von SiVArc-Ausdrücken	119
6.2.9	Einfluss des Anwenderprogramms auf eine Generiervorlage	121
6.2.10	Einfluss der Mehrsprachigkeit auf eine Generiervorlage	124
6.2.11	Ablagestrategien für generierte Objekte	126
6.2.12	Beispiel: Erreichen von hoher Flexibilität	129
6.2.13	Beispiel: Erreichen von hoher Wiederverwendbarkeit	131
6.2.14	Beispiel: Generiervorlage für Bildfenster erstellen	131
6.2.15	Beispiel: Generierung einer Bildvorlage mithilfe der Kopierregel	134
6.2.16	Beispiel: Generierung eines HMI-Bilds mithilfe von Vorlageneigenschaften	135
6.2.17	Beispiel: Generiervorlage mit Animation erstellen	137
6.2.18	Beispiel: Generiervorlage mit Ereignisprojektierung erstellen	139
6.2.19	Beispiel: Generiervorlage mit Skriptprojektierung erstellen	140
6.2.20	Beispiel: Generiervorlagen für Textlisten erstellen	142
6.2.21	Beispiel: Generiervorlage für eine Textliste für Bausteinparameter erstellen	145
6.2.22	Beispiel: Pop-up-Bilder und deren Verwendung generieren	147
6.2.23	Beispiel: Bildbausteine mit Animationen generieren	149
6.2.24	Beispiel: Animation "Position" für Bildbausteine generieren	154
6.2.25	Beispiel: Erstellung von Generiervorlagen für Kurvenanzeigen	155
6.2.26	Beispiel: Hardwarekonfiguration für Bilder	157
6.2.27	Beispiel: Hardwarekonfiguration für Meldungen	159
6.2.28	Generiervorlage für ein Bild erstellen	160
6.3	Regeln erstellen	161

6.3.1	SiVArc-Regeln	161
6.3.2	Erstellen von SiVArc-Regeln	165
6.3.3	Einsatz von SiVArc-Scripting in SiVArc-Regeln	168
6.3.4	Verarbeitung von Regeln	170
6.3.5	Generieren von Variablen	173
6.3.6	Einsatz von Kopierregeln	177
6.3.7	Zusammenhang zwischen SiVArc-Ausdrücken und Bedingungen	180
6.3.8	Prinzip der Variablengenerierung	181
6.3.9	Vermeiden von Konflikten beim Generieren	183
6.3.10	Variablenregel anlegen	186
6.3.11	Bildregel anlegen	188
6.3.12	Textlistenregel anlegen	189
6.3.13	Melderegeln erstellen	190
6.3.14	Kopierregel anlegen	192
6.3.15	Beispiel: Bildregel mit Bedingung anlegen	194
6.3.16	Beispiel: Bild- und Textlistenregel organisieren	195
6.3.17	Beispiel: Variablenamen anpassen	197
6.3.18	SiVArc-Regeln bearbeiten und verwalten	199
6.3.19	SiVArc-Regeln exportieren und importieren	201
6.3.20	Know-how-Schutz für ein SiVArc-Projekt einrichten	204
6.4	Visualisierung generieren	205
6.4.1	Grundlagen zur Generierung	205
6.4.2	Nachträgliche Änderungen	209
6.4.2.1	Ändern von generierten Objekten	209
6.4.2.2	Ändern von SiVArc-Regeln	212
6.4.2.3	Kennzeichnungen im SiVArc-Projekt	214
6.4.3	Visualisierung generieren	215
6.5	Ergebnis prüfen	218
6.5.1	Prüfen des Ergebnisses	218
6.5.2	Einsatz der Generiermatrix	221
6.5.3	Beispiel: Generiermatrix einsetzen	224
7	SiVArc Openness	227
7.1	Einleitung	227
7.2	SiVArc-Diensteigenschaften	227
7.3	Kopieren von Regeln oder Gruppen aus der Bibliothek	229
7.4	Suchen von Bildregeln oder Bildregelgruppen	231
7.5	Löschen von Regeln und Regelgruppen	232
7.6	UMAC-Einrichtung für Openness	233
7.7	SiVArc-Generierung	233
8	Referenz	237
8.1	SiVArc-Objekte	237
8.1.1	PLC-Variablen	237
8.1.2	Peripheriegerät	237
8.1.3	Software Units	237
8.1.4	Objekthierarchie	238
8.1.5	Block (Panels, Comfort Panels, RT Advanced, RT Professional)	239

8.1.6	DB (Panels, Comfort Panels, RT Advanced, RT Professional)	240
8.1.7	HMIApplication (Panels, Comfort Panels, RT Advanced, RT Professional)	241
8.1.8	HMIDevice (Panels, Comfort Panels, RT Advanced, RT Professional)	242
8.1.9	HMITag (Panels, Comfort Panels, RT Advanced, RT Professional)	242
8.1.10	LibraryObject (Panels, Comfort Panels, RT Advanced, RT Professional)	243
8.1.11	ModuleBlock (Panels, Comfort Panels, RT Advanced, RT Professional)	244
8.1.12	Parameters (Panels, Comfort Panels, RT Advanced, RT Professional)	245
8.1.13	S7Control (Panels, Comfort Panels, RT Advanced, RT Professional)	246
8.1.14	SubModuleBlock (Panels, Comfort Panels, RT Advanced, RT Professional)	246
8.1.15	StructureBlock (Panels, Comfort Panels, RT Advanced, RT Professional)	248
8.1.16	TagNaming (Panels, Comfort Panels, RT Advanced, RT Professional)	249
8.2	SiVArc-Objekteigenschaften	250
8.2.1	Assigned	250
8.2.2	Comment	250
8.2.3	FolderPath	251
8.2.4	HMITagPrefix	252
8.2.5	IndexEndChar	252
8.2.6	IndexStartChar	252
8.2.7	InitialValue	253
8.2.8	Name	253
8.2.9	NetworkComment	253
8.2.10	NetworkTitle	254
8.2.11	Number	255
8.2.12	SeparatorChar	255
8.2.13	SymbolComment	255
8.2.14	SymbolicName	256
8.2.15	Title	257
8.2.16	Type	257
8.2.17	Value	258
8.2.18	Version	258
8.3	SiVArc-Objekteigenschaften	259
8.4	Funktionen	261
8.4.1	Funktionen in SiVArc	261
8.4.2	Funktion "Contains"	261
8.4.3	Funktion "EndsWith"	261
8.4.4	Funktion "Format"	262
8.4.5	Funktion "FormatNumber"	262
8.4.6	Funktion "InStr"	264
8.4.7	Funktion "IsDefined"	265
8.4.8	Funktion "LBound"	265
8.4.9	Funktion "Left"	266
8.4.10	Funktion "Len"	266
8.4.11	Funktion "LTrim"	267
8.4.12	Funktion "Max"	267
8.4.13	Funktion "Mid"	267
8.4.14	Funktion "Min"	268
8.4.15	Funktion "Replace"	268
8.4.16	Funktion "Right"	269
8.4.17	Funktion "RTrim"	269
8.4.18	Funktion "Split"	269
8.4.19	Funktion "StartsWith"	270

8.4.20	Funktion "StrComp"	271
8.4.21	Funktion "TrailNum"	271
8.4.22	Funktion "Trim"	272
8.4.23	Funktion "UBound"	272
8.5	Operatoren	272
8.6	String-Indizierung	274
8.7	If-Bedingungen	275
8.8	Unterstützte Datentypen für PLC-Variablen	275
8.9	Unterstützte Systemfunktionen für Bildbausteine	277
8.10	Anzeige in den SiVArC-Editoren bearbeiten	277
8.11	Bildlegende	279
9	Tooltips	281
9.1	UMAC	281
9.1.1	CTHMISivarc	281
	Index	283

Security-Hinweise

Security-Hinweise

Siemens bietet Produkte und Lösungen mit Industrial-Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke vor Cyberbedrohungen zu schützen, ist es notwendig, ein ganzheitliches, modernes Industrial Security-Konzept umzusetzen und kontinuierlich zu pflegen. Die Produkte und Lösungen von Siemens stellen nur ein Element eines solchen Konzepts dar.

Der Kunde ist dafür verantwortlich, unbefugten Zugang zu seinen Anlagen, Systemen, Maschinen und Netzwerken zu verhindern. Solche Systeme, Maschinen und Komponenten dürfen nur im Bedarfsfall, im erforderlichen Umfang und bei entsprechenden Sicherheitsmaßnahmen (z. B. Firewalls und/oder Netzwerksegmentierung) mit einem Unternehmensnetzwerk oder dem Internet verbunden werden.

Außerdem sollten die Richtlinien von Siemens zu geeigneten Sicherheitsmaßnahmen berücksichtigt werden. Weitere Informationen zur Industrial Security finden Sie unter <http://www.siemens.com/industrialsecurity> (<http://support.automation.siemens.com>)

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie sicherer zu machen. Siemens empfiehlt, Produkt-Updates bei Verfügbarkeit unbedingt anzuwenden und immer die neuesten Produktversionen zu benutzen. Die Verwendung von nicht mehr unterstützten Produktversionen und die Nicht-Anwendung der neuesten Updates kann das Risiko des Kunden gegenüber Cyberbedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, melden Sie sich für den Siemens Industrial Security RSS-Feed unter folgender URL an:

<http://www.siemens.com/industrialsecurity>. (<http://www.industry.siemens.com/topics/global/de/industrial-security/Seiten/Default.aspx>)

Netzlaufwerk

Stellen Sie sicher, dass Netzlaufwerke in Ihrer Netzwerkinfrastruktur sowie Rechner vor unbefugtem Zugriff geschützt sind.

Kommunikation über Ethernet

Bei der Ethernet-basierten Kommunikation sind die Endanwender selbst für die Sicherheit ihres Datennetzwerks verantwortlich. Die einwandfreie Funktion des Geräts kann nicht unter allen Umständen garantiert werden. Zielgerichtete Angriffe beispielsweise können zur Überlastung des Geräts führen.

Datenschutz

Siemens beachtet die Datenschutzrichtlinien, insbesondere das Prinzip der Datensparsamkeit (eingebauter Datenschutz). Das bedeutet für dieses SIMATIC-Produkt, dass das Produkt keine personenbezogenen Daten verarbeitet/speichert.

Grundlagen

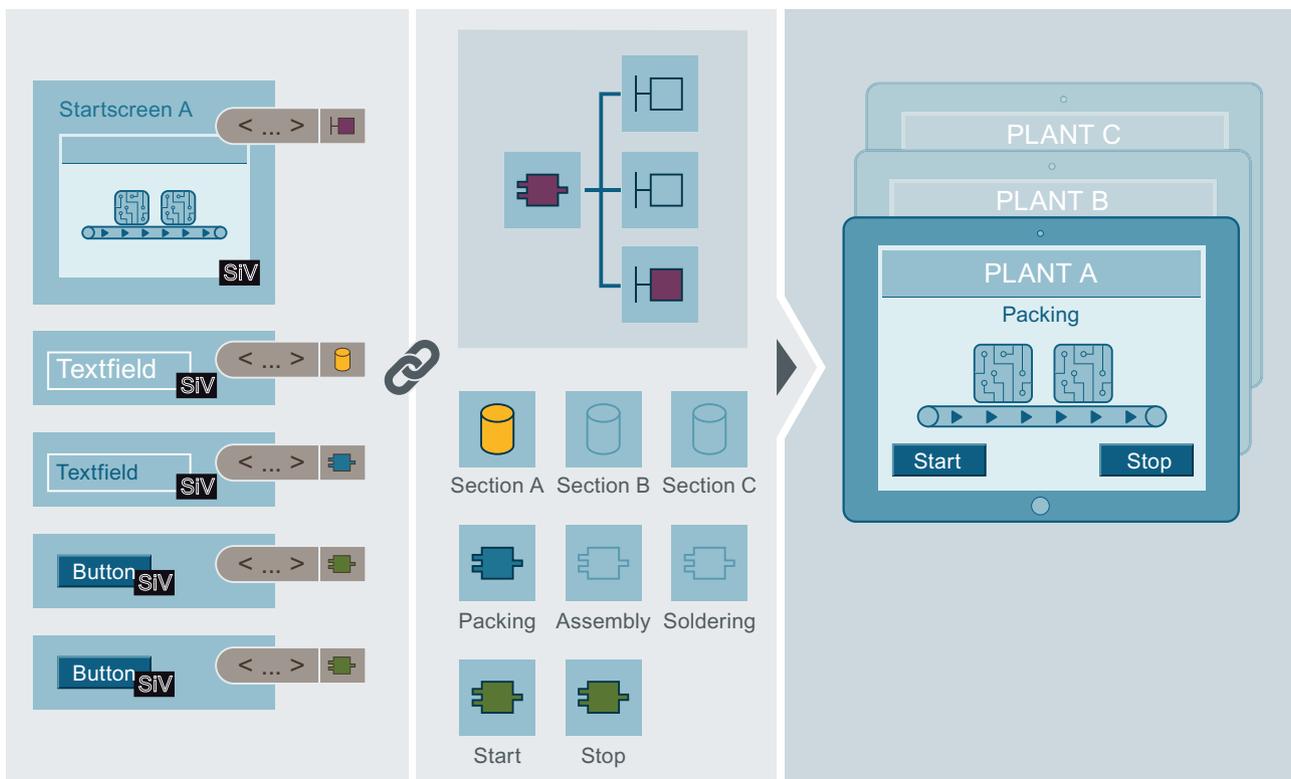
3.1 Einführung

Was ist SiVArc?

SiVArc (SIMATIC WinCC Visualization Architect) ist ein Optionspaket im TIA-Portal.

Mithilfe von SiVArc generieren Sie aus Programmbausteinen und Generiervorlagen die Visualisierung für jeweils mehrere Bediengeräte und PLCs.

Anhand von Generierregeln legen Sie fest, welche HMI-Objekte für welche Bausteine und Geräte generiert werden.



Funktionsumfang

Mit SiVArc generieren Sie folgende HMI-Objekte aus Steuerungsdaten:

- Bilder, Bildbausteine und eine Auswahl von Anzeige- und Bedienobjekten
- Externe Variablen
- HMI-Textlisten

Ohne Bezug zum Steuerungsprogramm können Sie eine Auswahl an Objekten aus Ihrer WinCC-Projektbibliothek mit SiVArC in Ihr WinCC-Projekt generieren oder als Instanz verwenden.

Für die Generierung verwenden Sie Generiervorlagen aus der Projektbibliothek oder der globalen Bibliothek.

SiVArC kann zeitgleich die Visualisierung für mehrere Bediengeräte, mehrere PLCs und Geräte-Proxys generieren. Während Sie mit SiVArC die Visualisierung generieren, können Sie mit dem TIA-Portal in einer zweiten Instanz weiterarbeiten. Mithilfe von SiVArC und der TIA Portal-Option "TIA Portal Multiuser" können Sie auch mit mehreren Benutzern an einem SiVArC-Projekt arbeiten.

In den SiVArC-Editoren können Sie System- oder Skript-Funktionen kopieren, einfügen, rückgängig machen und wiederholen. Mithilfe der globalen Suche können Sie nach System- oder Skript-Funktionsnamen oder deren Parameterwerten suchen. Die Suchergebnisse enthalten eine "Gehe zu"-Verknüpfung, über die man zu den gesuchten Elementen gelangt.

Die SiVArC-Einstellungen sind als Teil der globalen Einstellungen verfügbar.

Weitere Informationen zu Automatisierungsaufgaben mit SiVArC finden Sie im YouTube-Kanal von Siemens (<https://www.youtube.com/watch?v=vdq98OwMEnE>).

Weitere Informationen zu SiVArC finden Sie unter Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/de/view/109751096>).

Siehe auch

Unterstützte Objekte im Anwenderprogramm (Seite 106)

Bedien- und Beobachtungslösung mit SiVArC projektieren (Seite 19)

Übersicht zu SiVArC-Ausdrücken (Seite 109)

Bildlegende (Seite 279)

Erstellen von Generiervorlagen (Seite 93)

3.2 Einsatzmöglichkeiten

Überblick

Sie setzen SiVArC in Automatisierungslösungen mit hohem Standardisierungsgrad ein.

SiVArC unterstützt den Projektteur während des Engineerings bei folgenden Aufgaben:

- Automatische Generierung der Visualisierung einschließlich Prozessanschluss
- Einheitliches Layout von Benutzeroberflächen
- Konsistente Benennung von Bedienelementen
- Strukturierte Ablage von Projektierungsdaten

SiVArc bietet auch Unterstützung in der Betriebsphase:

- **Inbetriebnahme**
SiVArc ist auch während der Inbetriebnahme nützlich, weil der Inbetriebnahme-Ingenieur mithilfe einer Generiermatrix auch ohne spezielle SiVArc-Kenntnisse kurzfristig Änderungen am Projekt vornehmen kann.
- **Anpassungen**
Um Änderungen auf ein ganzes Projekt anzuwenden, müssen Sie nur mit SiVArc zentrale Vorlagen anpassen.
- **Anlagenwartung**
Die Möglichkeit der Generierung einzelner Geräte bedeutet auch, dass ein Austausch von Bediengeräten problemlos möglich ist.

Ebenso eignet sich SiVArc für die Erhöhung des Standardisierungsgrads in Ihrem Projekt und zur kontinuierlichen Optimierung Ihrer Projekte.

Vorteile

Der grundsätzliche Mehrwert von SiVArc im Vergleich zur konventionellen Konfiguration der Visualisierung beruht auf den folgenden SiVArc-Prinzipien:

- Die generierte Visualisierung behält ihren Bezug zum SiVArc-Projekt. Mit SiVArc vorgenommene Anpassungen und Optimierungen sorgen für eine hoch leistungsfähige und klar strukturierte Datengrundlage.
- Die Visualisierung ist direkt mit dem Anwenderprogramm verknüpft. Änderungen im Anwenderprogramm erfordern nur minimale Anpassungen im WinCC-Projekt.
- Layout, Design und die konsistente Benennung im Display werden zentral über STEP 7 und WinCC gesteuert.

Anforderungen an den Projektteur

Für den Einsatz von SiVArc sind folgende Vorkenntnisse erforderlich:

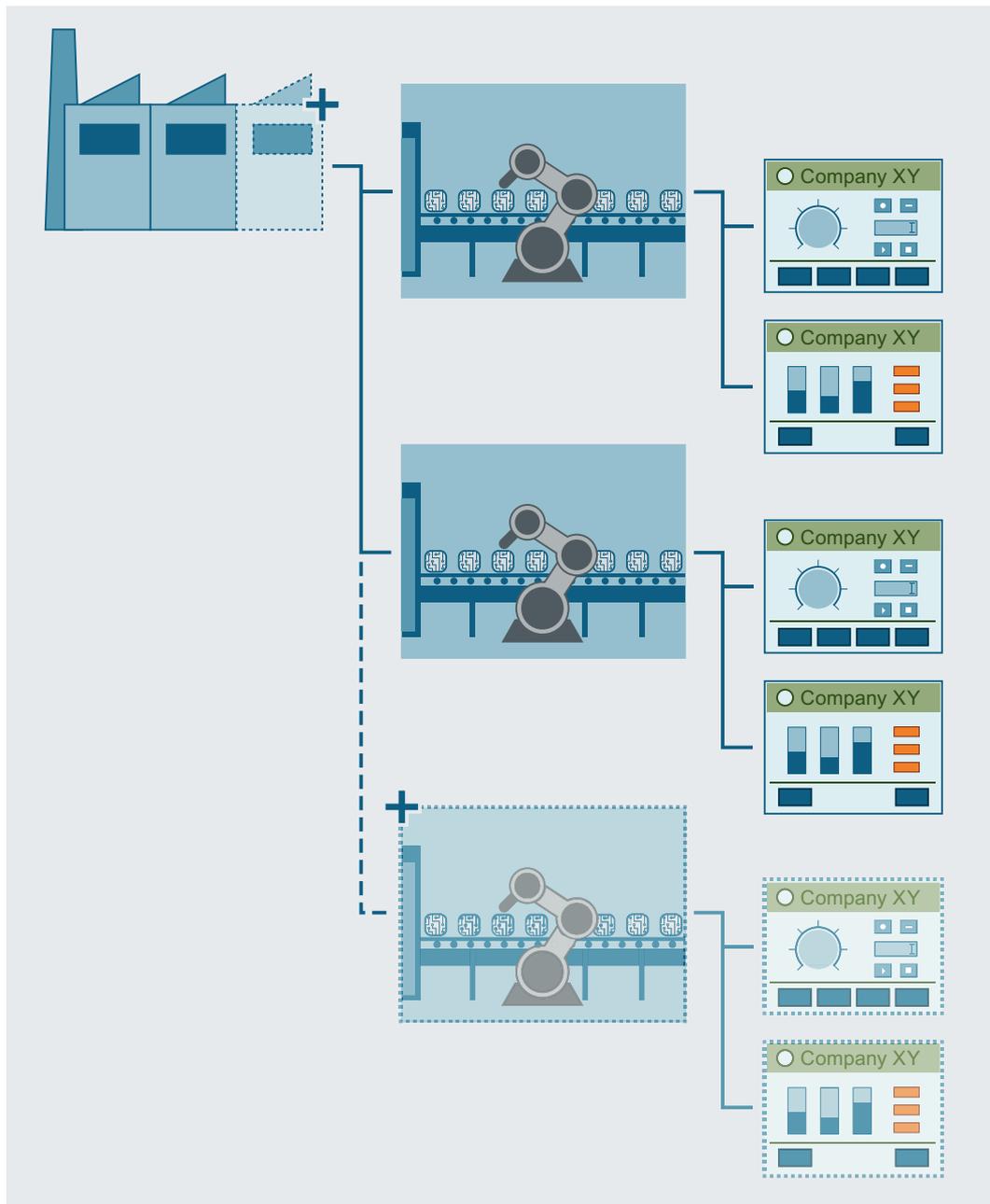
- Sie haben Projektierungserfahrung in STEP 7 und WinCC.
- Sie verfügen über Grundkenntnisse in Visual Basic Script (VBS).

3.3 Beispiel: Einsatz von SiVArc zum Generieren der Visualisierung

Beispielszenario

Eine bestehende Fabrik für Leiterplatten wird um eine dritte Fertigungsstrecke erweitert. Das Unternehmen beauftragt ein externes Ingenieurbüro mit der Visualisierung der Erweiterung auf Basis des bestehenden Steuerungsprogramms und des bestehenden Visualisierungskonzepts. Die Aufgabe des Ingenieurbüros enthält folgende Anforderungen:

- Der Kunde ist neu. Kein Vorprojekt existiert.
- Das neue Corporate Design des Kunden soll in die Visualisierung eingearbeitet werden.
- Der Kunde möchte die Standardisierung im Unternehmen optimieren.



Konzeption einer Parametrierungslösung

Das Ingenieurbüro entscheidet für die Umsetzung der Aufgabe ein bestehendes SiVArc-Musterprojekt anzupassen. Dazu beauftragt es jeweils einen PLC-Programmierer und einen Visualisierungsexperten mit der Analyse des Anwenderprogramms und des Visualisierungskonzepts.

Gemeinsam definieren Sie Folgendes:

- Anzahl der Layout-Vorlagen abhängig von den verwendeten Bediengeräten
- Erforderliche externe Variablen

- Namenskonzepte zur Benennung von externen Variablen
- Textquellen im Anwenderprogramm, die in der Visualisierung verwendet werden
- Zuordnung der Programmblöcke zu Generiervorlagen
- Aufbau der SiVArc-Ausdrücke in den Generiervorlagen
- Ablagestrukturen im SiVArc-Projekt

Im Anschluss ermittelt der Visualisierungsexperte die Anzahl und Art der notwendigen Generiervorlagen, SiVArc-Regeln und SiVArc-Variablen.

Umsetzung der Parametrierungslösung

Der PLC-Programmierer passt das Anwenderprogramm an die konzipierte Lösung an:

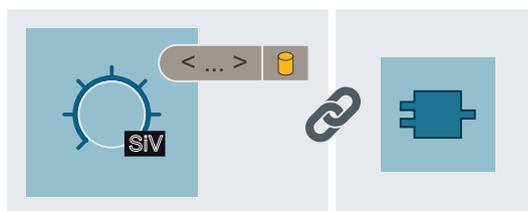
- PLC-Variablen auf "Erreichbar in HMI" setzen
- Textquellen in STEP 7 prüfen und gegebenenfalls konsistent machen
- Ablage der Programmbausteine in der Projektnavigation optimieren
- Bestehende Bibliotheken erweitern

Der Visualisierungsexperte setzt das Corporate Design des Kunden für mehrere Bediengeräte auf Bildvorlagen um.

Auf Basis der Bildvorlagen wird pro Gerät ein Positionierungsschema erzeugt.

Die Generiervorlagen des Beispielprojekts für Standardobjekte werden an das Visualisierungskonzept angepasst.

Die Generiervorlagen und Funktionsbausteine werden in den Bildregeln miteinander verknüpft.



SIV

SiVArc-Generiervorlage



SiVArc-Eigenschaft mit referenzierter Textquelle, z. B. ein Datenbaustein



Funktionsbaustein

Ergebnis

Auf Basis eines SiVArc-Musterprojekts ist ein neues, kundenspezifisch modifiziertes und agiles SiVArc-Projekt entstanden. Weitere Erweiterungen der Anlage und des Anwenderprogramms erfordern jetzt nur noch minimale Eingriffe im SiVArc-Projekt.

3.4 Beispiel: Einsatz von SiVArc zum Generieren von Variablen

Beispielszenario

Bei einem Anlagenbauer kommt es bei der Inbetriebnahme häufig zu ungeplanten Verzögerungen. Analysen ergeben, dass bestehende Namenskonventionen für Variablen nicht konsequent umgesetzt werden. Das erneute Anlegen der Variablen belastet das Speichervolumen der Bediengeräte.

Das Unternehmen wendet sich an ein Ingenieurbüro, das die Variablennamen standardisieren und neu verknüpfen soll.

Die Ausfallzeiten sollen minimiert und Speicherplatz auf den Bediengeräten frei gemacht werden.

Lösungskonzept

Das Ingenieurbüro analysiert das Anwenderprogramm und setzt die erforderlichen Variablen auf "Erreichbar in HMI".

Abhängig von der Art der verwendeten PLC-Variablen, UDTs oder Arrays projiziert der Ingenieur die Synchronisation der Variablennamen.

SiVArc startet die Generierung der Variablen. Dabei generiert SiVArc nur die für die Visualisierung erforderlichen Variablen.

Auf Basis des Namenskonzepts für Variablen werden über SiVArc-Ausdrücke die gewünschten Variablennamen erzeugt.

Hinweis

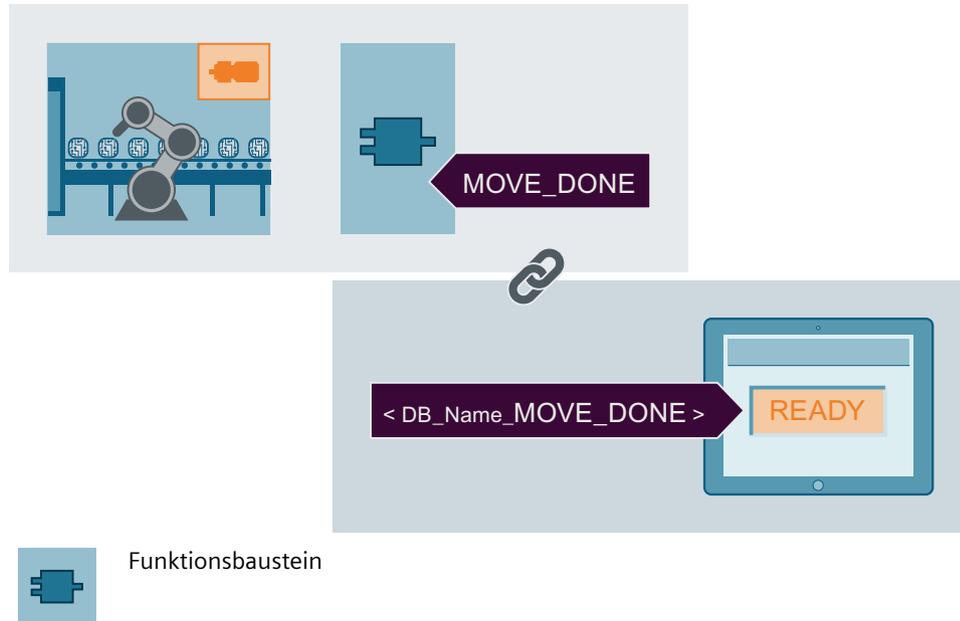
Variablennamen

WinCC unterstützt weniger Zeichen als STEP 7. Wenn Sie im PLC-Variablennamen ein von WinCC nicht unterstütztes Zeichen verwenden, wird das Zeichen beim Generieren des Namens der externen Variablen gelöscht. Dadurch können mehrere Variablen mit dem gleichen Namen entstehen. Dadurch wird ein Fehler erzeugt, weil SiVArc keine gleichnamigen Variablen generiert.

Verwenden Sie bei der Namensvergabe für PLC-Variablen nur von WinCC unterstützte Zeichen.

Ergebnis

Die erforderlichen Variablen sind einheitlich benannt. Der Bezug zur PLC-Variablen ist an der Verschaltungsstelle im WinCC-Projekt ablesbar.



Im WinCC-Projekt existieren nur die wirklich erforderlichen Variablen. Eine Weiterbearbeitung und kontinuierliche Bereinigung der Variablen ist mit SiVArc möglich.

Siehe auch

Prinzip der Variablengenerierung (Seite 181)

3.5 Bedien- und Beobachtungslösung mit SiVArc projektieren

Einleitung

Die Projektierung von HMI-Lösungen mit SiVArc erfordert ein standardisiertes Projekt. Je standardisierter ein Projekt ist, umso einfacher und effektiver kann SiVArc für die Visualisierung eingesetzt werden.

Voraussetzung

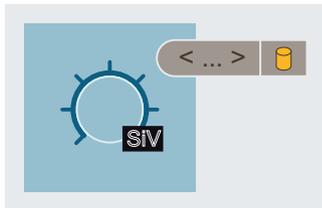
- Die Anlage ist eine Standardanlage.
- Das strukturierte Anwenderprogramm ist erstellt.
- Ein Visualisierungs- und Bedienungskonzept ist erstellt.
- Auf Standardbausteine im Anwenderprogramm kann über Bibliotheken zugegriffen werden.

- Auf Bildbausteine für Standardanwendungen kann über Bibliotheken zugegriffen werden.
- Das Projekt ist standardisiert und übertragbar.

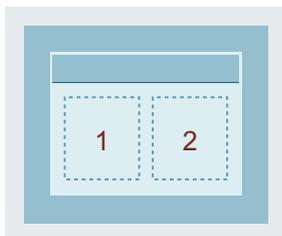
Vorgehensweise

Um HMI-Lösungen mit SiVArc zu generieren, gehen Sie wie folgt vor:

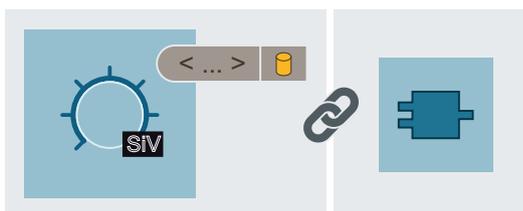
1. Entwerfen Sie die Darstellung.
 - Welche HMI-Geräte werden benutzt?
 - Wie wird das Corporate Design im Bild berücksichtigt?
 - Wie viele Generiervorlagen für Bilder werden benötigt?
 - Wie viele Positionierungsschemata werden benötigt?
2. Legen Sie fest, welche externen Variablen von SiVArc generiert werden sollen.
3. Erstellen Sie die Generiervorlagen für HMI-Objekte und speichern Sie diese in der Bibliothek.



4. Erstellen Sie die Positionierungsschemata für Bilder und speichern Sie diese in der Bibliothek.



5. Erstellen Sie Bildregeln für die Verknüpfung der Generiervorlagen mit den Funktionsbausteinen.

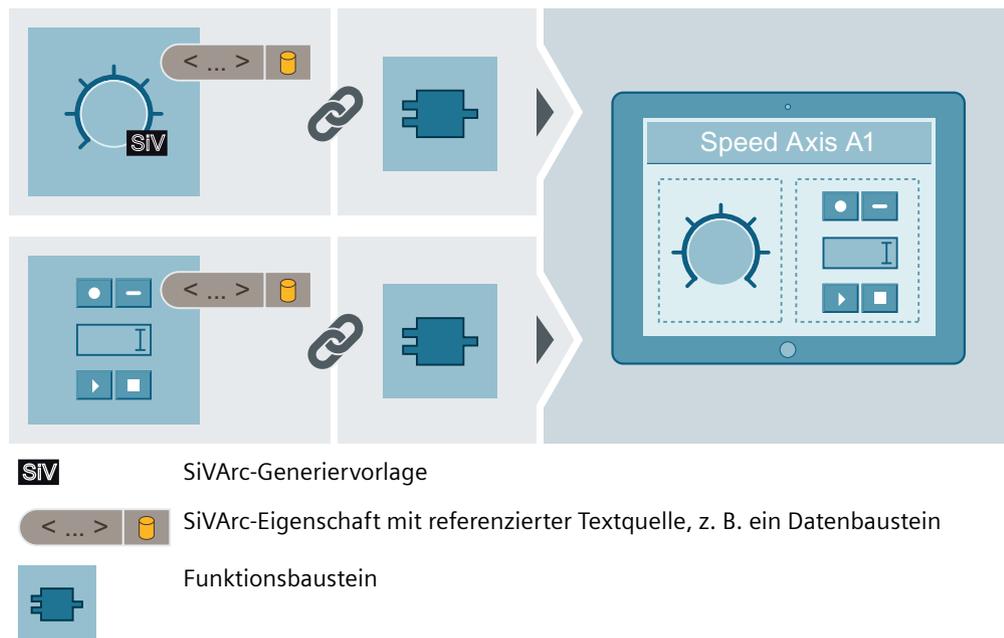


6. Erstellen Sie Variablenregeln für die Speicherung generierter Variablen.
7. Erstellen Sie Kopierregeln zum Kopieren erfasster HMI-Objekte aus der Bibliothek in das Projekt.
8. Legen Sie die Textlisteneinträge fest.
9. Generieren Sie die vollständige Visualisierung oder nur die Visualisierung ausgewählter Geräte.

Ergebnis

Die generierten HMI-Objekte sind in der Projektnavigation erstellt und als generierte Objekte markiert.

Die generierten Bildobjekte werden anhand ihrer Positionierungsschemata in den generierten Bildern angeordnet.



Weitere Verarbeitung

Hinweis

Spätere Namensänderungen generierter Objekte

Wurde der Name eines generierten HMI-Objekts geändert, wird das Objekt bei der nächsten SiVArc-Generierung erneut erstellt und verschaltet. Das umbenannte Objekt bleibt verfügbar.

Ändern Sie die Namen generierter Objekte nur im Anwenderprogramm.

Starten Sie die Generierung nach jeder Änderung im SiVArc-Projekt oder im Anwenderprogramm neu.

SiVArc verarbeitet die geänderten Informationen, ersetzt die bestehende Generierung und generiert bei Bedarf zusätzliche HMI-Objekte.

Für die Suche nach SiVArc-konfigurierten Ausdrücken können Sie die globale Suche verwenden.

Siehe auch

Layout planen (Seite 59)

Erstellen von Generiervorlagen (Seite 93)

Regeln erstellen (Seite 161)

Visualisierung generieren (Seite 205)

Installation

4.1 SiVArc installieren

Einleitung

Das Setup-Programm des Add-on-Pakets "SiVArc" startet nach dem Einlegen des Installationsdatenträgers in das entsprechende Laufwerk.

Um SiVArc zu installieren, benötigen Sie eine gültige Lizenz. Um Ihre License Keys zu verwalten, verwenden Sie den "Automation License Manager".

Hinweis

Versionskompatibilität

Ihre Version von SiVArc ist nur kompatibel mit der entsprechenden Version von STEP 7 sowie WinCC Professional, WinCC Advanced oder WinCC Unified.

Um Ihre TIA-Portal-Version hochzurüsten, müssen Sie auch Ihre SiVArc-Version hochrüsten und umgekehrt. Wenn Sie WinCC oder STEP 7 deinstallieren, wird auch SiVArc deinstalliert.

Um mit verschiedenen TIA-Portal-Versionen zu arbeiten, wählen Sie eine Side-By-Side-Installation.

Voraussetzung

- STEP7 Professional V17.0 ist installiert.
- SIMATIC WinCC Professional V17.0 oder SIMATIC WinCC Advanced, SIMATIC WinCC Unified V17.0 ist installiert.

Vorgehensweise

Zum Installieren des Add-on-Pakets von "SiVArc" gehen Sie wie folgt vor:

1. Legen Sie den Installationsdatenträger in das Laufwerk ein.
Um das Setup bei Bedarf manuell zu starten, doppelklicken Sie im Explorer auf die Datei "Start.exe".
2. Wählen Sie eine Installationssprache und klicken Sie "Weiter".
3. Wählen Sie das gewünschte Produkt und klicken Sie "Weiter".
4. Wenn Sie mit der Installation fortfahren wollen, lesen und akzeptieren Sie alle Lizenzvereinbarungen und klicken Sie auf "Weiter".
Verhindern Sicherheits- und Berechtigungseinstellungen die Installation des TIA Portals, wird der Dialog für die Sicherheitseinstellungen geöffnet.
5. Um die Installation fortzusetzen, akzeptieren Sie die Änderungen an den Sicherheits- und Berechtigungseinstellungen.

4.1 SiVArc installieren

6. Prüfen Sie im Überblick die ausgewählten Installationseinstellungen.
7. Ändern Sie die Einstellungen bei Bedarf und klicken Sie anschließend auf "Installieren".
Die Installation wird gestartet.
Der Abschluss der Installation wird angezeigt.
8. Starten Sie den PC bei Bedarf neu oder beenden Sie das Setup.

Ergebnis

Das Add-on-Paket "SiVArc" ist auf Ihrem PC installiert.

Arbeiten mit bestehenden SiVArc-Projekten

Bestehende SiVArc-Projekte können Sie im TIA-Portal auch ohne SiVArc-Installation mit einer Basisinstallation öffnen.

Wenn Sie das Projekt anschließend mit SiVArc öffnen, sind alle SiVArc -Funktionen wieder aktiv.

Um ein SiVArc-Projekt hochzurüsten, benötigen Sie eine SiVArc-Installation.

Eine Basisinstallation besteht aus folgenden Software-Paketen:

- STEP7 Professional
- SIMATIC WinCC Professional
oder
- SIMATIC WinCC Advanced
- SIMATIC WinCC Unified

Um die Referenz auf SiVArc in Ihrem Projekt zu entfernen, löschen Sie alle SiVArc-Projektierungen aus Ihrem Projekt. Wenn Sie das Projekt mit einer Basisinstallation öffnen, wird kein Hinweis auf die fehlende SiVArc--Installation mehr ausgegeben.

Elemente und Grundeinstellungen

5.1 SiVArc-Einstellungen

Generierungs-Einstellungen

Sie können die folgenden Funktionen unter Einstellungen > SiVArc > Generierung > "Einstellungen Warnungen" aktivieren:

- Warnungen verbergen, wenn Bildobjekt bereits existiert - anwendbar, wenn ein Bildobjekt bereits im generierten Bild verfügbar ist
- Warnung verbergen, wenn Variable in multilingualem Kontext nicht definiert ist - anwendbar bei Variablen mit sprachspezifischem Kontext
- Warnung verbergen, wenn Bildgröße geändert wird - anwendbar bei geräteübergreifender Generierung mit nicht übereinstimmenden Bildgrößen
- Warnungen verbergen, wenn Texteinträge nicht aufgelöst werden können - anwendbar, wenn unter "S7 Plug-in Editor > Textdefinitionen" kein Text definiert ist

Bibliothekstyp-Einstellungen

Standardversion des Bibliothekentyps für SiVArc-Generierung verwenden - erlaubt Ihnen, die Standardversion des Bibliothekentyps in den Bildbausteinen für die SiVArc-Generierung zu verwenden.

Einstellungen für Erfassungszyklus und Erfassungsmodus für WinCC Unified

Ein Projekt aus einem Unified HMI-Gerät und einem Advanced-Gerät wird mit Variablen Mitgliedeinstellungen auf Bausteinebene oder Einstellungen auf Projektebene projiziert und wenn das Advanced-Gerät den Erfassungszyklus unterstützt:

- wird standardmäßig T1s für HMI-Variablen in Unified-Geräten projiziert.

Ein Projekt aus einem Unified HMI-Gerät und einem Advanced-Gerät wird mit Variablen Mitgliedeinstellungen auf Bausteinebene oder Einstellungen auf Projektebene projiziert und wenn das Unified-Gerät den Erfassungszyklus unterstützt:

- wird standardmäßig 1s für HMI-Variablen in Advanced-Geräten projiziert.

Der Erfassungsmodus kann auf Projekt- oder auf Bausteinebene projiziert werden. Auf Projektebene sind die Typen:

- Zyklisch im Betrieb
- Auf Anforderung

Auf **PLC-Bausteinebene** können Sie für Variablen Mitgliedeinstellungen die Option "**Gemeinsame Projektierung verwenden**" wählen, um über alle Eingangsvariablen hinweg die gleiche Einstellung für Erfassungszyklus und Erfassungsmodus zu wählen.

Weitere Informationen zur Gerätefunktion für WinCC Unified finden Sie im *Benutzerhandbuch zum TIA Portal*.

Von SiVArc unterstützte Objekte, die für WinCC Unified-Geräte konfiguriert wurden, werden als Unified-Objekte bezeichnet. Beispiel: In der Kopiervorlage einer Bildregel wird eine Unified-Schaltfläche verwendet. Alle Änderungen an SiVArc-Objekten gelten auch für Unified-Objekte. Für Unified-Geräte können Sie Plug-in-Eigenschaften für Bilder konfigurieren. Von SiVArc unterstützte Ausdrücke wie `HmiDevice.Name`, `HmiDevice.Type`, `HmiApplication.Name`, `HmiApplication.Type`, `HmiDevice.Resolution.Width`, `HmiDevice.Resolution.Height` können in den Plug-in-Eigenschaften konfiguriert werden.

5.2 User Management Control (UMAC) in SiVArc

Rolle von UMAC in SiVArc

Mit User Management Control (UMAC) wird nur berechtigten Benutzern Zugang zu SiVArc-bezogenen Funktionen erlaubt. Für die Nutzung von User Management Control in SiVArc benötigen Sie Administratorrechte für SiVArc. Diese Rechte können von einem Benutzer mit Administratorrolle Engineering Standard (ES) zugewiesen werden. Weitere Informationen über die Verwendung von UMAC (User Management Control) finden Sie in der Online-Hilfe des TIA Portals.

Hinweis

- Ein Benutzer muss die Rechte für Engineering Standard besitzen, damit er Zugriff auf die Engineering-Eigenschaften und -Funktionen hat. Ist dieser Zugriff nicht möglich, kann der Benutzer das Projekt nicht öffnen und eine Fehlermeldung wird angezeigt.
- Wenn ein Benutzer Administratorrechte für Engineering, aber nicht für SiVArc besitzt, und er versucht, das Projekt zu schließen und neu zu öffnen, wird eine Fehlermeldung wegen eines unbefugten Zugriffs angezeigt.

ACHTUNG

Know-how-Schutz

Bei der Konvertierung von Projekten aus TIA Portal V15 und älteren Versionen zu V15.1 werden die für die Funktion "Know-how-Schutz" eingestellten Anmeldedaten automatisch gelöscht. Stellen Sie sicher, dass das Projekt durch UMAC geschützt ist.

5.3 SiVArc-Editoren

5.3.1 Editor "Bildregeln"

Beschreibung

Im Editor "Bildregeln" definieren Sie die Bildregeln, nach denen SiVArc HMI-Objekte in Bildern für verschiedene Geräte generiert. Eine Regel beinhaltet folgende Elemente:

- Name
Eindeutiger Name der Bildregel
- Programmbaustein
Funktionsbaustein oder Funktionsaufruf, der an beliebiger Stelle im Anwenderprogramm aufgerufen wird.
- Bildobjekt
Kopiervorlage oder Typ eines HMI-Objekts, das generiert wird. Die Kopiervorlage oder der Typ muss in einer Bibliothek abgelegt sein.
- Bild
Generiervorlage des Bilds, auf dem das HMI-Objekt generiert wird. Die Generiervorlage muss in einer Bibliothek abgelegt sein.
- Positionierungsfeld
Positionierungsfeld, das im Positionierungsschema des Bilds enthalten ist. Über das Positionierungsfeld legen Sie die Positionierung des zu generierenden HMI-Objekts fest.
- Bedingung (optional)
SiVArc-Ausdruck, der beim Bearbeiten dieser Bildregel ausgewertet wird. Wenn keine Bedingung formuliert ist, wird die Bildregel immer ausgeführt. Die Bedingung gilt für alle Elemente einer Regelgruppe. Über Operanden können Sie für einzelne Regeln der Regelgruppe die Bedingung verfeinern.
WinCC Unified-Bilder unterstützen SiVArc-Ausdrücke.
- Kommentar (optional)
Individueller Kommentar zur Bildregel

Folgende Spalten blenden Sie bei Bedarf über die Symbole in Symbolleiste ein:

- PLC
Die Bildregel wird für die ausgewählten Steuerungen ausgeführt. Wenn Sie keine Steuerung auswählen, gilt die Regel für alle Steuerungen Ihres Projekts.
- HMI-Gerät
Die Bildregel wird für die ausgewählten HMI-Geräte ausgeführt. Wenn Sie kein HMI-Gerät auswählen, gilt die Regel für alle HMI-Geräte Ihres Projekts.
- HMI-Gerätetyp
Wenn mehrere HMI-Geräte gleichen Typs in Ihrem Projekt existieren, können Sie auch Typen von HMI-Geräten auswählen. Ob eine Regel auf einem HMI-Gerät oder für eine Steuerung angewandt werden kann, wird bei der Generierung geprüft und angezeigt.

Wenn Sie zu einem Programmbaustein ein Bild ohne Bildobjekt generieren wollen, dann lassen Sie das Feld "Bildobjekt" leer.

Jeder Editor, der Ausdrücke unterstützt, speichert die Werte standardmäßig in der entsprechenden Eigenschaft, wenn Sie den Editor wechseln, ohne den Ausdruck manuell zu speichern.

Sie können in jedem beliebigen Editor eine neue Regel/Regelgruppe einem beliebigen Index hinzufügen. Dazu führen Sie einen Rechtsklick auf die Regel aus > und fügen anschließend eine neue Regel/Regelgruppe ein.

1. Beim Einfügen einer neuen Regel/Regelgruppe an einem Indexpunkt wird die neue Regel/Regelgruppe unmittelbar unterhalb des ausgewählten Indexes eingefügt.
2. Wenn das Einfügen einer neuen Regel/Regelgruppe durch Auswählen einer Gruppe erfolgt, wird die neue Regel/Regelgruppe innerhalb der ausgewählten Gruppe am ersten Index eingefügt.

Beispiel

Einen Programmbaustein können Sie zur Steuerung eines Ventils oder eines Motors einsetzen. Abhängig vom Einsatz des Programmbausteins soll eine Schaltfläche mit der Beschriftung "Ventil öffnen" oder "Motor starten" generiert werden.

Sie benötigen jeweils eine Bildregel für das Ventilsymbol und für das Motorsymbol.

	Name	Program block	Screen object	Master copy of a screen	Layout field	Condition	Comment
1	ScnRule_Btn_Valve	Controller...	Button_1	StartScreen	Status	Block.Parameters("Tagname").Value = "Valve"	
2	ScnRule_Btn_Motor	Controller...	Button_1	StartScreen	Status	Block.Parameters("Tagname").Value = "Motor"	
3	<create new rule>						

Wenn der Programmbaustein von SiVArC beim Generieren der HMI-Objekte bearbeitet wird, wertet SiVArC die Bedingung jeder Bildregel aus. In diesem Beispiel wird der Einsatz des Programmbausteins über einen Eingang definiert, zum Beispiel `Block.Parameters("Variablenname").Value = "Valve"`. Somit trifft die Bedingung der ersten Bildregel zu, die dann die Schaltfläche mit der Beschriftung "Ventil öffnen" generiert. Die maximale Länge für "Bedingung" beträgt 1000 Zeichen.

Siehe auch

- SiVArC-Variablen (Seite 110)
- Anzeige in den SiVArC-Editoren bearbeiten (Seite 277)
- SiVArC-Regeln exportieren und importieren (Seite 201)
- SiVArC-Regeln bearbeiten und verwalten (Seite 199)

5.3.2 Editor "Variablenregeln"

Beschreibung

Im Editor "Variablenregeln" definieren Sie Variablenregeln, nach denen die von SiVArc generierten externen Variablen strukturiert in der Projektnavigation abgelegt werden.

Doppelklicken Sie in der Projektnavigation auf "Gemeinsame Daten > SiVArc > Variablenregeln", um den Editor "Variablenregeln" zu öffnen.

Eine Variablenregel enthält die folgenden Elemente:

- Name
Eindeutiger Name der Variablenregel
- Index
Gibt an, in welcher Reihenfolge die Regeln ausgeführt werden. Den Index können Sie mittels Drag&Drop in den Tabellenzeilen ändern.
- Variablengruppe
Name der Variablengruppe, in der die externe Variable generiert wird
- Variablentabelle
Name der Variablentabelle, in der die externe Variable generiert wird
- Bedingung (optional)
SiVArc-Ausdruck, der bei der Verarbeitung dieser Variablenregel ausgewertet wird
- Kommentar (optional)
Individueller Kommentar zur Variablenregel
- Die Werkzeugleiste enthält gerätespezifische Spaltenanzeigen für die verfügbaren HMI-Geräte und Typen im TIA Portal-Projekt. Die gleichen Optionen stehen im Bereich "SiVArc-Eigenschaften" zur Verfügung.

	Name	Index	Tag group hierarchy	Tag table	Condition
1	Tag rule	0	HmiTag.DB.FolderPath	HmiTag.DB.SymbolicName	
2	<create new rule>				

Siehe auch

- SiVArc-Objekteigenschaften (Seite 259)
- Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)
- SiVArc-Regeln exportieren und importieren (Seite 201)
- SiVArc-Regeln bearbeiten und verwalten (Seite 199)
- Editor "Kopierregeln" (Seite 34)
- Ergebnis prüfen (Seite 218)

5.3.3 Editor "Textlistenregeln"

Beschreibung

Im Editor "Textlistenregeln" definieren Sie SiVArc-Regeln, nach denen Textlisten für verschiedene Geräte generiert werden. Eine Textlistenregel beinhaltet folgende Elemente:

- Name
Eindeutiger Name der Textlistenregel
- Programmbaustein
Funktionsbaustein oder Funktionsaufruf, der an beliebiger Stelle im Anwenderprogramm aufgerufen wird.
- Textliste
Kopiervorlagen der Textlisten werden während der Generierung im Editor "Text- und Grafiklisten" gespeichert.
- Bedingung (optional)
SiVArc-Ausdruck, der bei der Verarbeitung dieser Textlistenregel ausgewertet wird. Wird keine Bedingung angegeben, wird die Textlistenregel immer ausgeführt.
- Kommentar (optional)
Individueller Kommentar zur Textlistenregel
- Die Werkzeugleiste enthält gerätespezifische Spaltenanzeigen für die verfügbaren HMI-Geräte und PLCs im TIA Portal-Projekt. Die gleichen Optionen stehen im Bereich "SiVArc-Eigenschaften" zur Verfügung.



Siehe auch

- Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)
- SiVArc-Regeln exportieren und importieren (Seite 201)
- SiVArc-Regeln bearbeiten und verwalten (Seite 199)

5.3.4 Editor "Melderegeln"

Einleitung

Im Editor "Melderegeln" definieren Sie die SiVArc-Melderegeln, nach denen die Meldungen für verschiedene Bediengeräte definiert werden. Der Editor "Melderegeln" setzt sich wie folgt zusammen:

- Name - eindeutiger Name der Meldung.
- Regel-Trigger - Funktionsbaustein oder Funktionsaufruf, der an beliebiger Position im Anwenderprogramm aufgerufen wird.
- Kopiervorlagen von Meldungen/Klassen/Gruppen - Kopiervorlagen von Meldungen, Klassen oder Gruppen, die durchsucht und abgerufen werden können.
- Bedingung (optional) - SiVArc-Ausdruck, der bei der Bearbeitung einer Melderegeln ausgewertet wird. Wird keine Bedingung angegeben, wird die Melderegeln immer ausgeführt.
- Kommentar (optional) - spezifischer Kommentar für Melderegeln.

	Name	Program block	Master copy of Alarms/Classes/Groups	Condition
1	(A... (All objects)	(All objects)	(All objects)	(All objects)
2	Alarm rule	Block_2	Alarm_group_1	NetworkTitle = "Motor"
3	Alarm rule_1	Block_2	Alarm_group_1	NetworkTitle = "Valve"
4	<create new rule>			

Der Editor "Melderegeln" verfügt über folgende Funktionen:

- IntelliSense-Unterstützung
- Automatische Vervollständigung (für die automatische Vervollständigung von Werten die Ecken der Zelle ziehen)
- Optionen zum Ausschneiden und Kopieren/Einfügen
- Erstellen einer einzelnen Regel mit identischen Einträgen, wenn sich nur Objekte unterscheiden. Sie können die Objekte auch unter "Kopiervorlage von Meldungen/Klassen/Gruppen" auswählen. Bei Mehrfachauswahl von Meldeobjekten zeigt der Editor den Eintrag als "Mehrfachauswahl" an, und bei Einfachselektion wird der Objektname angezeigt.

Voraussetzungen für die Generierung von Meldungen

- Mit dem PLC generierte Programmbausteine (FBs und FCs) werden projektiert und mit dem Bediengerät verbunden.
- Meldungen werden im Editor "Melderegeln" in WinCC erstellt und projektiert.
- Kopiervorlagen (Meldungen oder Bausteintypen) werden in der globalen Bibliothek erstellt und projektiert.

Melderegeln können Sie mithilfe des Editors "Melderegeln" in der globalen Bibliothek projektieren und speichern. Während der Projektierung der Melderegeln werden die PLC-Bausteine eines S7-Geräts den in Bibliotheken gespeicherten Kopiervorlagen zugeordnet und es

werden Bedingungen definiert. Bei der Generierung mit SiVArc gemäß den benutzerdefinierten Bedingungen verarbeitet das System die Regeln entsprechend und generiert die Meldeobjekte. Weitere Details zur Generierung von Melderegeln finden Sie im Abschnitt "Generiermatrix (Seite 36)".

Hinweis

- Die Meldeobjekte werden erst verarbeitet, wenn die Regel die benutzerdefinierte Bedingung erfüllt hat.
 - Bei der Auswahl der Programmbausteine können Sie zwischen PLC-Bausteinen oder PLC-Typen wählen.
-

Allgemeine Hinweise

- Sie können die systemgenerierten Meldeklassen nicht löschen, sondern nur bearbeiten.
- Die Eigenschaften von SiVArc sind denen des Engineering System vollständig nachgebildet. Die Optionen im Ausdruck des statischen Werts und Ausdruck der Variablen ähneln den Optionen des Engineering System.
- Häufig vorkommende Meldeklassen können Sie auch mit dem Editor "Plug-Ins" projektieren.
- Wenn bei der SiVArc-Generierung der resultierende Objektname dem Namen der standardmäßigen Meldeklasse ähnlich ist, zeigt SiVArc eine Fehlermeldung an.
- Gruppen von Meldungen, die ähnliche Klassen darstellen, werden unter "Klassengruppe" angezeigt. Mithilfe der SiVArc-Eigenschaften können Sie für Meldegruppen und -untergruppen das Quittierverhalten sowie den Status und die Meldungsunterdrückung projektieren.
- Alle neu hinzugefügten Meldeklassen werden automatisch im Bereich "Klassengruppe" angezeigt.
- Ein bestehendes Meldeobjekt kann in SiVArc bearbeitet werden.
- Wenn identische Melderegeln erstellt werden, zeigt das System folgenden Fehler an: "Melderegeln für <Objektname> ist mit <Objektname> identisch. Sie wird daher nicht bei der Generierung berücksichtigt und kann gelöscht werden."
- Bei Eigenschaften vom Typ "Aufzählung" werden die Funktionen Ausschneiden, Kopieren und Einfügen nicht unterstützt.

Siehe auch

Melderegeln erstellen (Seite 190)

Editor "Generiermatrix" (Seite 36)

5.3.5 Editor "SiVArc-Ausdrücke"

Beschreibung

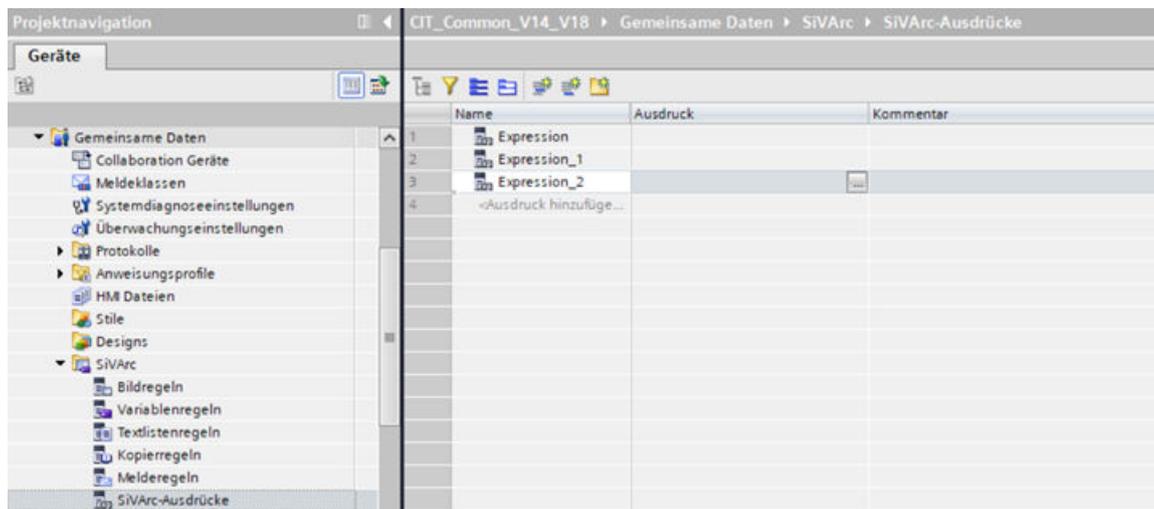
Mit dem Editor für SiVArc-Ausdrücke können Sie Ausdrücke für die Verwendung in der Projektbibliothek und in der globalen Bibliothek hinzufügen, festlegen und gruppieren. Die Ausdrücke können global durch Aufruf des jeweiligen Namens verwendet werden.

SiVArc-Ausdrücke bestehen aus:

- Name:
Eindeutiger Name des Ausdrucks
- Ausdruck:
Geben Sie einen von SiVArc unterstützten Ausdruck ein. Der Ausdruckseditor unterstützt `StringPos(Parameter1,Parameter2)`.
Fall 1: Wenn Sie einen Ausdruck mit der Funktion `StringPos` konfigurieren: `StringPos (Parameter1, Parameter2)`; System gibt die Position des ersten Zeichens von `Parameter2` (Teilzeichenkette) in `Parameter1` (Hauptzeichenkette) zurück.
Beispiel: `StringPos("Sivarc_Projekt","Projekt")=8` da ("Projekt" ist eine Teilzeichenkette von "Sivarc_Projekt" – Zurückgegeben wird 8, die Position von 'P' in der ursprünglichen Zeichenkette)
Fall 2: Wenn Sie eine Ausdruckszeichenkette mit "StringPos" konfigurieren, in der die Parameter zum Extrahieren der Zeichen in der Groß-/Kleinschreibung nicht übereinstimmen oder die Zeichenkette nicht verfügbar ist; System gibt 0 zurück.
Beispiel: `StringPos("Sivarc_Projekt","Por")=0` da ("Por" keine Teilzeichenkette von "Sivarc_Projekt" ist – Zurückgegeben wird 0)
Fall 3: Wenn Sie eine Ausdruckszeichenkette mit "StringPos" konfigurieren, bei der beide Zeichenketten oder eine der Zeichenketten in der Funktion "StringPos" leer ist, gibt die Funktion -1 zurück.
Beispiel: `StringPos("", "")=-1` da (beide Parameter sind leer)
- Kommentar:
Kommentar für den SiVArc-Ausdruck

Die Symbolleiste enthält Funktionen zum Gruppieren, Filtern, Einklappen aller erweiterten Zeilen, Erweitern aller hierarchischen Zeilen, Einfügen oberhalb und Einfügen unterhalb. Im Ausdruckseditor können Sie folgende Aufgaben ausführen:

1. Klicken Sie auf "Create new expression", um einen neuen Ausdruck hinzuzufügen.
2. Sie können die Ausdrücke markieren und durch Klicken auf die Gruppierungsoption in der Symbolleiste gruppieren.
3. Sie können ferner eine Gruppe durch Rechtsklick auf die Zeile > "Add a new expression group" erstellen.
4. Schließlich können Sie eine Gruppe innerhalb einer Gruppe mit Rechtsklick auf die Gruppe > "Insert a new expression group" erstellen.
5. Neue Ausdrücke können Sie mit der Funktion in der Symbolleiste oberhalb bzw. unterhalb einer ausgewählten Zeile einfügen.



Sie können die erstellten Ausdrücke als Objekte in die Projektbibliothek/globale Bibliothek ziehen. In diesem Fall steht das Objekt in der Objektauswahl des Ausdruckseditors zur Verfügung. Ein Ausdruck kann in mehreren Instanzen verwendet werden. Dies sind so genannte globale Ausdrücke. Globale Ausdrücke können Sie in der Bedingungsspalte einer Regeltabelle für Bilder, Meldungen, Variablen, Kopierregeln verwenden.

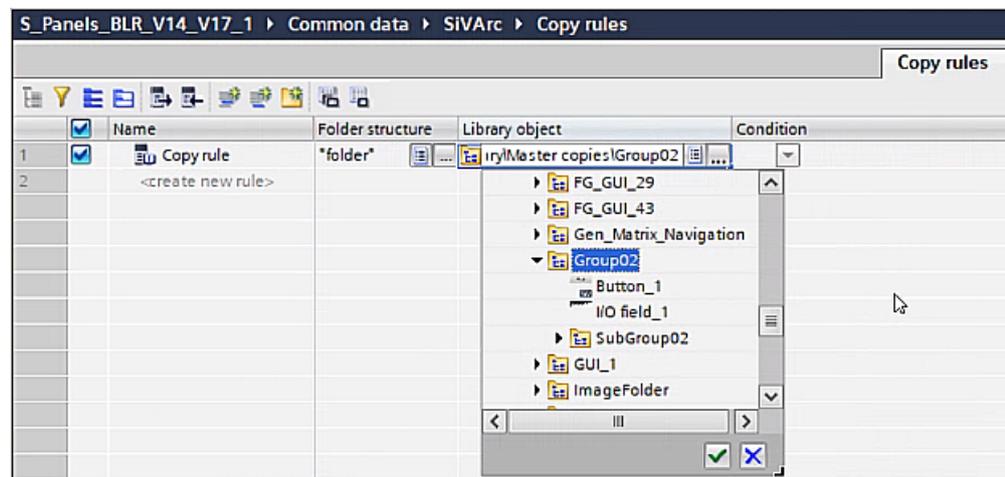
5.3.6 Editor "Kopierregeln"

Beschreibung

Eine Regel beinhaltet folgende Elemente:

- Name
Eindeutiger Name der Kopierregel
- Ordnerstruktur
Die benutzerdefinierte Ordnerstruktur kann entweder manuell oder mit Hilfe von Ausdrücken erstellt werden. Dies gilt für Bibliotheksobjekte, bei denen das Erstellen von Ordnern unterstützt wird.
Wenn der Wert in der Spalte "Ordnerstruktur" leer ist, dann generiert SiVArC alle Objekte, die vom Stammknoten bis zu einer verschachtelten Tiefe der Ebene '4' unter dem PNV-definierten Standardordner konfiguriert wurden.
Wenn Sie die Objekte mit der in der Projektbibliothek konfigurierten Ordnerstruktur replizieren wollen, dann können Sie die Spalte Ordnerstruktur mithilfe eines Ausdrucks konfigurieren.
Sie können auch jede beliebige gewünschte Ordnerstruktur für die SiVArC-Generierung in der Spalte "Ordnerstruktur" wählen.
- Projektbibliothek
Die Ordnerstruktur besteht aus einem Stammordner der Ebene "0" und den nachrangigen Ordnern innerhalb des Stammordners in Form von verschachtelten Ordnern, Ebene "1", Ebene "2" usw. Bezüglich der Verschachtelungstiefe berücksichtigt SiVArC bei der Generierung nur Ordner bis zur Ebene "4".

- Bibliotheksobjekt
Kopiervorlage oder Typ eines HMI-Objekts, das generiert wird, oder Bibliotheksordner, der **Bibliotheksobjekte > Positionierungsfelder** enthält.
Wenn sich das Positionierungsfeld im Bearbeitungsmodus befindet, wird das zuvor freigegebene Positionierungsfeld ausgewählt. Die Kopiervorlage bzw. der Bibliothekstyp muss in der Projektbibliothek enthalten sein. Ein Unified-Bild steht als Objekt in der Objektauswahl zur Verfügung und bei der SiVArc-Generierung wird das Bild nur auf dem Unified-Gerät generiert. Grafiken werden als Bibliotheksobjekt unterstützt.
- Bedingung (optional) – SiVArc-Ausdruck, der bei der Verarbeitung der Kopierregel ausgewertet wird. Wird keine Bedingung angegeben, wird die Kopierregel immer ausgeführt.
- Kommentar (optional)
Individueller Kommentar zur Kopierregel



Sie können folgende Spalten bei Bedarf über die Symbole in der Symbolleiste einblenden:

- HMI-Gerät
Die Bildregel wird für die ausgewählten HMI-Geräte ausgeführt. Wenn Sie kein HMI-Gerät auswählen, gilt die Regel für alle HMI-Geräte Ihres Projekts. Wenn die Kopierregel auf Systemtextlisten beruht, wird bei Unified-Geräten während der SiVArc-Generierung ein Fehler erzeugt.
- HMI-Gerätetyp
Wenn mehrere HMI-Geräte vom gleichen Typ in Ihrem Projekt existieren, können Sie auch Typen von HMI-Geräten auswählen. Ob eine Regel auf ein HMI-Gerät oder eine Steuerung angewandt werden kann, wird bei der Generierung geprüft und angezeigt.

Siehe auch

Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)

SiVArc-Regeln exportieren und importieren (Seite 201)

SiVArc-Regeln bearbeiten und verwalten (Seite 199)

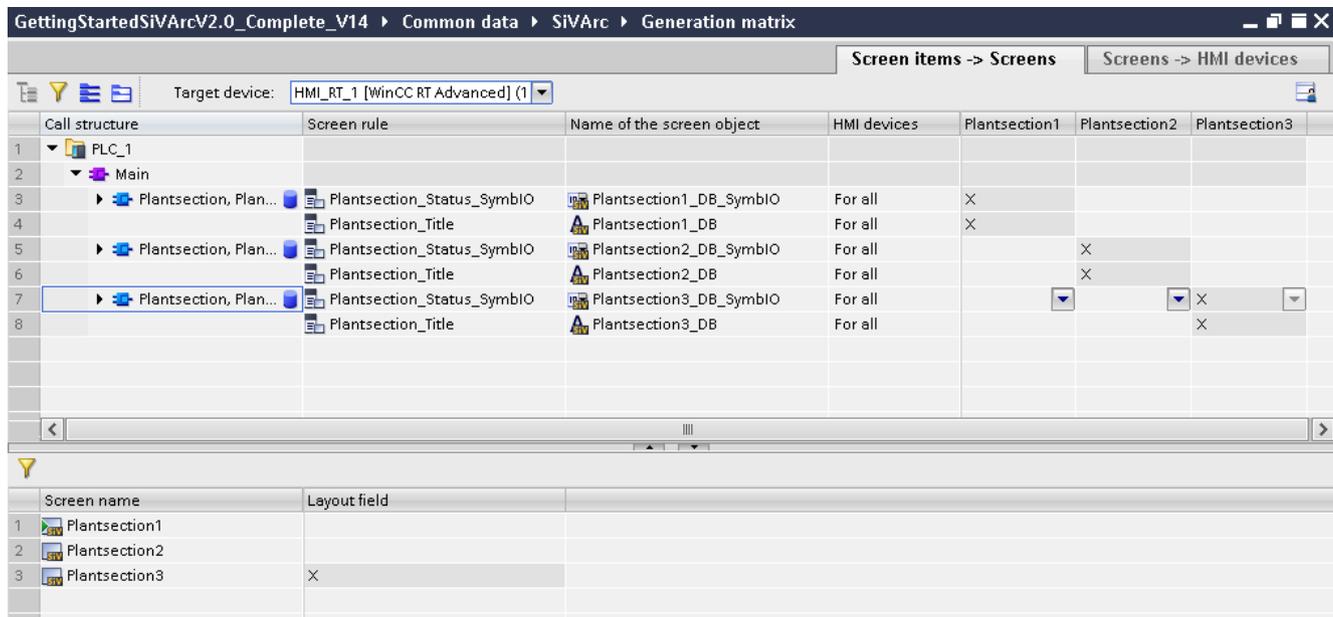
5.3.7 Editor "Generiermatrix"

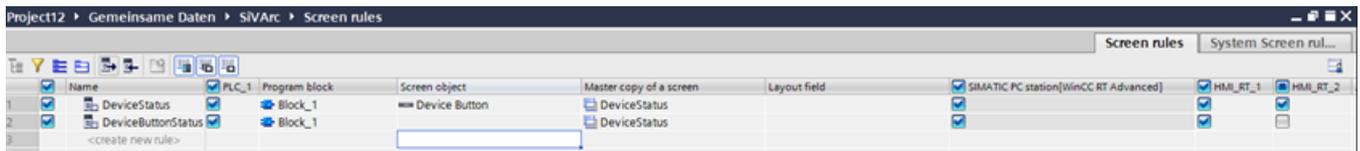
Registerkarte "Bildobjekte -> Bilder"

In der Symbolleiste des Editors wählen Sie das Bediengerät (Legacy/WinCC Unified) aus, für das die Matrix unter "Zielgerät" angezeigt werden soll. SiV Arc zeigt außerdem den Gerätetyp für alle Geräte an.

Weisen Sie in dieser Registerkarte ein generiertes Bildobjekt einem anderen Bild zu. Die Registerkarte enthält die folgenden Spalten:

- Aufrufstruktur
Zeigt für jede Zeile die Bausteininstanzen an, die im Benutzerprogramm aufgerufen und für das Generieren von Bildobjekten verwendet werden.
- Bildregel
Zeigt die Bildregeln an, die für jede Bausteininstanz ausgeführt werden.
- Name des Bildobjekts
Zeigt das generierte Bildobjekt an.
- HMI-Geräte
Listet für jedes Bildobjekt die HMI-Geräte auf, für die das Bildobjekt generiert wurde.
- Bildspalten
Für jedes Bild wird eine separate Spalte angezeigt. Die Spalten sind alphabetisch sortiert.
 - "X": Das Bildobjekt ist nicht in einem Positionierungsfeld positioniert.
 - "<Name des Positionierungsfelds>": Das Bildobjekt ist in dem angegebenen Positionierungsfeld enthalten.



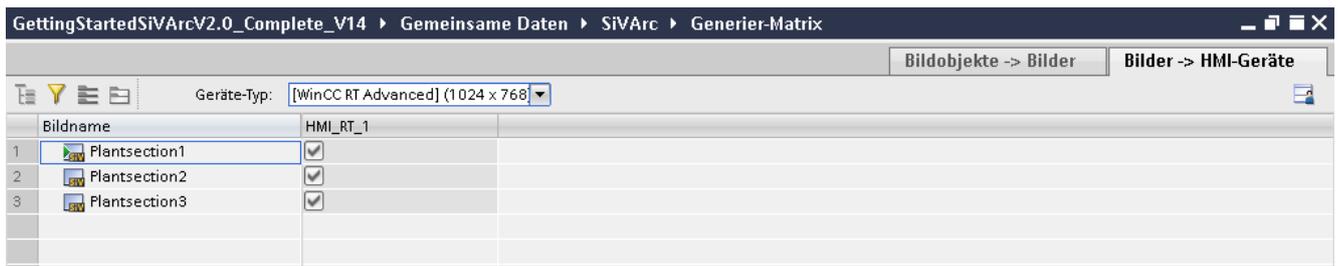


Registerkarte "Bilder -> HMI-Geräte"

In der Symbolleiste des Editors wählen Sie den HMI-Gerätetyp, für das die Matrix unter "Geräte-Typ" angezeigt werden soll. Daraufhin zeigt der Editor die Bilder aller HMI-Geräte dieses Typs an.

Weisen Sie in dieser Registerkarte ein generiertes Bild einem anderen HMI-Gerät zu. Die Registerkarte enthält die folgenden Spalten:

- Bild
Zeigt die generierten Bilder an.
- HMI-Geräte
Zeigt die HMI-Geräte an (einschließlich WinCC Unified). Für jedes HMI-Gerät wird eine separate Spalte angezeigt. Die Spalten sind alphabetisch sortiert.

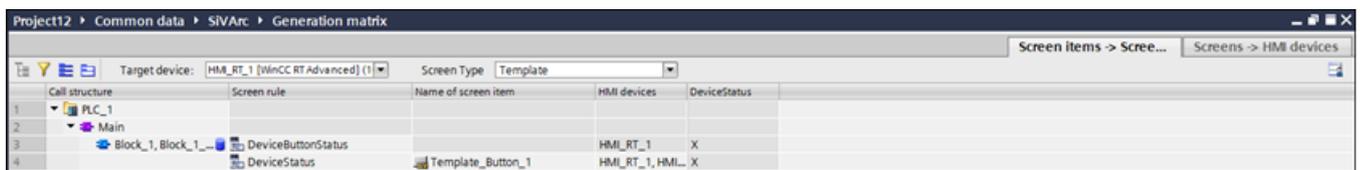


Vorlagen- und Pop-up-Bilder

Die Generiermatrix umfasst zwei Funktionen:

Sie können Vorlagen-/Pop-up-Bilder oder Bildobjekte verschiedenen HMI-Zielgeräten zuweisen. Sie können Vorlagen-/Pop-up-Bildobjekte verschiedenen HMI-Zielgeräten zuweisen, die mit Bildern vergleichbar sind. Die Symbolleiste des Editors zeigt die Dropdown-Option "Bildtyp" an, die die Elemente Bild, Vorlage und Pop-up enthält, wie im nachfolgenden Screenshot dargestellt:

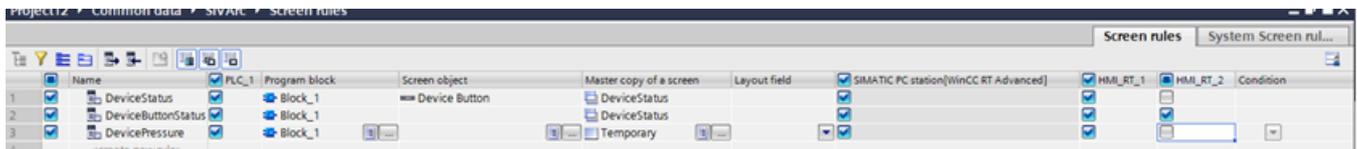
- Matrix 1 - Bildobjekte -> Bilder - Sie können Bildobjekte auf Bildern verschiedener HMI-Geräte zuweisen
- Matrix 2 - Bilder -> HMI-Geräte - Sie können Bilder verschiedenen HMI-Geräten zuweisen



Szenario: Navigieren von Objekten zu verschiedenen HMI-Geräten ohne Zugriff auf Bilder

Ein Inbetriebnahmetechniker verschiebt Bildobjekte auf verschiedene HMI-Zielgeräte ausschließlich mit Hilfe der SiVArC-Visualisierung, da er keine Berechtigung für den Zugriff auf Bildregeln besitzt. Betrachten wir die nachfolgende Bildregelkonfiguration:

1. Konfigurieren Sie die Bildregeln, wie nachfolgend erwähnt:
 - DeviceButtonStatus, der das Vorlagenbild enthält und nur in HMI-Gerät 1 generiert wird
 - DeviceStatus, der darin das Bildobjekt "Template_button_1" enthält und in HMI-Gerät 1 generiert wird

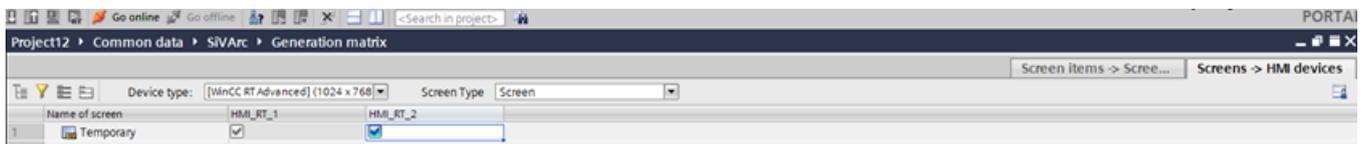


2. Bei der SiVArC-Generierung wird das Bild "DeviceButtonStatus" mit dem Bildobjekt "Device Button" auf HMI_RT_1 und "Device Status" auf HMI_RT_2 generiert.
3. Erstellen Sie ein Zielbild mit dem darin positionierten Bildobjekt "Device Button". Konfigurieren Sie Ereignisse mit "Klicken" als "Bild aktivieren" auf das Zielbild gerichtet.

Hinweis

Bei der Konfigurierung von Bildern können Sie Ereignisse für ein Bildobjekt direkt konfigurieren. Bei Vorlagen- und Pop-up-Bildern müssen Sie jedoch ein Zielbild verwenden, um das Bildobjekt zum Vorlagenbild zu navigieren. Ein Zielbild ist erforderlich, da Ereignisse für Vorlagen- und Pop-up-Bilder nicht direkt konfiguriert werden können.

4. Konfigurieren Sie in "Bildregeln" ein Zielbild als "Temporär" und wählen Sie das HMI-Zielgerät HMI_RT_1 aus.
5. Klicken Sie auf die SiVArC-Einstellungen > "Matrix Einstellungen" > "Navigationsschaltflächen generieren".
6. Gehen Sie zu Generiermatrix > "Bilder -> HMI-Geräte", wählen Sie "Bildtyp" als "Bild" aus und markieren Sie das Optionskästchen HMI_RT_2, wie im nachfolgenden Screenshot dargestellt:



7. Führen Sie die SiVArC-Generierung auf HMI-Gerät 2 aus. Beachten Sie Folgendes:
 - Das in Schritt 2 erstellte Bildobjekt wird in das Bild unter dem Ordner "Bildverwaltung" verschoben.
 - Das Bild mit dem Bildobjekt wird generiert und ist unter dem Ordner "Bilder" verfügbar.

Das Vorlagenbildobjekt "Device Button" wird auf das Vorlagenbild "DeviceButtonstatus" über das Zielbild verschoben, d.h. "Temporär" in "Bilder -> HMI-Geräte".

Hinweis

- Die vorstehend beschriebene Vorgehensweise trifft auch auf Pop-up-Bilder zu.
 - Bei Kopierregeln müssen Sie nach dem Ändern der Bildregel das Gerät manuell für das entsprechende geänderte Bild in der Generiermatrix einstellen (einschließlich Navigationsschaltflächen).
-

5.3.8 Expression Resolver

Beschreibung

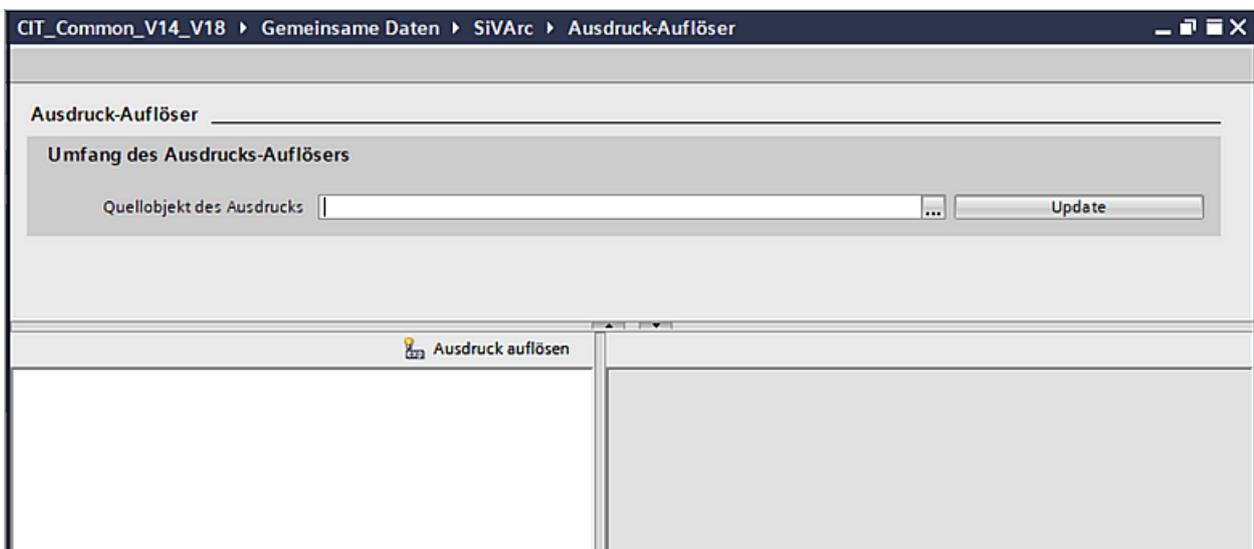
Der Expression Resolver teilt Ihnen den aufgelösten Wert des SiVArc-Ausdrucks für das vorgegebene Ausdrucks-Quellobjekt mit. Der eingegebene Ausdruck wird aufgelöst und die Ergebnisse werden umgehend ohne SiVArc -Generierung im Editor angezeigt.

Das "Ausdrucks-Quellobjekt" kann für eine Baueinstanz, ein HMI-Gerät, ein IO-Device oder ein Bibliotheksobjekt spezifisch sein.

Der Expression Resolver umfasst 3 Bereiche:

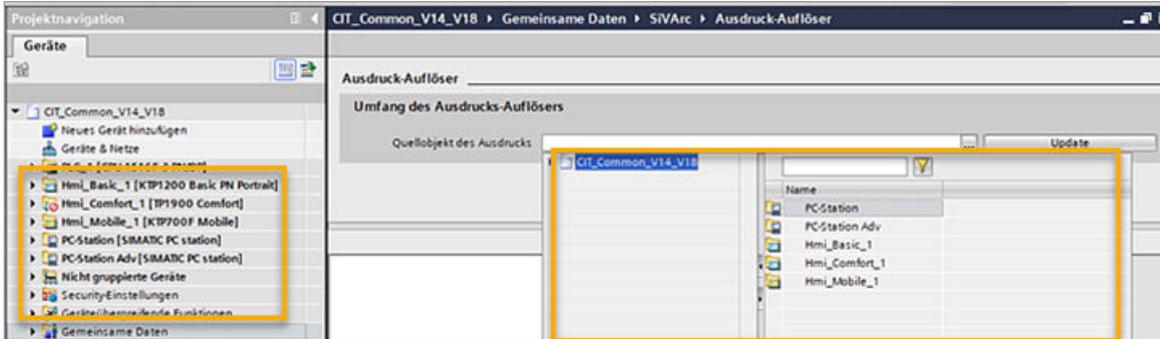
1. Konfigurationseditor für Ausdrucks-Quellobjekte
2. Konfigurationseditor für SiVArc-Ausdrücke
3. Fenster für Ausdrucksergebnisse

Die nachfolgende Abbildung zeigt den Expression Resolver:

**Quellobjektkonfiguration**

Der Expression Resolver unterstützt nur Baueinstanz und HMI-Gerät als Ausdrucks-Quellobjekt.

HMI-Gerät als Quellobjekt - Bei einem Projekt mit HMI-Geräten werden bei Auswahl des Projektknotens auf der linken Seite/in der hierarchischen Ansicht in der Objektauswahl die verfügbaren HMI-Geräte auf der rechten Seite angezeigt, wie in der nachfolgenden Abbildung dargestellt:



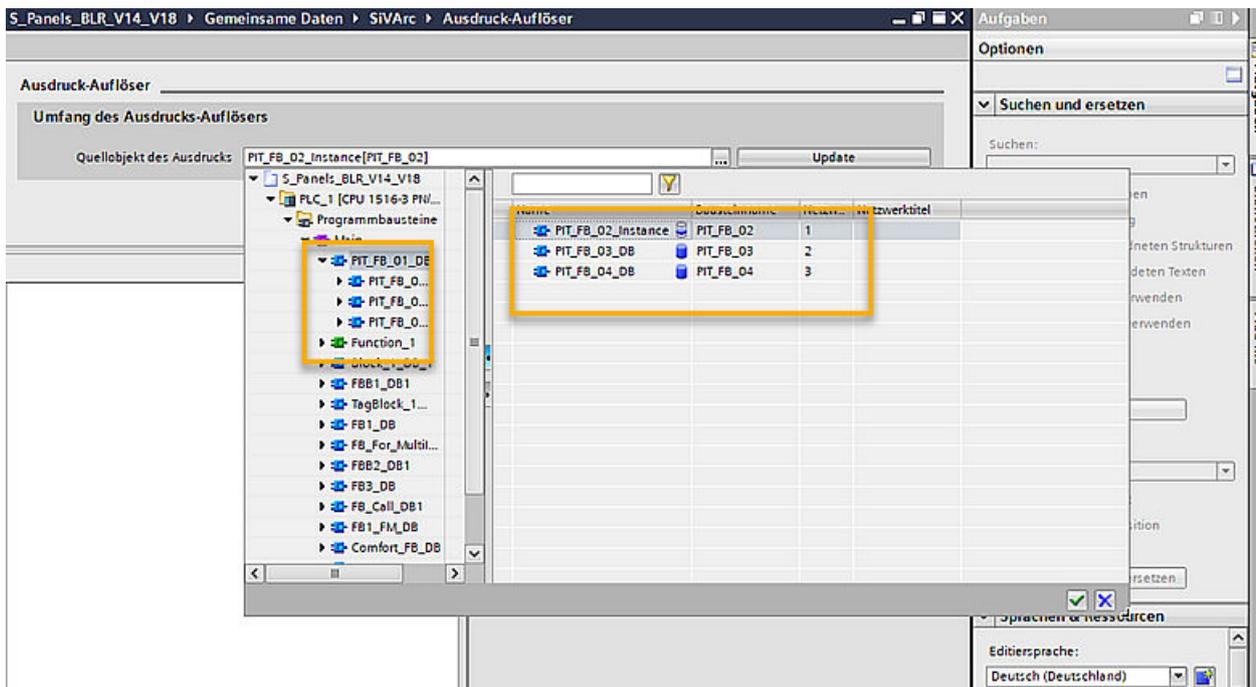
Programmbausteinanruf als Ausdrucks-Quellobjekt

In der Objektauswahl wird die Aufrufstruktur von im TIA-Portalprojekt konfigurierten PLCs angezeigt.

Wenn ein Organisationsbaustein oder ein Bausteinanruf in der Baumansicht (linke Seite) ausgewählt wird, werden auf der rechten Seite der Objektauswahl die verfügbaren Programmbausteinanrufe im ausgewählten Aufruf in Spalten wie folgt angezeigt:

- Name - zeigt bei Funktionsbausteinanrufen den Instanznamen und bei Funktionsaufrufen den Bausteinnamen an
- Bausteinname - zeigt den Namen des dem Aufruf zugeordneten Bausteins an
- Netzwerknummer - zeigt die Nummer des Netzwerks an, in dem der aktuelle Aufruf verfügbar ist
- Der Netzwerktitel für den aktuellen Aufruf wird angezeigt, wenn die Konfiguration im PLC-Programm für die aktuelle Editiersprache vorhanden ist.

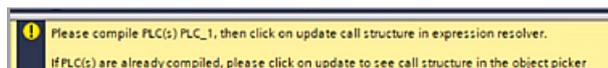
Sie können Filter auf die Spalten anwenden. In der nachfolgenden Abbildung sehen Sie die Objektauswahl mit den oben erwähnten Details:



In der Auswahl "Ausdrucks-Quellobjekt" wird die Aufrufstruktur nur für übersetzte PLCs im Projekt angezeigt.

Ist eine PLC nicht übersetzt, zeigt die Objektauswahl keine Aufrufstruktur für die in der PLC definierten Organisationsbausteine an.

Bei nicht übersetzten PLCs zeigt der Expression Resolver ein Banner mit den Informationen der PLCs an. Angenommen, Sie haben ein Projekt mit 3 PLCs: "PLC_1", "PLC_2" und "PLC_3". Wenn nur PLC_1 übersetzt ist, wird ein Banner mit folgender Meldung angezeigt:



Wenn der Expression Resolver zum ersten Mal geöffnet wird, lädt der Editor automatisch die Aufrufstruktur sämtlicher übersetzter PLCs. Der Editor zeigt das Banner nur an, wenn es nicht übersetzte PLCs gibt, und in der Objektauswahl wird basierend auf der geladenen Aufrufstruktur das gleiche angezeigt. Wenn Sie ein PLC-Programm ändern, zeigt der Expression Resolver die neuen/modifizierten Änderungen nicht an.

Wenn Sie möchten, dass im Expression Resolver die modifizierten/neuen Änderungen berücksichtigt werden, müssen Sie die PLC übersetzen und im Konfigurationseditor für Quellobjekte auf die Schaltfläche "Aktualisieren" klicken. Durch das "Aktualisieren" wird die verfügbare Aufrufstruktur übersetzter PLCs geladen und auch für die Auflösung von Ausdrücken verwendet.

Konfigurationseditor für SiVArc-Ausdrücke

Der Konfigurationseditor unterstützt Intellisense, Syntaxmarkierung, Kopieren, Einfügen, Löschen ähnlich wie bestehende lokale Editoren für die Ausdrucksauswahl.

Der Konfigurationseditor enthält eine Schaltfläche zum Auflösen in der Symbolleiste. Nach Eingabe des Ausdrucks klicken Sie auf die Schaltfläche "Ausdruck auflösen", um den aufgelösten Wert im Ergebnisfenster anzuzeigen.

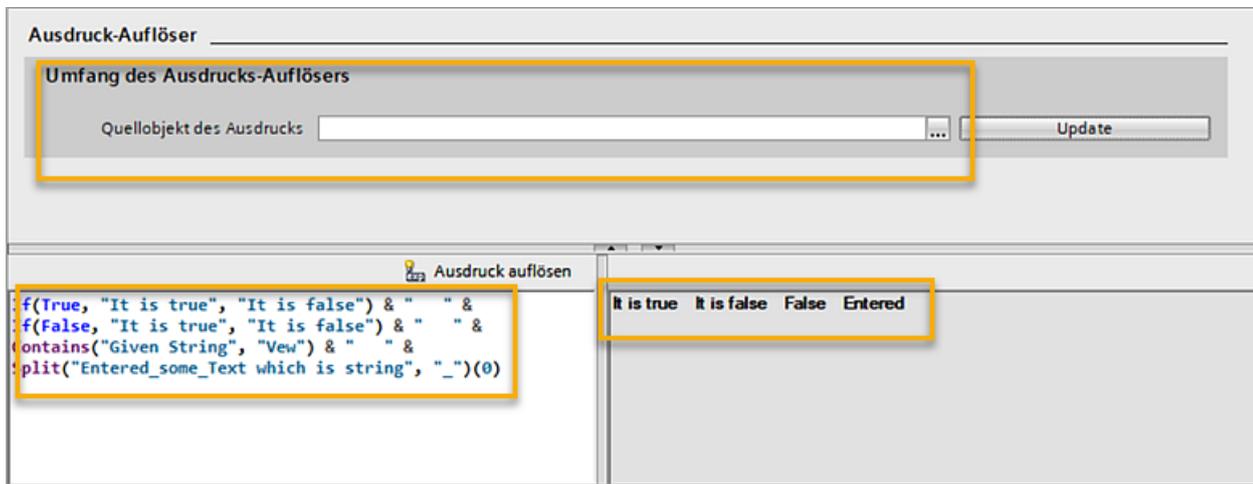
Fenster für Ausdrucksergebnisse

Zeigt den aufgelösten Wert für den eingegebenen Ausdruck im Konfigurationseditor für Ausdrücke. Dies ist abhängig von Ihrer Auswahl unter "Ausdrucks-Quellobjekt".

Betrachten Sie für die Arbeitsweise eines Resolvers folgende Szenarien:

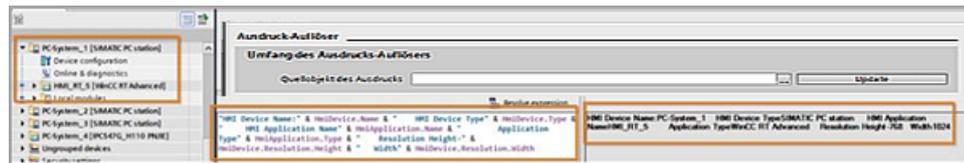
Fall 1: String-basierte Operationen ohne Ausdrucks-Quellobjekt.

Wenn unter "Ausdrucks-Quellobjekt" keine Auswahl getroffen wird, aber im SiVArc-Ausdruckseditor Ausdrücke eingegeben werden, wird der Ausdruck nach Klicken auf "Ausdruck auflösen" wie in der nachfolgenden Abbildung gezeigt aufgelöst:



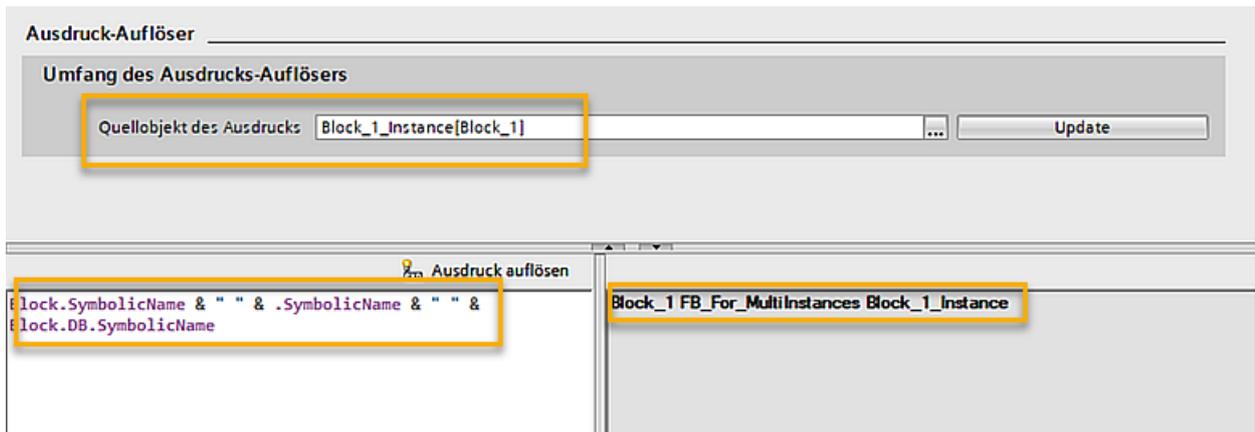
Fall 2: Auf HMI-Geräten basierende Ausdrücke

Wenn Sie unter "Ausdrucks-Quellobjekt" ein HMI-Gerät wählen und im SiVArc-Ausdruckseditor Ausdrücke eingeben, werden die folgenden Ergebnisse wie in der nachfolgenden Abbildung gezeigt angezeigt:



Fall 3: Bausteinbasierte Ausdrücke

Wenn Sie unter "Ausdrucks-Quellobjekt" einen Baustein wählen und im SiVArc-Ausdruckseditor Ausdrücke eingeben, werden die folgenden Ergebnisse wie in der nachfolgenden Abbildung gezeigt angezeigt:



Hinweis

Außer Textdefinitionen werden vom Resolver alle von SiVArC unterstützten bausteinbasierten Ausdrücke, Variablendefinitionen unterstützt.

Block.DB.HMITagPrefix- Während der SiVArC-Generierung wird der Ausdruck "Block.DB.HMITagPrefix" aufgelöst, indem ungültige Zeichen im aufgelösten Wert entfernt werden. Der Resolver umfasst keine HMI-Gerätevalidierung für den aufgelösten Wert.

Block.NetworkTitle and Block.NetworkComment - Der Ausdruck wird basierend auf der Editiersprache aufgelöst. Der Resolver ruft den konfigurierten Netzwerktitel und den Kommentar für die ausgewählte Editiersprache ab und zeigt beides in den Ergebnissen an.

Fall 4: Endlosschleife

Bei einer Endlosschleife in der PLC-Aufrufstruktur zeigt die SiVArC-Generierung eine Meldung an.

Der Resolver unterstützt die Endlosschleife in der PLC-Aufrufstruktur.

5.3.9 Generierübersicht

Beschreibung

In der Generierübersicht sind nach der ersten Generierung der Visualisierung alle generierten Bildobjekte aufgeführt. Die generierten Objekte sind auf die Registerkarten "Bilder/Bildobjekte", "Variablen" und "Textlisten" aufgeteilt.

Die Generierübersicht stellt über verschiedene Ansichten auch die Bezüge zwischen Bildregeln und generierten Objekten nach der Generierung dar. Mithilfe der Generierübersicht planen und projizieren Sie nachträgliche Änderungen für eine weitere Generierung.

ProjectV17inc5 > Common data > SiVArc > Generation Overview										
				Screens / Screen Items		Template Screens / Screen Items		Popup Screens / Screen Items		Tags
Name of screen	Name of screen item	Master copy / library type	HMI station	S7 station	Rule Trigger	Screen rule	Generated by matrix	Layout field	Call path	Generat
1	Template_1	Template_1	HM_RT_1	PLC_1	Block_2, Block_2_DB	Screen rule_4	<input type="checkbox"/>		main/block_2	11/25/2
2	Template_2	Template_2	HM_RT_1	PLC_1	Block_2, Block_2_DB	Screen rule_6	<input type="checkbox"/>		main/block_2	11/25/2
3	Template_1	Template_1	HM_1	PLC_1	Block_2, Block_2_DB	Screen rule_4	<input type="checkbox"/>		main/block_2	11/25/2
4	Template_2	Template_2	HM_1	PLC_1	Block_2, Block_2_DB	Screen rule_6	<input type="checkbox"/>		main/block_2	11/25/2

Die Inhalte der Generierübersicht setzen sich wie folgt zusammen:

Registerkarte "Bilder/Bildobjekte"	Registerkarte "Variablen"	Registerkarte "Textlisten"
Name des Bilds/Bildobjekts Eindeutiger Name des Objekts	Name Name der generierten Variablen-tabelle/ generierten Variablen	Textliste/Textlisteneintrag Name der Textliste und von deren Text- listeneinträgen
Kopiervorlage/Typ Name der Generiervorlage des Objekts	Datentyp Datentyp der generierten Variablen. Für den Datentyp "UDT" (PLC-Datentyp) wird der Name des UDT-Datentyps angezeigt.	Kopiervorlage/Typ Name der Generiervorlage für die Text- liste
Bediengerät Name des Bediengeräts, für welches das Objekt generiert wurde	Bediengerät Name des Bediengeräts, für das die ex- ternen Variablen generiert wurden	Bediengerät Name des Bediengeräts, für das die Text- liste generiert wurde
PLC-Gerät Name des PLC, für den das Objekt gene- riert wurde	PLC-Gerät Name der Steuerung, für die die Variab- len generiert wurden	PLC-Gerät Name der Steuerung, für die die Textliste generiert wurde
Programmbaustein FB oder FC, für den die Textliste gene- riert wurde	Programmbaustein DB, für den die Variable generiert wurde	Text Text, der den Textlisteneintrag enthält
Bildregel Bildregel, welche die Generierung des Objekts vorgegeben hat	PLC-Variablen Name der PLC-Variablen, für die die ex- terne Variable generiert wurde	Regelname Name der Textlistenregel, die die Gene- rierung der Textliste vorgegeben hat
Datum Zeitstempel des Datums, an dem das Objekt generiert wurde	Variablen-tabelle Name der Variablen-tabelle, in der die Va- riablen generiert wurden	Netzwerk Name des Netzwerks, das bei der Gene- rierung ausgewertet wurde
Matrixgeneriert Objekt wurde in einer nachgeschalteten Generierung über die Generiermatrix er- zeugt	Variablen-ordner Name des Ordners in der Projektnaviga- tion, in den die Variablen-tabellen und Variablen generiert wurden	Programmbaustein FB oder FC, für den die Textliste generiert wurde
Positionierungsfeld Wenn das Objekt in ein Positionierungs- feld generiert wurde, wird hier der Name des Felds angezeigt	Variablenregel Variablenregel, die die Ablagestruktur der generierten Variablen vorgegeben hat	Aufrufstruktur Aufrufpfad im Zyklus-OB "Main1", der die Generierung der Textliste vorgegeben hat
Aufrufstruktur Pfad des ausgewerteten Bausteins in der Aufrufhierarchie im Anwenderpro- gramm (OB1)	---	---

Ähnliche Optionen stehen für Popup-Bilder/Bildobjekte, Textlisten, Meldungen zur Verfügung.

5.3.10 Energy Suite Generierung

Einleitung

SiVArc erweitert seine Unterstützung bei der Generierung auf Clients wie Energy Suite. Energy Suite generiert Bilder und Bildobjekte basierend auf den konfigurierten Bildregeln, die als System-Bildregeln bezeichnet werden. Sie können die Generierung basierend auf den SiVArc-Bildregeln in Kombination mit den System-Bildregeln über das Dialogfeld der Station "Station und Steuerungen für SiVArc-Generierung auswählen" auswählen. Im Dialogfeld der Station "Station und Steuerungen für SiVArc-Generierung auswählen" können Sie in der Spalte "Regelsatz" den Regeltyp, den Sie generieren möchten, wie folgt auswählen:

- Vom Anwender erstellte Regeln - Bezieht sich auf vom Anwender erstellte Regeln innerhalb von SiVArc
- Energy Suite-Regeln - Bezieht sich auf vom Client Energy suite erstellte Regeln
- Alle Regeln - Kombination von SiVArc- und Energy Suite-Regeln

Hinweis

Bei Auswahl von Alle Regeln werden von SiVArc erstellte Regeln gegenüber Energy Suite - Regeln bei der Generierung mit Priorität behandelt.

- Ist Energy Suite nicht installiert, werden bei der Generierung durch SiVArc "Vom Anwender erstellte Regeln" in der Dropdown-Liste der Regeln nur im Lesemodus angezeigt.
- Sind Energy Suite-Regeln verfügbar, jedoch keine gültige Energy Suite-Lizenz vorhanden, wird bei dem Versuch, eine Generierung mit SiVArc vorzunehmen, sofern die Einstellung "Alle Regeln" gewählt ist, die Fehlermeldung angezeigt, dass die Lizenz fehlt.

Generierungsszenarien von SiVArc-Bildregeln und System-Bildregeln

Vom Anwender erstellte Regeln	System-Bildregeln	Ergebnis
Regel konfiguriert in SiVArc für Bild 1	Regel konfiguriert von Energy Suite für Bild 1	System generiert Bild 1
Regel konfiguriert in SiVArc für Bild 1 mit Schaltfläche 1	Regel konfiguriert von Energy Suite für Bild 1 mit Schaltfläche 2	System generiert Schaltfläche 1 und Schaltfläche 2 auf Bild 1
Regel konfiguriert in SiVArc für Bild 1 mit Schaltfläche 1	Regel konfiguriert von Energy Suite für Bild 1 mit Schaltfläche 1	System generiert Schaltfläche 1 auf Bild 1
Anwender erstellt Bild 1 mit Schaltfläche 1	Regel konfiguriert von Energy Suite für Bild 1 mit Schaltfläche 1	System benennt Bild 1 in Bild 1_Umbenannt um System generiert Bild 1 und Schaltfläche 1 darauf

Vom Anwender erstellte Regeln	System-Bildregeln	Ergebnis
Anwender erstellt Bild 1 mit Schaltfläche 1	Regel konfiguriert von Energy Suite für Bild 1 mit Schaltfläche 2	System benennt Bild 1 mit Schaltfläche 1 in Bild 1_Umbenannt um System generiert von Energy Suite-Regel erstelltes Bild 1 und positioniert Schaltfläche 2 auf Bild 1
Anwender erstellt Bild 1 mit Schaltfläche 1 Regel konfiguriert in SiVArc für Bild 1 mit Schaltfläche 2	Regel konfiguriert von Energy Suite für Bild 1 mit Schaltfläche 3	System benennt Bild 1 mit Schaltfläche 1 in Bild 1_Umbenannt um System generiert Schaltfläche 2 und/oder Schaltfläche 3 auf Bild 1 basierend auf Regelauswahl
Kopierregel konfiguriert in SiVArc für Bild 1	Regel konfiguriert von Energy Suite für Bild 1	System generiert Bild 1 und Bild 1 wird in Bild 1_Umbenannt umbenannt System generiert Bild 1 mittels Energy Suite-Regel
Kopierregel konfiguriert in SiVArc für Bild 1 (zweite Generierung verwendet diese Regel)	Regel konfiguriert von Energy Suite für Bild 1 (erste Generierung verwendet diese Regel)	Bei der ersten Generierung generiert das System Bild 1 Die Kopierregel wird bei der Generierung nicht berücksichtigt und wird mit einem Fehler ignoriert

Siehe auch

Generierübersicht (Seite 43)

Editor "Variablenregeln" (Seite 29)

5.4 SiVArc in HMI-Editoren

5.4.1 Eigenschaftskonfigurator

Beschreibung

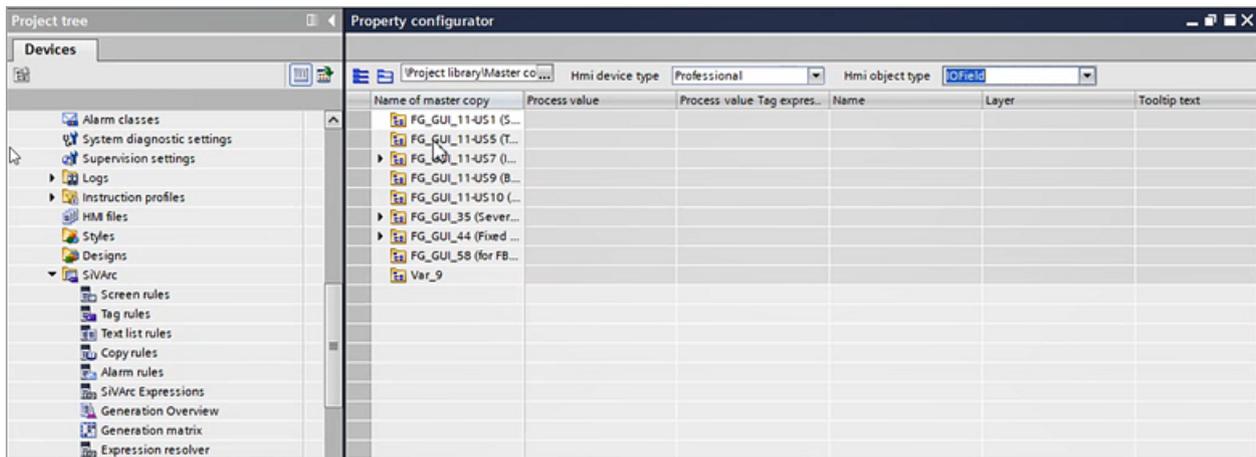
Der Eigenschaftskonfigurator ermöglicht Ihnen das Konfigurieren der SiVArc-Eigenschaften eines im Ordner **Masterkopien** abgelegten Bibliotheksobjekts. Der Eigenschaftskonfigurator bietet eine alternative Möglichkeit zum Auswählen der anzuzeigenden Bibliotheksobjekte und zum Konfigurieren der Eigenschaften für einen bestimmten HMI-Gerätetyp. Es besteht die Möglichkeit zum Durchsuchen verschiedener Ordner Ebenen im Ordner "Masterkopien", aus denen der Konfigurator die konfigurierbaren Eigenschaften für ein ausgewähltes Objekt und für einen ausgewählten HMI-Gerätetyp anzeigen soll. Der Editor "Eigenschaftskonfigurator" ermöglicht Ihnen das Erledigen notwendiger Aufgaben, die von anderen Editoren unterstützt werden.

Der Plugin-Editor für SiVArC fungiert als Reflexion der Objekte im "Eigenschaftskonfigurator". Der Editor "Eigenschaftskonfigurator" bietet Optionen wie unten aufgeführt:

- Bibliotheksordner auswählen – ermöglicht das Durchsuchen verschiedener Ordner in den Masterkopien mit den Masterkopieobjekten des ausgewählten HMI-Gerätetyps und HMI-Objekttyps
- HMI-Gerätetyp – Klappliste mit den unterstützten HMI-Gerätetypen
- HMI-Objekttyp – Klappliste mit den unterstützten HMI-Objekttypen

Betrachten wir als Beispiel das Konfigurieren von SiVArC-Eigenschaften eines Bibliotheksobjekts wie Button.

1. Im Editor "Eigenschaftskonfigurator" durchsuchen Sie einen Ordner innerhalb von "Projektbibliothek\Masterkopien", aus dem die SiVArC-Eigenschaften in Ordnern mit den Bibliotheksobjekten angezeigt/konfiguriert werden sollen. Sie haben auch die Möglichkeit zum Auswählen des Stammknotens, wenn die SiVArC-Eigenschaften für ein Bibliotheksobjekt angezeigt/konfiguriert werden sollen. Der nachstehende Screenshot zeigt die Ordnerauswahl innerhalb von "Projektbibliothek\Masterkopien":

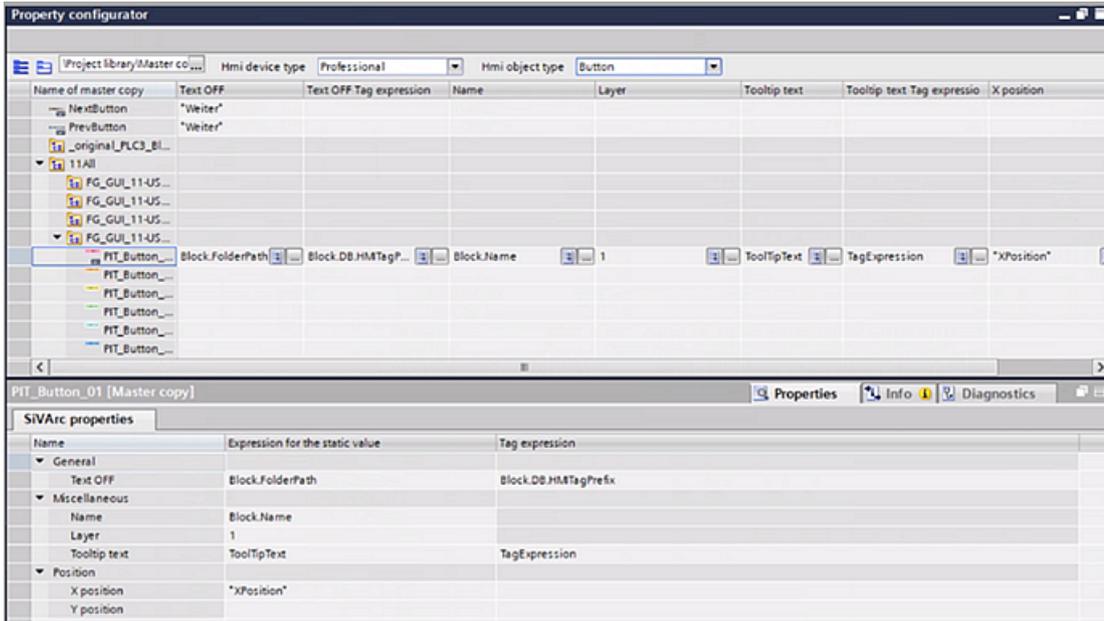


2. Wählen Sie aus der Klappliste "HMI-Gerätetyp" den Gerätetyp "Professional". In der Unified-Umgebung werden Geräte des Typs WinCC Unified angezeigt.
3. Wählen Sie wie im nachstehenden Screenshot dargestellt das Objekt Button aus der Klappliste "HMI-Objekttyp". Der Editor "Eigenschaftskonfigurator" zeigt die Masterkopien-Ordner an, die das Objekt Button enthalten.

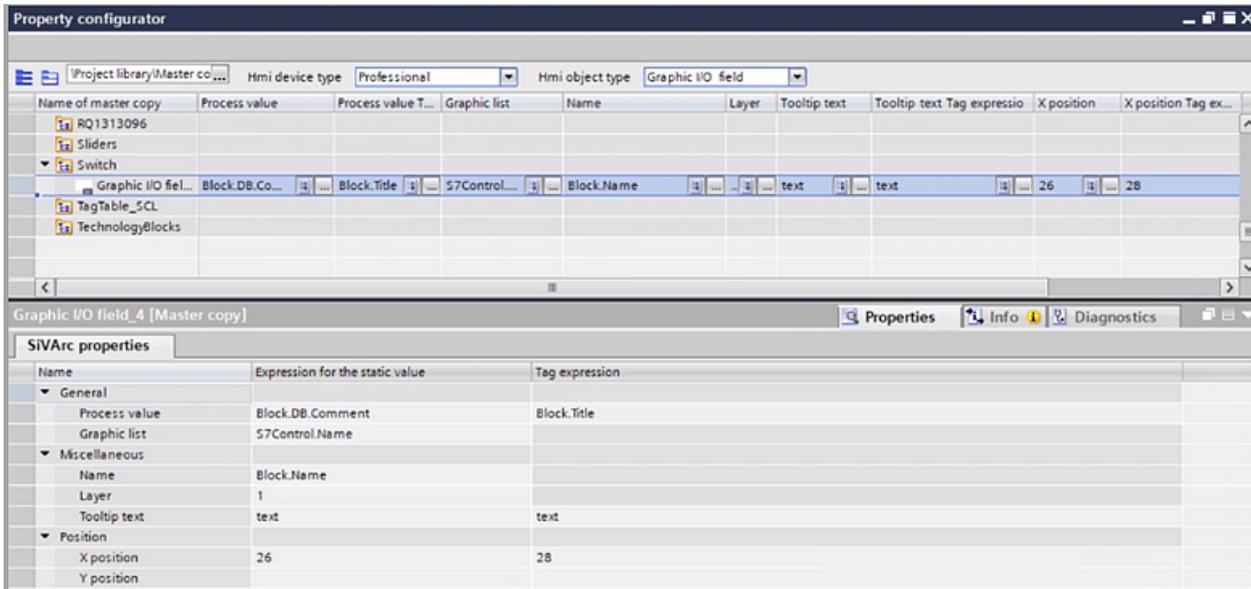
Hinweis

Die Spalte "HMI-Objekttyp" ist abhängig von dem HMI-Gerätetyp, den Sie in der Spalte "HMI-Gerätetyp" wählen.

- Die Spalten im Editor "Eigenschaftskonfigurator" enthalten die SiVArC-Eigenschaften, die mit SiVArC-Ausdrücken für Button konfiguriert werden können, siehe Screenshot unten:



- Wenn Sie "Grafisches EA-Feld" als "HMI-Objektyp" gewählt haben, zeigt der Editor "Eigenschaftskonfigurator" die von SiVArC unterstützten Spalten an. Jedoch zeigt der Plugin-Editor für SiVArC die auf dem Engineering-System aktivierten Eigenschaften an. Die Spalten im Editor "Eigenschaftskonfigurator" enthalten die Eigenschaften im Plugin-Editor für SiVArC, siehe Screenshot unten. Der "Eigenschaftskonfigurator" zeigt die von SiVArCunterstützten Eigenschaften an, während der Plugin-Editor für SiVArC die auf dem Engineering-System aktivierten Eigenschaften anzeigt.



Wichtige Hinweise:

- Jede an den Eigenschaften des Objekts vorgenommene Bearbeitung/Löschung im Ordner "Masterkopien" wirkt sich auch im Editor "Eigenschaftskonfigurator" aus und umgekehrt.
- Die Umbenennung oder Neuordnung eines Bibliotheksobjekts im Ordner "Masterkopien" schlägt sich im Editor "Eigenschaftskonfigurator" nieder.

Siehe auch

Editor "SiVArc-Ausdrücke" (Seite 33)

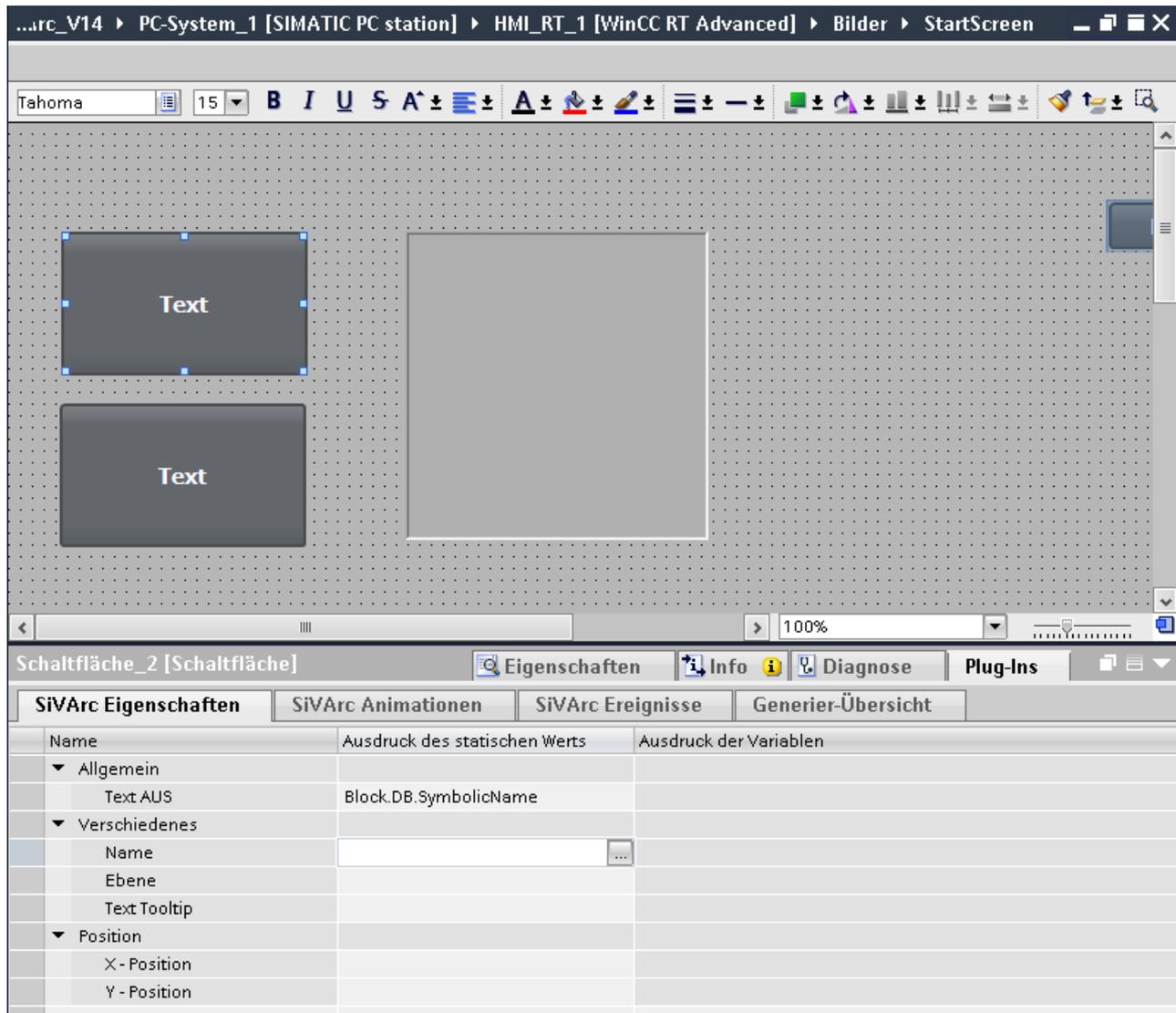
5.4.2 Registerkarte "SiVArc-Eigenschaften"

Beschreibung

Eine SiVArc-Eigenschaft ist eine Objekteigenschaft, die mit Hilfe eines SiVArc-Ausdrucks entweder statisch oder dynamisch projiziert wird.

Auf der Registerkarte "SiVArc-Eigenschaften" projizieren Sie mit Hilfe von SiVArc-Ausdrücken die Eigenschaften einer Textliste, eines Bilds oder Bildobjekts. Anschließend legen Sie das projizierte Objekt in der Projektbibliothek ab. Die SiVArc-Ausdrücke werden bei der Generierung der Visualisierung ausgewertet. Sie können Farbeigenschaften für Bilder und Bildobjekte in Unified-Geräten konfigurieren. Sie können Werte für Farbeigenschaften im RGB-Format "0-255, 0-255, 0-255" konfigurieren, z. B. "120, 120, 120".

Die Registerkarte "SiVArc-Eigenschaften" ist nur für Objekte verfügbar, die von SiVArc unterstützt werden.



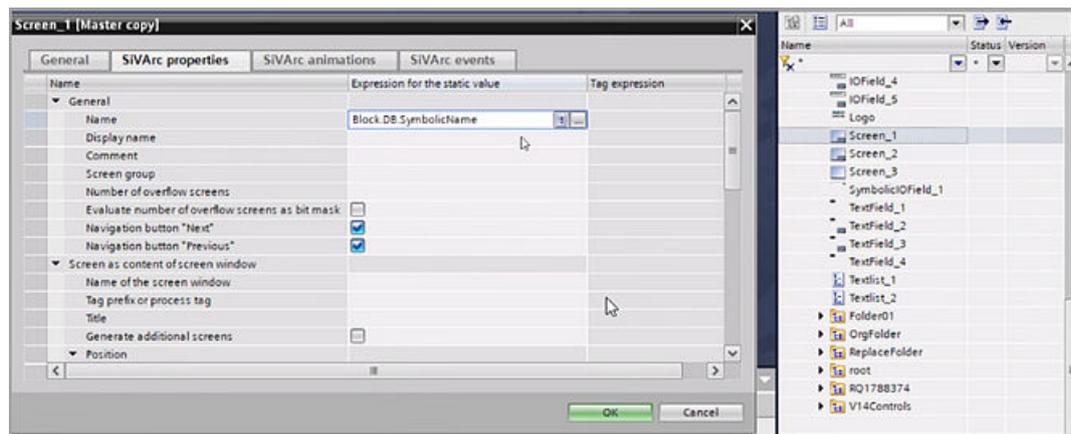
Bei der Projektierung der Überlaufbilder können Sie die Navigationsschaltflächen "Zurück" und "Weiter" separat im Editor SiVArc-Plug-Ins projektieren. Während der Projektaktualisierung:

- Wurde in der Vorgängerversion die Option "Navigationsschaltfläche generieren" gewählt, werden die Bildkopiervorlagen auf folgende Weise auf die neueste Version aktualisiert:
 - Wurde in der Vorgängerversion die Option "Navigationsschaltfläche generieren" gewählt, werden die Schaltflächen "Weiter" und "Zurück" in der neuesten Version automatisch gewählt.
 - Wurde in der Vorgängerversion die Option "Navigationsschaltfläche generieren" nicht gewählt, werden die Schaltflächen "Weiter" und "Zurück" in der neuesten Version nicht gewählt.
- Für Bilder, die auf dem HMI-Gerät in der älteren Version vorliegen, sind die Optionen "Navigationsschaltfläche Zurück" und "Navigationsschaltfläche Weiter" ausgewählt.
- Für Bilder, die als Bibliothekstyp in der älteren Version vorliegen, sind die Optionen "Navigationsschaltfläche Zurück" und "Navigationsschaltfläche Weiter" ausgewählt.

Zugriff auf SiVArC-Eigenschaft über Masterkopie

Eine alternative Möglichkeit zur Navigation durch den Projektordner, um die Eigenschaft eines Objekts anzuzeigen bzw. zu bearbeiten, ist über den Ordner Projektbibliothek > Masterkopien. Gehen Sie wie folgt vor:

1. Klicken Sie mit der rechten Maustaste auf das Objekt, das Sie bearbeiten möchten, und wählen Sie die Option Eigenschaften.
2. Das Popup SiVArC-Eigenschaften wird geöffnet. Indem Sie über die Masterkopie auf die SiVArC-Eigenschaften zugreifen, können Sie die Eigenschaften über den lokalen Ausdruck und den globalen Ausdruck konfigurieren.
3. Sie müssen die Änderungen nach dem Aktualisieren der SiVArC-Eigenschaften für ein Objekt speichern.



Hinweis

Mehrsprachige Unterstützung für Bildobjekte ist nicht verfügbar, das gilt für alle Geräte mit Ausnahme von Geräten des Typs WinCC Unified.

Layout

Die Registerkarte enthält drei Spalten:

- **Name**
In dieser Spalte sind die verfügbaren Eigenschaften aufgeführt.
- **Ausdruck des statischen Werts**
In dieser Spalte projektieren Sie eine Eigenschaft mit einem festen Wert oder einem SiVArC-Ausdruck, der eine Zeichenkette oder eine Zahl zurückgibt.
Feste Werte werden beim Generieren der Visualisierung in jede Instanz dieser Kopiervorlage eingetragen. Achten Sie zum Beispiel bei der Eigenschaft "Name" darauf, dass die Eindeutigkeit des Objektname auch bei Mehrfachverwendung in einem Bild gewährleistet ist.
- **Ausdruck der Variablen**
In dieser Spalte projektieren Sie eine Eigenschaft mit einem Variablennamen oder einem SiVArC-Ausdruck, der einen Variablennamen zurückgibt.

5.4.3 Registerkarte "SiVArc-Ereignisse"

Beschreibung

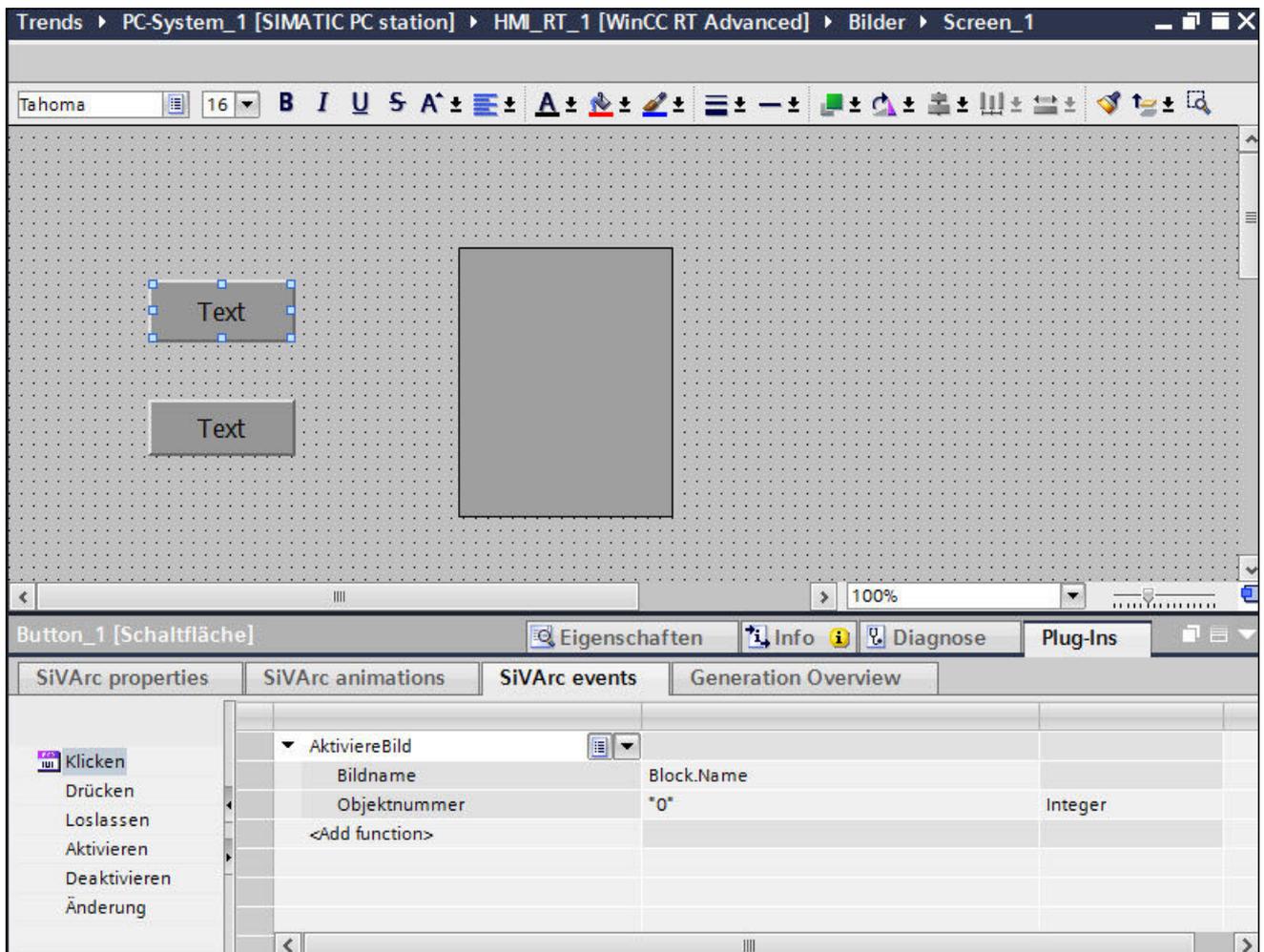
Fall 1: Auf der Registerkarte "SiVArc-Ereignisse" projektieren Sie eine Funktionsliste an ein Ereignis einer Generiervorlage eines Bildes oder Bildobjekts. Sie können Ereignisse für Bilder und Bildobjekte konfigurieren. In der Funktionsliste fügen Sie Systemfunktionen oder eine Skriptfunktion ein.

Die Parameter der Systemfunktion oder des Skripts können Sie mit SiVArc-Ausdrücken parametrieren. Konfigurieren Sie die Bildregeln für die SiVArc-Generierung. Nach der Generierung wird der Ausdruck, den Sie unter SiVArc-Ereignissen festlegen, aufgelöst und im Bildereignisfenster unter SiVArc Plug-ins angezeigt.

Fall 2: Bilder mit konfigurierten Ereignissen können unter Bibliothekstypen positioniert werden und werden als Bildobjekte in Bildregeln verwendet. Nach der SiVArc-Generierung wird das Bild unter **Bilder** in der Betriebsart PNV generiert.

Geräte des Typs WinCC Comfort Unified / Unified SCADA RT:

- Es werden globale Module auf Basis von Java-Skripts unterstützt.
- Die Aktivierung eines Bildereignisses wird durch die Systemfunktion "Bild ändern" konfiguriert.



Layout

Spalte 1: In Spalte 1 wählen Sie die Funktion oder das Skript aus.

Spalte 2: In Spalte 2 geben Sie einen SiVArc-Ausdruck ein.

Spalte 3: Wenn Sie ein Skript ausgewählt haben, wählen Sie in Spalte 3 einen Datentyp aus.

Bibliothekselemente

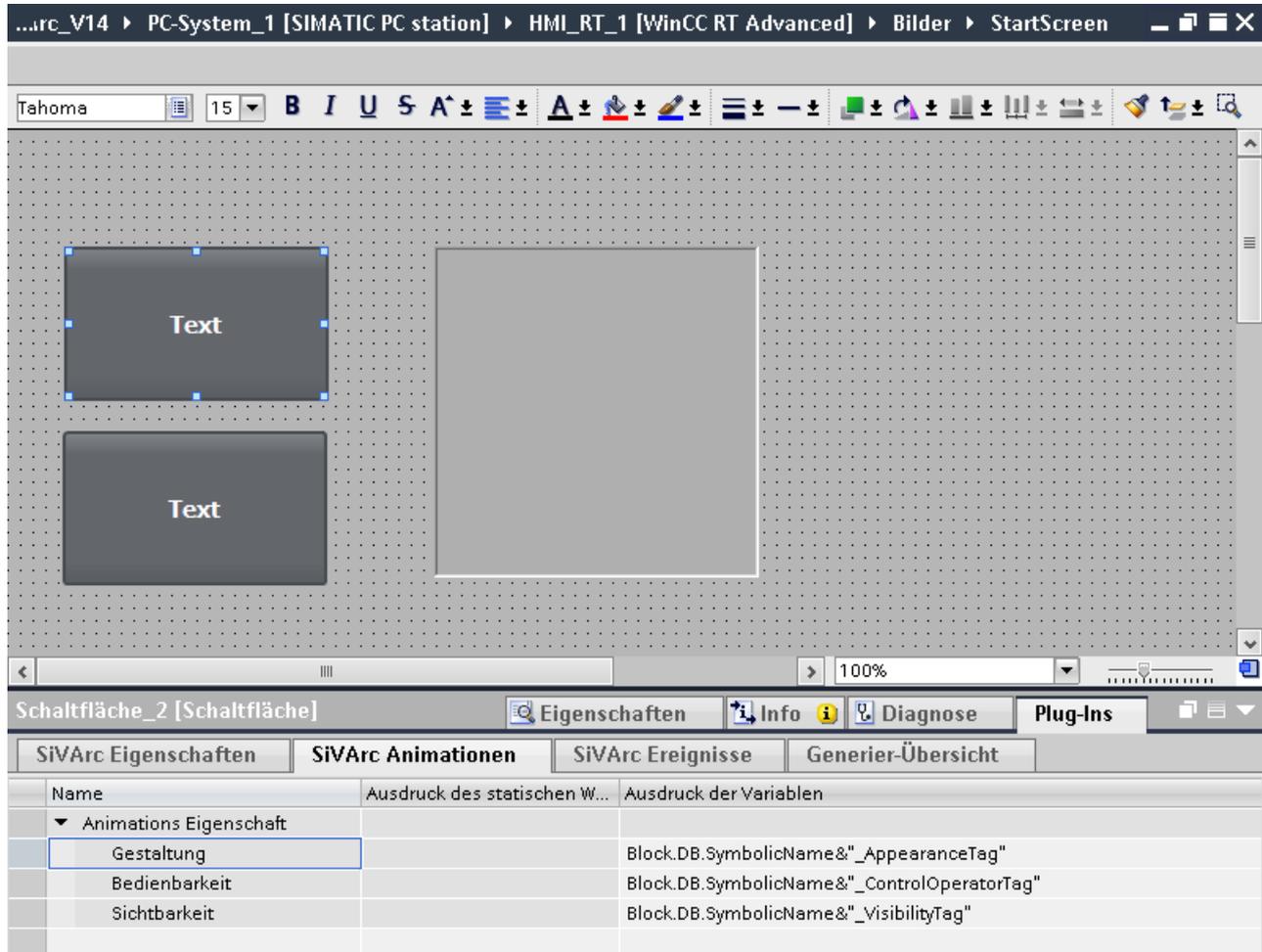
Ereignisse für Bibliothekselemente können Sie durch Öffnen des Eigenschaftsdialogs konfigurieren, und das SiVArc-Eigenschaftsregister ist für Anzeige-/Bearbeitungsoptionen verfügbar.

5.4.4 Registerkarte "SiVArc-Animationen"

Beschreibung

Auf der Registerkarte "SiVArc-Animationen" werden am Bildobjekt konfigurierte Animationen aufgelistet.

Die Registerkarte "SiVArc-Eigenschaften" ist nur für HMI-Objekte verfügbar, die von SiVArc unterstützt werden.



Sie können Animationen für unterstützte Bibliotheksobjekte konfigurieren, indem Sie mit der rechten Maustaste auf ein Objekt klicken, und der Plugin-Editor für SiVArc ist für Anzeige-/ Bearbeitungsoptionen verfügbar.

Layout

Die Registerkarte "SiVArc-Animationen" enthält die folgenden Spalten:

- Name
In dieser Spalte werden die unter "Eigenschaften > Animationen" projektierten Animationen aufgelistet.
- Ausdruck des statischen Werts
Diese Spalte ist für Animationen nicht editierbar.
- Ausdruck der Variablen
In dieser Spalte parametrieren Sie die Prozessvariablen für die Animation mit einem SiVArc-Ausdruck. Der SiVArc-Ausdruck muss einen Variablennamen zurückgeben.

Bibliotheksubjekte

Animationen für Bibliothekselemente können Sie durch Öffnen des Eigenschaftsdialogs konfigurieren, und das SiVArc-Animationsregister ist für Anzeige-/Bearbeitungsoptionen verfügbar.

5.4.5 Registerkarte "Generierübersicht"

Beschreibung

Nach der ersten Generierung wird die Registerkarte "Generier-Übersicht" im Inspektorfenster eines generierten Bildes angezeigt. Die Anzahl der angezeigten Objekte beschränkt sich auf die im gewählten Bild generierten Anzeige- und Bedienobjekte.

Die Registerkarte "Generier-Übersicht" umfasst bis auf folgende Ausnahmen die gleichen Bearbeitungsmöglichkeiten wie der SiVArc-Editor "Generier-Übersicht":

- Filterfunktion
- Sortierfunktion
- Schaltflächen "Alles öffnen" und "Alles ausklappen"

	Name of screen	Name of screen item	Master copy / library type	HMI station	S7 station	Rule Trigger	Screen rule	Generated by matrix	Layout field	Call path	G
1	▶ Template_1	Template_1	Template_1	HMI_RT_1	PLC_1	Block_2, Block_2_DB	Screen rule_4	<input type="checkbox"/>		main/block_2	1
2	▶ Template_2	Template_2	Template_2	HMI_RT_1	PLC_1	Block_2, Block_2_DB	Screen rule_6	<input type="checkbox"/>		main/block_2	1
3	▶ Template_1	Template_1	Template_1	HMI_1	PLC_1	Block_2, Block_2_DB	Screen rule_4	<input type="checkbox"/>		main/block_2	1
4	▶ Template_2	Template_2	Template_2	HMI_1	PLC_1	Block_2, Block_2_DB	Screen rule_6	<input type="checkbox"/>		main/block_2	1

Auf den Registerkarten "Vorlagenbilder/Bildelemente" und "Pop-up-Bilder/Bildelemente" werden die generierten Objektdetails von Vorlagenbildern/Bildelementen bzw. von Pop-up-Bildern/Bildelementen und -regeln angezeigt.

Bei generierten Bildobjekten erscheint die Generierübersicht auf der Registerkarte "Plug-ins".

5.5 SiVArc in PLC-Editoren

5.5.1 Unterstützung für Software-Einheiten

Software-Einheiten in SiVArc

Software-Einheiten agieren als Container, der verschiedene vom Anwender definierte Programme trennt. Mit SiVArc können Sie Ausdrücke für verschiedene Eigenschaften von HMI-Objekten konfigurieren, die sich in den Namen der Software-Einheit auflösen, die den bei der Erstellung der SiVArc-Regel verwendeten PLC-Baustein enthält. Weitere Informationen zu Software-Einheiten finden Sie in der *TIA Portal-Hilfe*.

Voraussetzung

- Ein vorhandenes TIA Portal-Projekt
- PLC mit Firmware V2.6 und höher
- Mit einem HMI-Gerät verbundener PLC

Vorgehensweise

Um SiVArc -Ausdrücke für Software-Einheiten zu konfigurieren, führen Sie folgende Schritte aus:

1. Klicken Sie unter dem Software-Einheiten-Ordner auf "Neue Software-Einheit hinzufügen", um eine neue Software-Einheit hinzuzufügen.
2. Konfigurieren Sie die Programmbausteine gemäß Ihren Anforderungen.
3. Fügen Sie ein Bild hinzu und konfigurieren Sie den Bildnamen mit einem Ausdruck, der sich in den Namen der Software-Einheit auflöst. Beispiel: Name = "Software-Einheit.Name". Weitere Informationen zum Konfigurieren von Bildern finden Sie unter SiVArc-Editoren (Seite 27).
4. Konfigurieren Sie eine SiVArc-Bildregel, die aus einem aus der Software-Einheit ausgewählten Programmbaustein besteht.
5. Starten Sie die SiVArc-Generierung. Der generierte Bildname löst sich in den Namen der Software-Einheit auf, die den Programmbaustein enthält, der bei der Erstellung der SiVArc-Bildregel verwendet wurde.

Hinweis

Die Unterstützung für die Software-Einheit ist für alle HMI-Objekte mit Hilfe der SiVArc-Plugin-Editoren konfigurierbar.

5.5.2 SiVArC-Unterstützung für Programmbausteine mit Hilfe von Structured Control Language

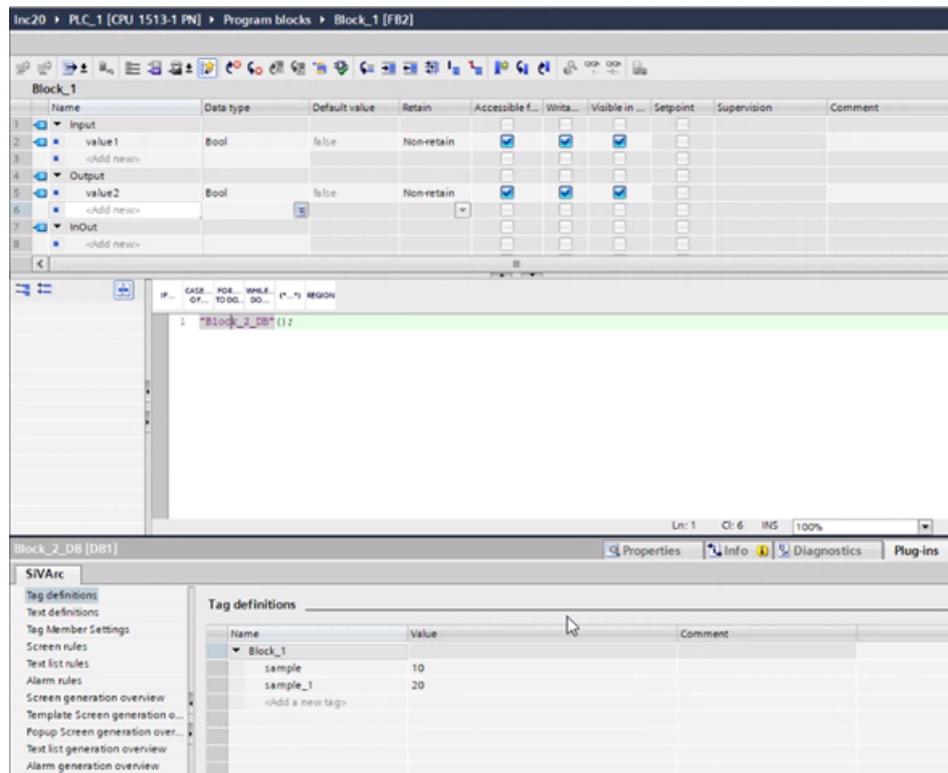
SiVArC-Unterstützung für Programmbausteine mit Hilfe von SCL

Mit Programmiersprachen wie AWL, FUP, KOP unterstützt SiVArC SCL für Programmbausteine über Einzel- oder Multiinstanz-Aufrufe. Sie können die SiVArC-Eigenschaften im Plug-in-Editor projektieren. Dies gilt z. B. für Text-/Variablendefinitionen/Variablen Mitgliedeinstellungen/ Regeleditoren für SCL-Bausteine. SiVArC unterstützt die Projektierung für SCL nur auf Bausteinebene. Sie können SiVArC-Daten auf der Bausteinebene projektieren, die für alle Instanzaufrufe in einem SCL-Baustein gültig sind.

Beispielszenario für die Verwendung von SCL in den Variablendefinitionen

Betrachten wir die Projektierung von Variablendefinitionen für den SCL-Baustein. Sie können Variablendefinitionen für den Aufrufbaustein (SCL) projektieren. Die Variablendefinitionen gelten für alle Instanzaufrufe im SCL-Baustein. Die definierten Variablen werden in SiVArC generiert.

1. Beispiel: Baustein "Block_2" wird in Baustein_1 mit Instanz-DB "Block_2_DB" aufgerufen.
2. Wählen Sie einen beliebigen Bereich im SCL-Programmieditor und klicken Sie auf SiVArC Plug-ins.
3. Definieren und projektieren Sie Variablen für "Block_1" wie in der Abbildung gezeigt:



4. Die definierte Variable kann in einem beliebigen SiVArc-Ausdruckseditor wie SiVArc-Eigenschaften für Bildobjekte oder Bedingungen im Regeleditor verwendet werden. Beim Projektieren von Bildobjekten in SiVArc Plug-ins geben Sie den Bildnamen mit der Variablen "Sample" ein. Weitere Einzelheiten zum Projektieren von Bildobjekten finden Sie im Abschnitt zum Editor "Bildregeln".
5. Beim Generieren mit SiVArc wird ein Bild mit dem Namen "10" generiert. Weitere Einzelheiten über den Generierungsvorgang finden Sie im Abschnitt Visualisierung generieren (Seite 205).

Hinweis

- Die Projektierung von Variablen-/Textdefinitionen/Variablen Mitgliedeinstellungen für SCL-Bausteine ist ähnlich wie für KOP-Bausteine. Weitere Einzelheiten zur Projektierung auf Bausteinebene finden Sie in der KOP-Dokumentation im Benutzerleitfaden von TIA Portal.
 - Variablen Mitgliedeinstellungen können nur in Geräten mit WinCC Advanced projektiert werden.
 - Bild-/Textlisten/ Alarmregeln ermöglichen den schnellen Zugriff für Anzeige und Bearbeitung der projektierten Regeln.
 - Bild-, Textlisten und Alarmregeln des Aufrufbausteins werden angezeigt, wenn ein Baustein mit Multiinstanz-Aufruf ausgewählt wird.
 - Die Übersicht über die Bild-/Textlisten/Alarmgenerierung ermöglicht einen schnellen Zugriff, mit dem die Anzeige der SiVArc-Generierung aufgerufen wird.
-

Wenn Sie PLC-Bausteinvariablen mit benutzerdefinierten Datentypen/Array-Datentypen und PLC-Variablen mit benutzerdefinierten Datentypen erstellen, werden im Fall von Unified HMI-Geräten bei der SiVArc-**Generierung > AllHmiTags** die HMI-Variablen für PLC-Bausteinvariablen und PLC-Variablen für benutzerdefinierte/Array-Datentypen erzeugt.

Siehe auch

Visualisierung generieren (Seite 205)

Arbeiten mit SiVArc

6.1 Layout planen

6.1.1 Positionierungsschemata

6.1.1.1 Übersicht zur Positionierung von generierten Objekten

Einleitung

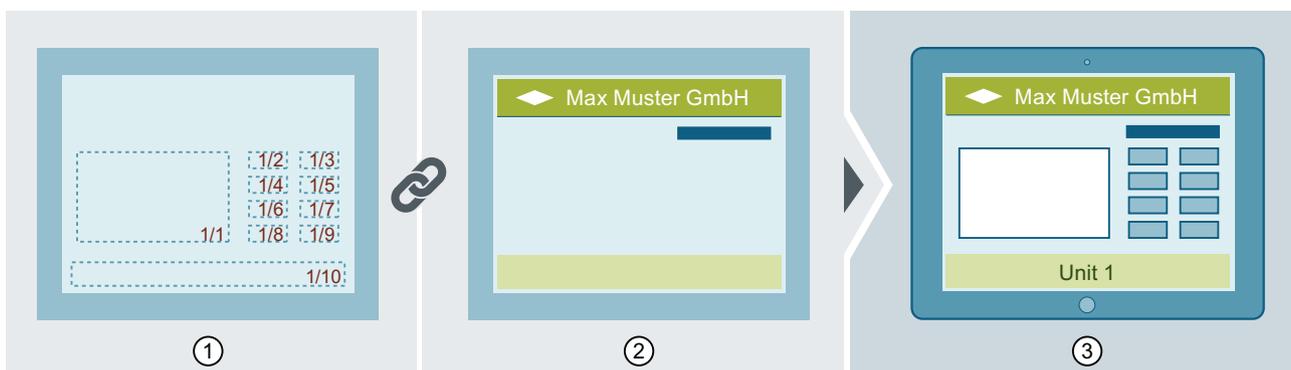
Mit SiVArc teilt sich das Layout Ihrer Anlagenbilder in folgende beiden Aufgaben auf:

- Grafische Gestaltung der Anlagenbilder und Anzeige- und Bedienobjekte wie in WinCC.
- Positionierung der generierten Anzeige- und Bedienobjekte

SiVArc stellt nachfolgend beschriebene Methoden zur Steuerung der Positionierung generierter Objekte zur Verfügung. Für alle Methoden gilt, dass manuelle Positionsänderungen nach der ersten Generierung auch bei den nächsten Generierungen erhalten bleiben.

Gesteuerte Positionierung

Sie steuern über Positionierungsschemata, an welcher Stelle im Anlagenbild die Anzeige- und Bildobjekte generiert werden. Dazu kombinieren Sie eine Generiervorlage eines Bildes mit Positionierungsfeldern eines definierten Schemas.



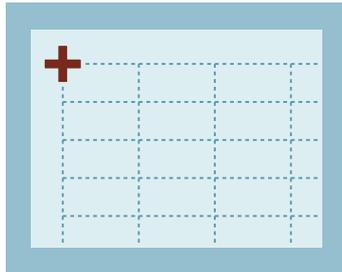
- ① Benutzerdefiniertes Positionierungsschema mit definierten Bildbereichen
- ② Generiervorlage für ein Bild
- ③ Generiertes Prozessbild

Freie Positionierung

Alternativ zur gesteuerten Positionierung positionieren Sie die generierten Objekte von Hand.

Dazu ist jeder Generiervorlage für Bilder das SiVArC-Positionierungsschema hinterlegt. Das SiVArC-Positionierungsschema ist ein konfigurierbares Raster.

Das folgende Bild zeigt die Startposition des Rasters:



In der Vorlage stellen Sie die Startposition des Rasters und die Zeilen- und Spaltenabstände ein.

Plantsection1 [Bild]				
SiVArC Eigenschaften		SiVArC Animationen	SiVArC Ereignisse	Generier-Übersicht
Name	Ausdruck des statischen Werts	Ausdruck der Variablen		
▶ Allgemein				
▶ Bild als Inhalt des Bildfen...				
▼ Positionierungsschema				
X - Position	150			
Y - Position	150			
Zeilenabstand	200			
Spaltenabstand	600			
▶ Darstellung				

Nach dem Generieren verschieben Sie die Objekte nach Bedarf. Diese Positionen bleiben auch bei der nächsten Generierung erhalten.

Zusätzlich können Sie für einzelne Objekte feste Positionskordinaten vergeben:

Plantsection1_DB_SymbIO [Symbolisches E/...				
SiVArC Eigenschaften		SiVArC Animationen	SiVArC Ereignisse	Generier-Übersicht
Name	Ausdruck des statischen Werts	Ausdruck der Variablen		
▶ Allgemein				
▶ Verschiedenes				
▼ Position				
X - Position	675			
Y - Position	100			

Priorität der Positionierungsmethoden

Wenn für ein Anzeige- und Bedienobjekt in den Bildregeln ein eigenes Positionierungsschema hinterlegt wurde, werden alle anderen Vorgaben zur Position beim Generieren ignoriert.

Wenn Sie kein eigenes Positionierungsschema hinterlegen, werden die generierten Anzeige- und Bedienobjekte entsprechend der festen Positionierung oder dem SiVArc-Positionierungsschema angeordnet.

Generierte Objekte mit fester Positionierung oder mit einem Positionierungsschema überdecken bereits bestehende Anzeige- und Bedienobjekte an einer konfigurierten Position.

SiVArc verarbeitet die einzelnen Positionierungsmethoden mit folgender Priorität:

1. Positionierungsschema
2. Feste Positionierung (SiVArc-Eigenschaft)
3. Feste Positionierung (WinCC-Eigenschaft)
4. Freie Positionierung

Wenn Sie ein Anzeige- und Bedienobjekt über ein Positionierungsschema generieren und im Bild positionieren, wird eine am Objekt projizierte feste Positionierung ignoriert. Ebenso wird dann das SiVArc-Positionierungsschema ignoriert.

Siehe auch

Positionierung nach definierten Schemata (Seite 61)

Feste Positionierung des generierten Objekts (Seite 71)

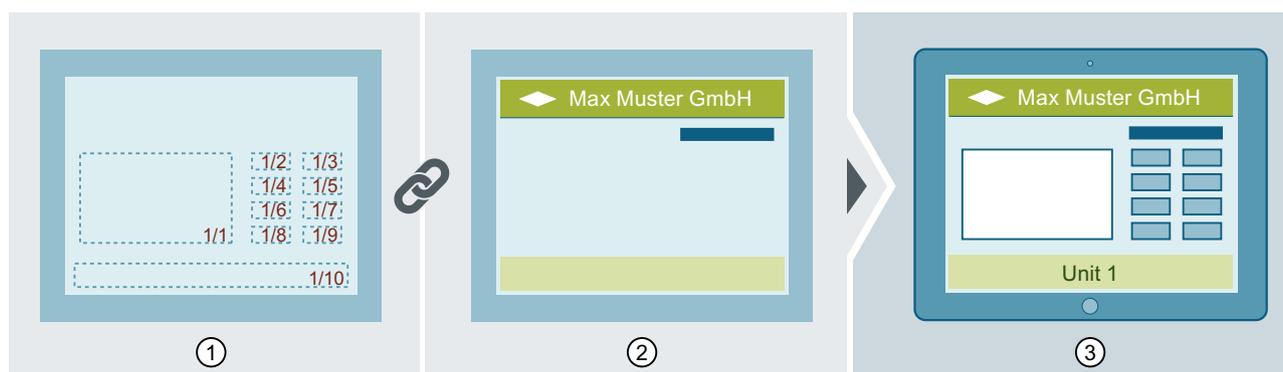
Freie Positionierung (Seite 72)

Planen eines Layouts (Seite 79)

6.1.1.2 Positionierung nach definierten Schemata

Definition

Ein benutzerdefiniertes Positionierungsschema ist ein pixelgenau konfigurierbares Raster. Sie können dieses Raster in unterschiedliche Bereiche einteilen. Dieses Schema ordnen Sie einem Bild zu. Anzeige- und Bedienobjekte werden von SiVArc dann direkt in den gewünschten Bereich generiert.



- ① Benutzerdefiniertes Positionierungsschema mit definierten Bildbereichen
- ② Generiertvorlage für ein Bild
- ③ Generiertes Prozessbild

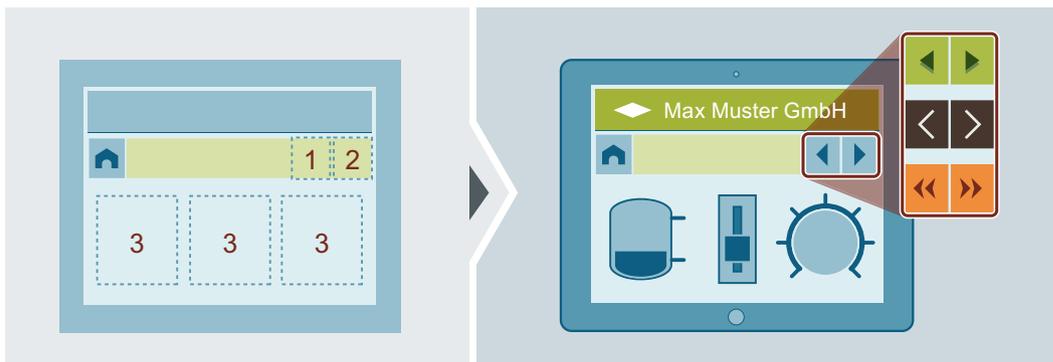
Verwendung

Wenn Ihr Projekt eine pixelgenaue und standardisierte Positionierung der Anzeige- und Bedienobjekte erfordert, verwenden Sie ein eigenes Positionierungsschema aus der Bibliothek.

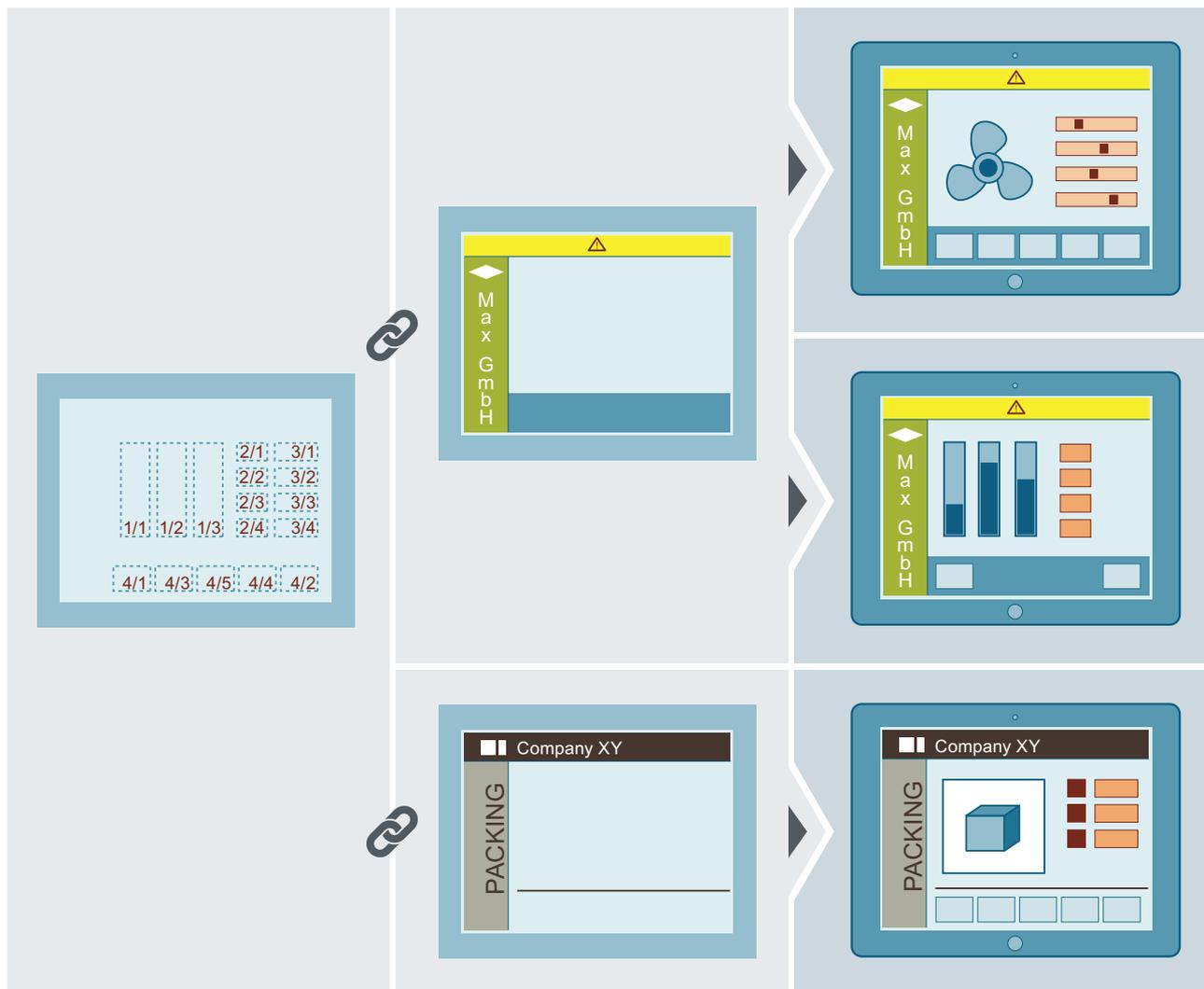
Vorteile

Ein benutzerdefiniertes Positionierungsschema bietet mehrere Vorteile:

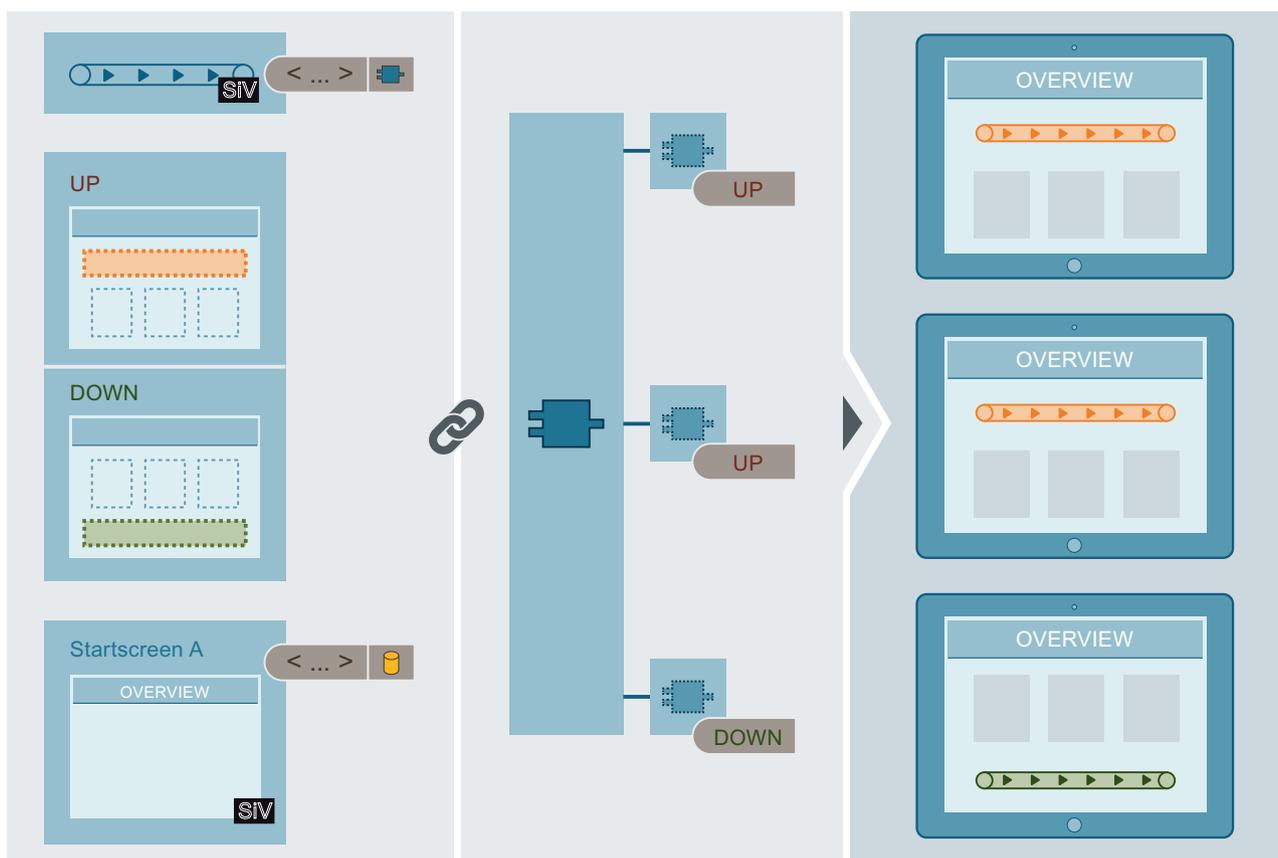
- Erhöhung der Standardisierung im Projekt
Mit eigenen Positionierungsschemata steuern und verwalten Sie zentral die Anordnung der generierten Objekte auf unterschiedlichen Bediengeräten. Innerhalb des Schemas können Sie auch gezielt und individuell Navigationsschaltflächen platzieren.



- Trennung von Layout und Positionierung
Aufgrund der Trennung von Layout und Positionierung können Sie bestehende Positionierungsschemata auch über Projekte hinweg verwenden.



- **Dynamisches Positionierungsschema**
Ein Positionierungsschema kann den Bildern dynamisch zugeordnet werden, z. B. über Bedingungen im Anwenderprogramm. Auf diese Weise reduzieren Sie die Anzahl von Bildregeln im SiVArc-Projekt.
Das folgende Bild zeigt ein die Bildregel für eine Fließbandgrafik, die abhängig von einer SiVArc-Variablen am Netzwerk im generierten Bild oben oder unten angeordnet wird:



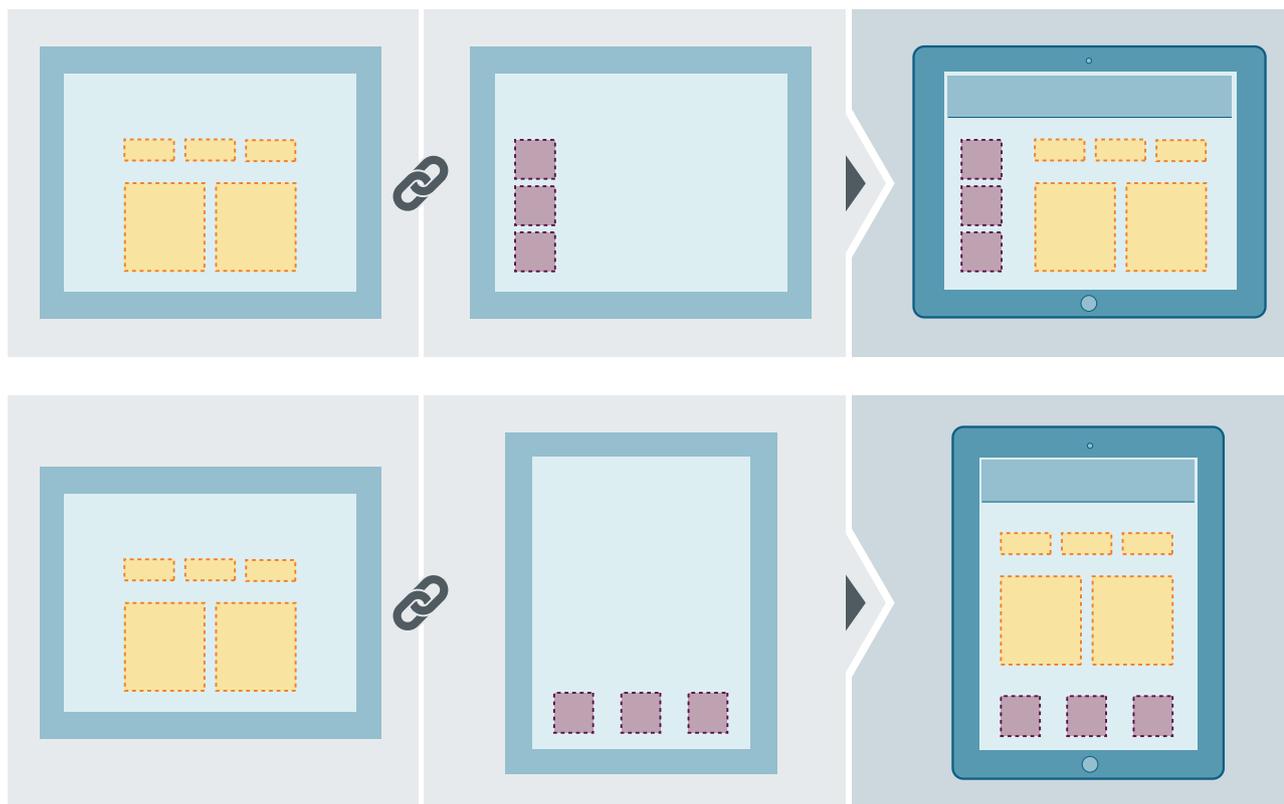
SiVArc SiVArc-Generiervorlage

< ... >  SiVArc-Eigenschaft mit referenzierter Textquelle

 Funktionsbaustein

 Instanzierter Funktionsbaustein

- **Kombiniertes Positionierungsschema**
Ein bestehendes Positionierungsschema ist mit einem anderen Schema kombinierbar. Auf diese Weise teilen Sie z. B. einen Anzeigebereich in Module, die Sie beliebig als Varianten für unterschiedliche Bediengeräte zusammenstellen können.
Das folgende Bild zeigt ein Positionierungsschema, dem unterschiedliche andere Schemas zugeordnet werden können:



Wenn im übergeordneten Positionierungsschema ein Modul vorhanden ist, das gleichnamig ist mit dem eines darunterliegenden Schemas, wird der darunterliegende Bereich ignoriert.

- Vorab-Ansicht der Positionierung
- Planbares Positionieren vor dem ersten Generat
- Geringe Fehleranfälligkeit
- Zentrale Verfügbarkeit von Layoutvarianten

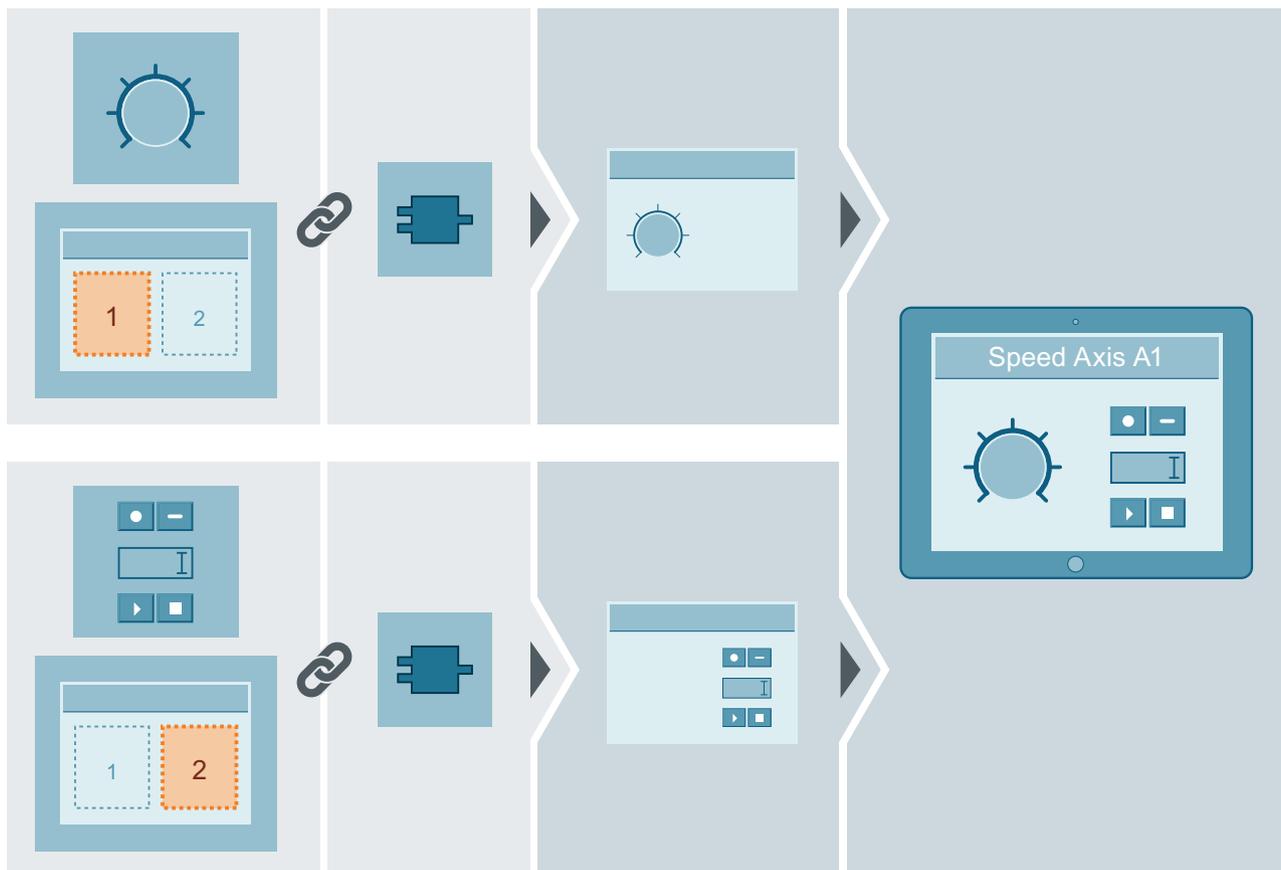
Hinweis

Positionierungsschemata für Pop-up-Bilder

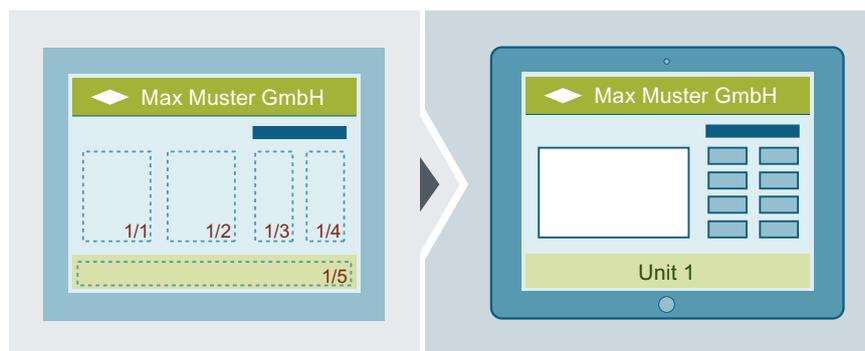
Ein Positionierungsschema für ein Pop-up-Bild ist für kein anderes Anzeige- und Bedienobjekt verwendbar.

Funktionsweise

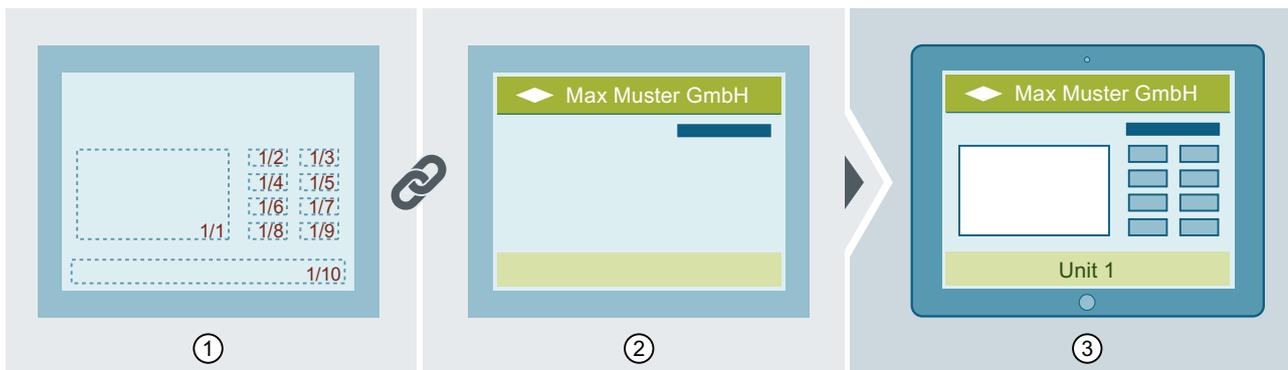
Sie können einem Anzeige- und Bedienobjekt in einer Bildregel die Information mitgeben, in welchem Bereich des Positionierungsschemas es platziert werden soll. Wenn Sie ein kombiniertes Schema verwenden, können Sie unabhängig von der Schachtelungstiefe des Schemas alle enthaltenen Positionierungsfelder mit Generiervorlagen verschalten.



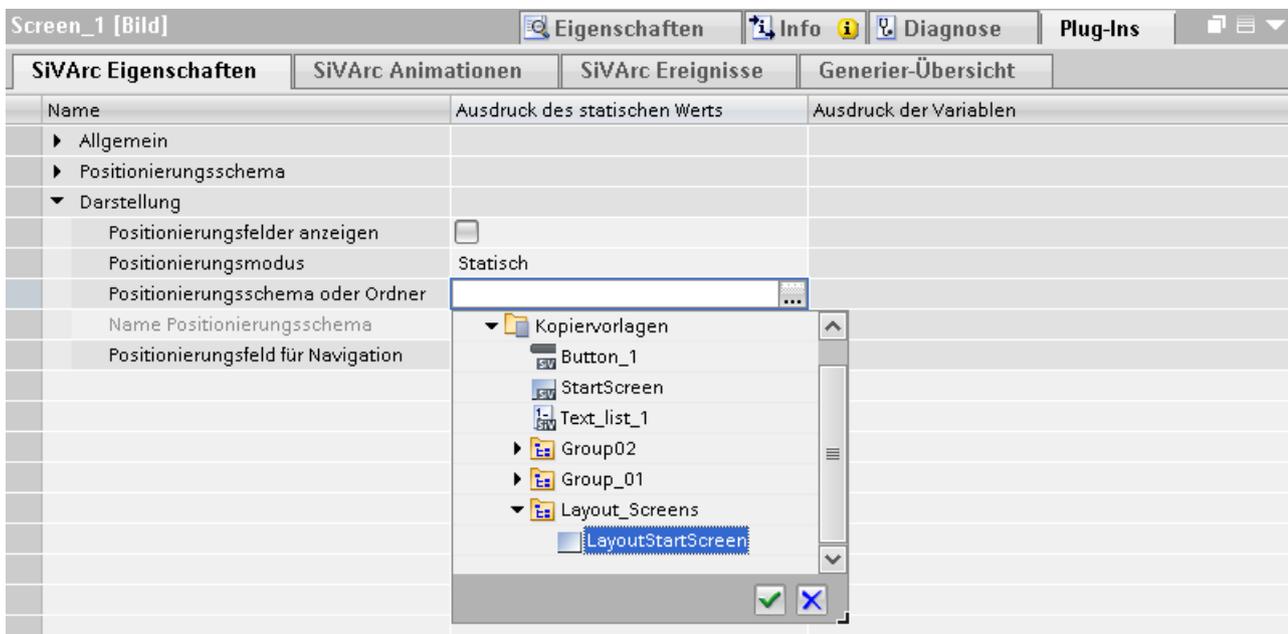
Ein Positionierungsschema können Sie wie eine Generiervorlage für ein Bild direkt in einer Bildregel verwenden. Dann sind Layout und Positionierung in der gleichen Vorlage enthalten:



Um Layout und Positionierung zu trennen, ordnen Sie das Positionierungsschema einer Generiervorlage für ein Bild fest oder dynamisch zu:



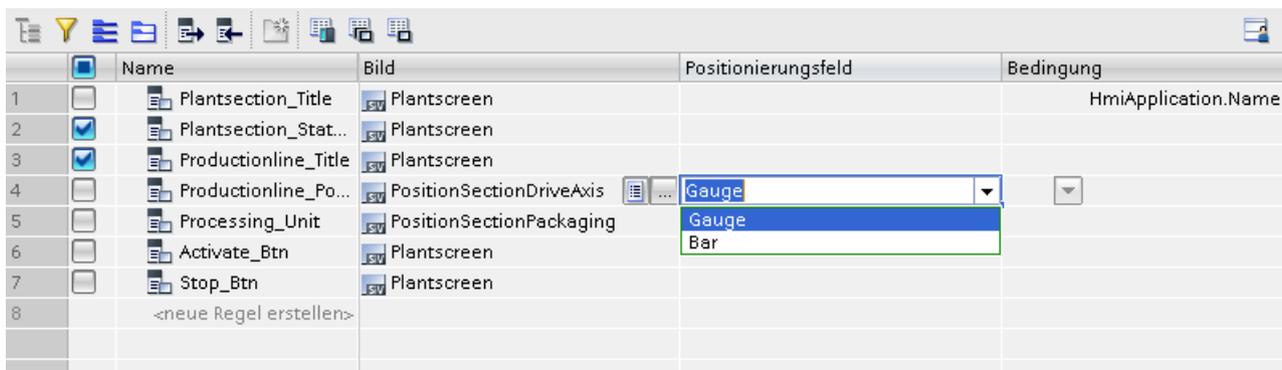
Die folgende Abbildung zeigt die dazugehörige Einstellung in den SiVArC-Eigenschaften einer Generiervorlage für Bilder:



Positionierungsfelder

In der Bildregel wählen Sie den Bereich des Positionierungsschemas aus, in den das Objekt generiert wird. SiVArC generiert das Bildobjekt in diesen Bereich in das Positionierungsfeld mit dem Index 1. Das nächste generierte Objekt wird in das Feld mit dem Index 2 generiert und so weiter.

Das folgende Bild zeigt die Projektierung der Positionsfelder in den Bildregeln:



	Name	Bild	Positionierungsfeld	Bedingung
1	Plantsection_Title	Plantscreen		HmiApplication.Name
2	Plantsection_Stat...	Plantscreen		
3	Productionline_Title	Plantscreen		
4	Productionline_Po...	PositionSectionDriveAxis	Gauge	
5	Processing_Unit	PositionSectionPackaging	Gauge	
6	Activate_Btn	Plantscreen	Bar	
7	Stop_Btn	Plantscreen		
8	<neue Regel erstellen>			

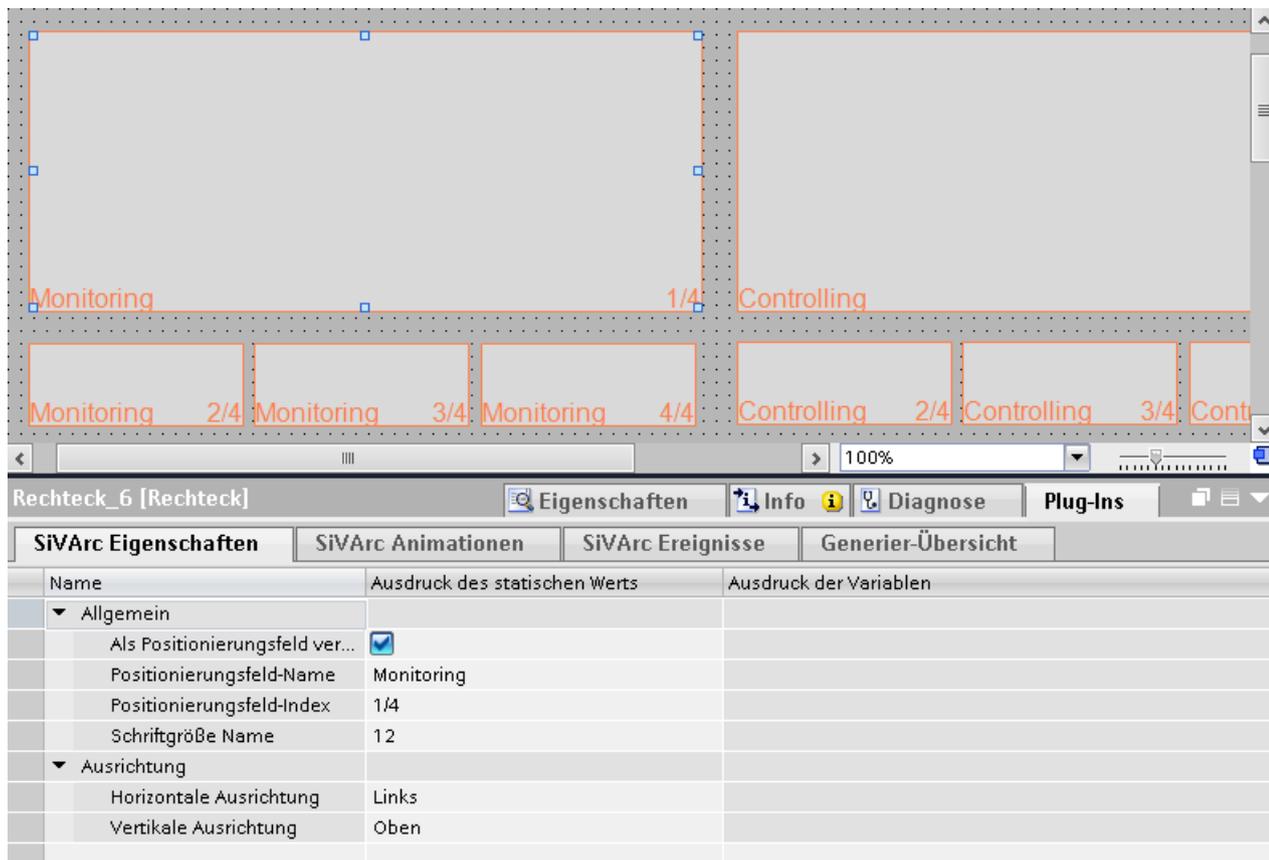
Ebenenzuordnung

Wenn Sie einer Kopiervorlage eine feste Ebene vorgegeben haben und beim Generieren ein eigenes Positionierungsschema verwendet haben, wird das HMI-Objekt in die im Positionierungsschema vorgegebene Ebene generiert.

Aufbau

Ein benutzerdefiniertes Positionierungsschema besteht aus einem Bild, das Positionierungsfelder für generierte Anzeige- und Bedienobjekte enthält. Das Positionierungsschema ordnen Sie einer Generiervorlage zu und erzeugen so ein Prozessbild.

Indem Sie Positionierungsfeldern gleiche Namen geben, fassen Sie solche Positionierungsfelder zu einer logischen Einheit zusammen. Innerhalb einer logischen Einheit werden Positionierungsfelder in der Reihenfolge des Index befüllt.



Nachträgliche Änderungen

Wenn Sie die Position eines generierten Anzeige- und Bedienobjekts manuell ändern, bleibt diese Änderung bei der nächsten Generierung erhalten. Das gilt auch, wenn die Position über ein eigenes Positionierungsschema definiert wurde. Auch wenn Sie das eigene Positionierungsschema ändern, bleibt die manuell eingerichtete Position nach einer weiteren Generierung erhalten.

Siehe auch

Übersicht zur Positionierung von generierten Objekten (Seite 59)

Überlaufmechanismen (Seite 75)

Benutzerdefiniertes Positionierungsschema anlegen (Seite 81)

Beispiel: Einsatz eines dynamischen Layouts (Seite 88)

Beispiel: Einsatz eines kombinierten Layouts (Seite 91)

Beispiel: Einsatz generierter Bildnavigation (Seite 92)

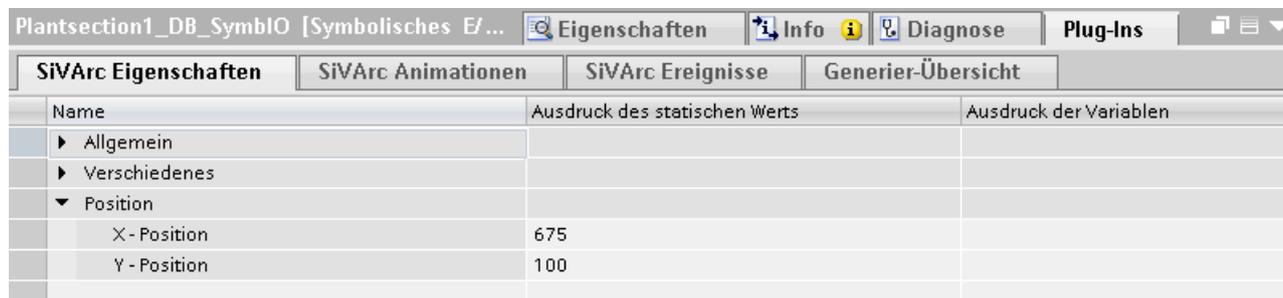
Bildlegende (Seite 279)

6.1.1.3 Feste Positionierung des generierten Objekts

Wenn Objekte immer an der gleichen Position im Bild verankert werden sollen, zum Beispiel Standardobjekte, wählen Sie eine feste Position.

Die feste Positionierung hängt von der Bildauflösung ab. In einem HMI-Gerät mit hoher Auflösung wird das Anzeige- und Bedienungsobjekt weiter oben und links als in einem HMI-Gerät gleicher Größe mit niedrigerer Auflösung dargestellt.

Legen Sie die Koordinaten des Objekts einzeln und unabhängig vom Positionierschema in den SiVArc-Eigenschaften einer Generiervorlage für ein Anzeige- und Bedienungsobjekt fest.



Plantsection1_DB_SymbIO [Symbolisches E/...				
SiVArc Eigenschaften		SiVArc Animationen	SiVArc Ereignisse	Generier-Übersicht
Name	Ausdruck des statischen Werts		Ausdruck der Variablen	
▶ Allgemein				
▶ Verschiedenes				
▼ Position				
X - Position	675			
Y - Position	100			

Hinweis

Unveränderbare, feste Positionierung von Bildobjekten

Bei Bildobjekten mit fester Positionierung wird bei der nächsten Generierung eine manuelle Änderung der Position ignoriert.

Ausdruckseditor für X- und Y-Positionseigenschaften projektieren

SiVArc unterstützt die Projektierung der Position statischer und dynamischer Bildobjekte mit Ausdrücken, die in Integerwerte oder direkte Integerwerte in SiVArc "Plug-ins" Editor > "Position" umgewandelt werden. Sie projektieren die X- und Y-Position von Bildobjekten in der Eigenschaft "Position" mit SiVArc-Ausdrücken und fügen das Bildobjekt in der Kopiervorlage hinzu. Sie definieren die Bildregel, die aus dem unter der Kopiervorlage platzierten Objekt und dem Bild, das generiert werden soll, besteht. Während der Generierung in SiVArc muss der für die X- und Y-Position definierte Ausdruck in Integerwerte umgewandelt werden und generiert die entsprechenden Bildobjekte an dieser Stelle.

Sie können auch dynamische Variablenwerte für X und Y durch "Variablen drucken" konfigurieren. Während der Generierung in SiVArc muss der in Spalte "Variablen drucken" definierte Ausdruck in einen Variablennamen umgewandelt werden und generiert die entsprechenden Bildobjekte. Weitere Informationen über den Generierungsvorgang finden Sie im AbschnittÜbersicht zur Positionierung von generierten Objekten (Seite 59).

Siehe auch

Übersicht zur Positionierung von generierten Objekten (Seite 59)

Grundlagen zur Generierung (Seite 205)

6.1.1.4 Freie Positionierung

Übersicht

Im generierten Bild ist ein Raster hinterlegt, nach dem die Bildobjekte beim Generieren angeordnet werden. Dieses Raster ist konfigurierbar.

Beim ersten Generieren werden die Objekte im Raster auf das Bild generiert. Anschließend ordnen Sie die generierten Objekte individuell an. Die neue Anordnung bleibt bei jedem weiteren Generierlauf erhalten.

Diese Methode bietet Ihnen folgende Vorteile:

- Im Vorfeld sind keine größeren Planungen zum Layout erforderlich.
- Nach jeder Generierung können Sie das Layout weiter anpassen und mehr Festlegungen treffen.
- Das Layout entwickelt sich mit dem SiVArc-Projekt.

Für kleinere Einzel- oder Entwicklungsprojekte ist dieses Vorgehen gut geeignet. Wenn das Projekt größer wird, erhöht sich der Nachbearbeitungsaufwand.

Aufbau und Befüllung des Positionierungsschemas

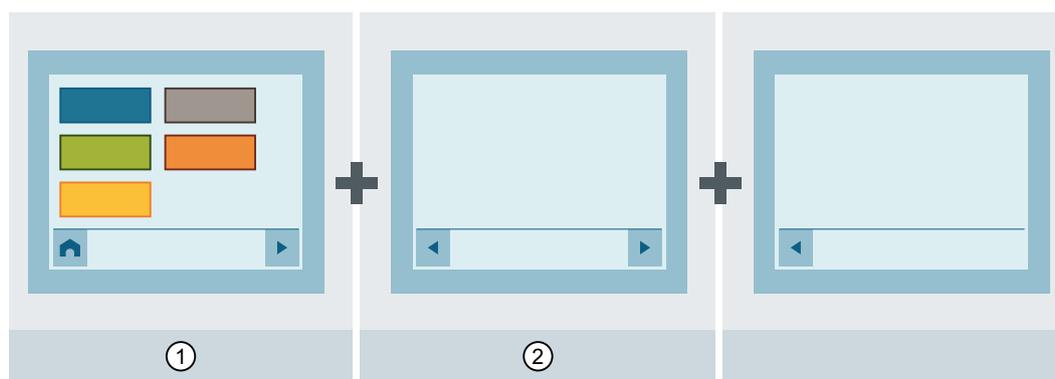
Sie konfigurieren das Positionierungsschema der Objekte in den SiVArc-Eigenschaften des Bildes:

Plantsection1 [Bild]		
Eigenschaften		
Info		
Diagnose		
Plug-Ins		
SiVArc Eigenschaften	SiVArc Animationen	SiVArc Ereignisse
Generier-Übersicht		
Name	Ausdruck des statischen Werts	Ausdruck der Variablen
▶ Allgemein		
▶ Bild als Inhalt des Bildfen...		
▼ Positionierungsschema		
X - Position	150	
Y - Position	150	
Zeilenabstand	200	
Spaltenabstand	600	
▶ Darstellung		

Nach der erstmaligen Generierung werden die HMI-Objekte abhängig vom Positionierungsschema positioniert. Das Positionierungsschema bezieht sich auf die Startposition des ersten Objekts und die Abstände in x- und y-Position.

Wenn den Überlaufbildern keine Bildobjekte zugeordnet wurden, werden die Bildobjekte nach dem erstmaligen Generieren der Visualisierung standardmäßig im Basisbild angeordnet.

Die folgende Abbildung zeigt schematisch die standardmäßige Anordnung der Bildobjekte im Basisbild.



- ① Die generierten Bildobjekte werden spaltenweise pro Bild von oben nach unten und von links nach rechts positioniert. Dabei haben die Bildobjekte immer den gleichen Abstand zueinander.
- ② Wenn für ein Bild Überlaufbilder generiert werden, fügt SiVArc automatisch Navigationsschaltflächen mit projizierten Bildwechslern ein.

Siehe auch

Visualisierung generieren (Seite 215)

Übersicht zur Positionierung von generierten Objekten (Seite 59)

Überlaufmechanismen (Seite 75)

Beispiel: Einsatz eines Layouts mit freier Positionierung (Seite 85)

6.1.1.5 Tiefenstaffelung

Tiefenstaffelung in Bildebenen

In der SiVArc-Kopiervorlage richten Sie über die Ebenenhierarchie die Tiefenstaffelung der zu generierenden Objekte ein. Diese Einstellung wird bei der Generierung beibehalten.

Hinweis

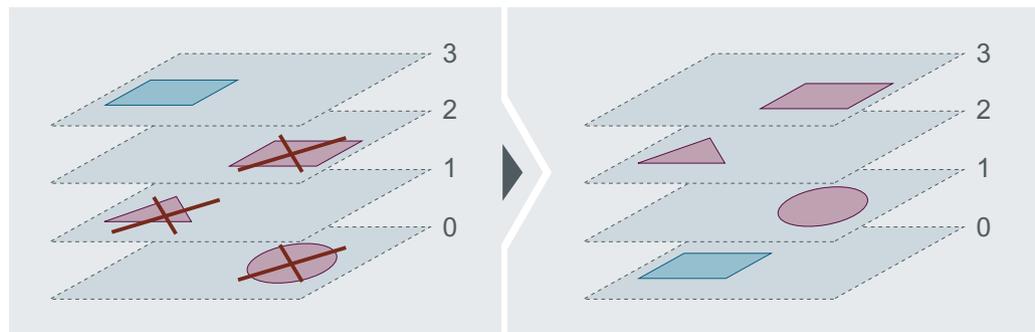
Ebene ändern

Wenn Sie im generierten Bild die Ebenenzuordnung von Objekten ändern, bleibt diese Zuordnung bei einer nachfolgenden Generierung erhalten.

Tiefenstaffelung innerhalb einer Bildebene

Innerhalb der gleichen Ebene im generierten Bild gilt Folgendes:

- Wenn Sie generierte Objekte löschen und dann manuell Objekte einfügen, werden bei der nächsten Generierung die Objekte auch in der Tiefenstaffelung über den manuell eingefügten Objekten generiert.

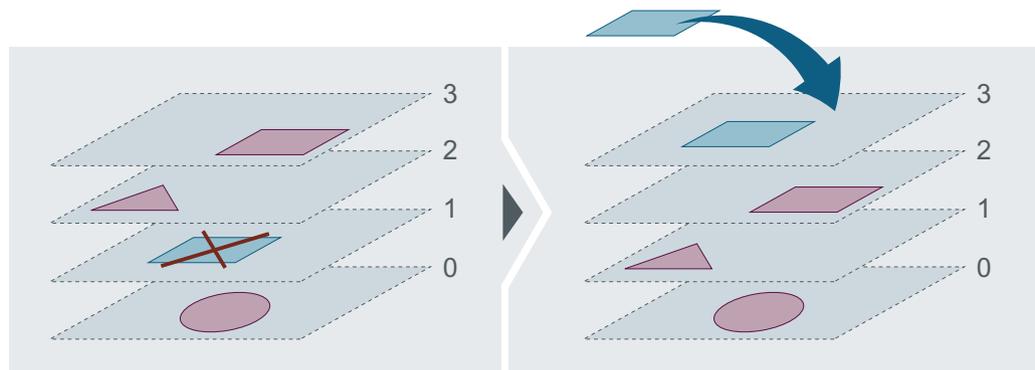


Generiertes Anzeige- und Bedienobjekt



Manuell in das generierte Bild eingefügtes Anzeige- und Bedienobjekt

- Wenn Sie im generierten Bild ein manuell eingefügtes Objekt in einer bestimmten Tiefe einordnen und anschließend löschen, ist die vorherige Einordnung nicht mehr SiVArc-relevant. Bei der nächsten Generierung werden die Bildobjekte an unterster Stelle in der Ebene eingeordnet. Wenn Sie das gelöschte Objekt wieder manuell einfügen, liegt es an oberster Stelle.



Generiertes Anzeige- und Bedienobjekt



Manuell in das generierte Bild eingefügtes Anzeige- und Bedienobjekt

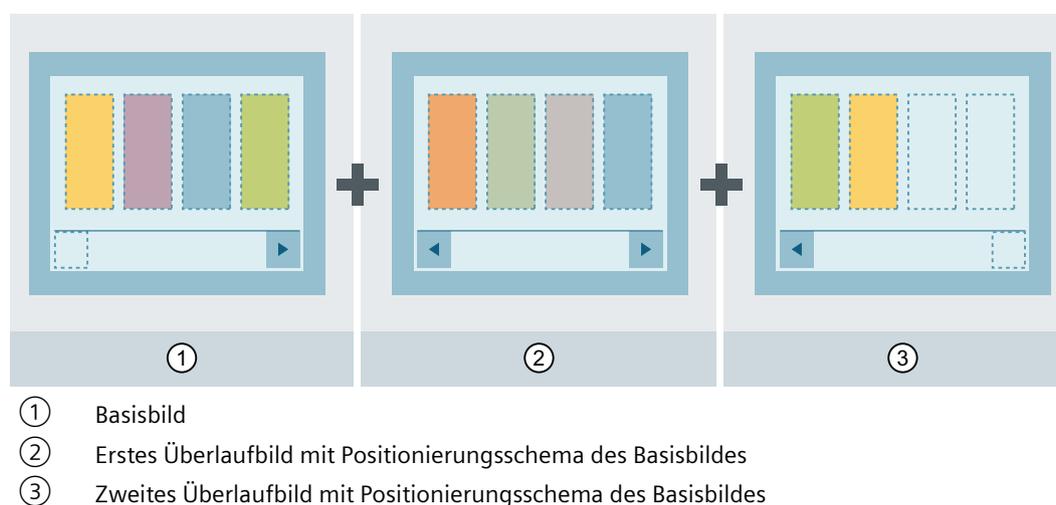
6.1.2 Überlaufmechanismen

Definition

Überlaufbilder sind Bilder, die generiert werden, wenn der Platz auf einem Bild für die Anzahl der generierten Bildobjekte nicht ausreicht. Abhängig vom verwendeten Positionierungsschema werden Überlaufbilder unterschiedlich generiert. Die Überlaufbilder werden für jede Instanz einer Generiervorlage generiert.

Überlaufbilder auf Basis eines definierten Positionierungsschemas

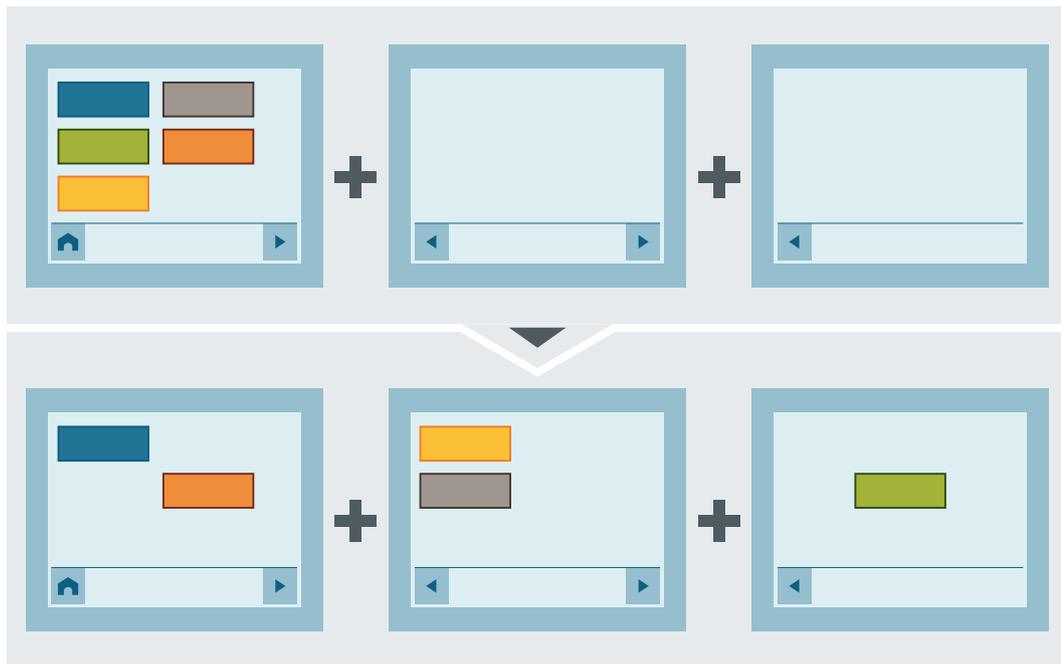
Wenn die projektierten Positionierungsfelder für alle generierten Anzeige- und Bedienobjekte nicht ausreichen, werden Überlaufbilder auf Basis des Positionierungsschemas erzeugt.



Überlaufbilder manuell befüllen

Wenn Sie kein eigenes Positionierungsschema verwenden und in den SiVArc-Eigenschaften der Generiervorlage des Bildes unter "Allgemein" die Anzahl der Überlaufbilder als Dezimalzahl angeben, werden die Bildobjekte nur im Basisbild angeordnet. Um das Generieren von Überlaufbildern einzuschränken, formulieren Sie unter "Anzahl Überlaufbilder" eine Bedingung. Deaktivieren Sie die Option "Anzahl Überlaufbilder als Bitmaske auswerten".

Nach der ersten Generierung verschieben Sie die Bildobjekte auf die gewünschten Positionen in den Überlaufbildern.



Bei jeder weiteren Generierung werden die geänderten Positionen der Bildobjekte beibehalten.

Hinweis

Generierte Anzeige- und Bedienobjekte in ein Überlaufbild kopieren

Wenn Sie die Anzahl Überlaufbilder als Dezimalzahl angeben, beachten Sie Folgendes:

Wenn Sie mit SiVArC generierte Objekte von einem Basisbild in ein Überlaufbild manuell kopieren, bleibt diese Änderung bei einer erneuten Generierung erhalten. Die Kopie wird zusammen mit dem HMI-Objekt in dem Basisbild wie ein generiertes Objekt von SiVArC behandelt und hat Bezug zu SiVArC.

Voraussetzung: Der Name der Kopie muss mit dem Namen des Originals übereinstimmen.

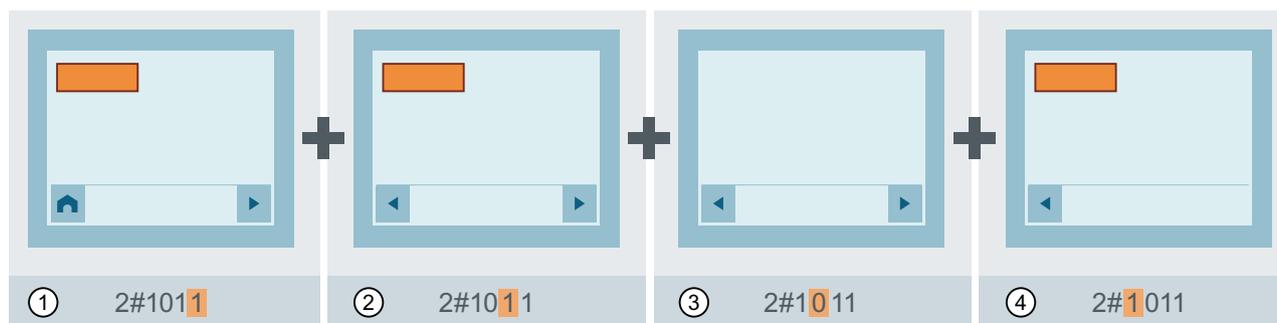
Mit Bitmaske gesteuerte Überlaufbilder

Sie definieren die Bitmaske am Programmbaustein oder in der Generiervorlage des Bildes. Dazu verwenden Sie einen statischen Wert oder eine Variable.

Wenn Sie die Verteilung der Überlaufbilder als Bitmaske angeben, werden die Bildobjekte im Basisbild und in Überlaufbildern angeordnet.

Mit der Bitmaske legen Sie Folgendes fest:

- Anzahl Überlaufbilder
Die Anzahl der Bitpositionen in der Bitmaske definiert die Anzahl der Überlaufbilder. Die erste Position in der Bitmaske entspricht dabei dem Bild der Kopiervorlage. Die zweite Position entspricht dem ersten Überlaufbild, die dritte Position dem zweiten Überlaufbild usw. Die Bitmaske ist begrenzt auf 31 Überlaufbilder. Wenn Sie die Bitmaske 2#0 verwenden, wird kein Überlaufbild erzeugt.
- Überlaufbilder mit Bildobjekten
Wenn auf ein Überlaufbild das Bildobjekt der verwendeten Bildregel generiert werden soll, setzen Sie das entsprechende Bit in der Bitmaske auf 1.
Beispiel: Sie verwenden die Bitmaske 2#1011. Bei der Generierung werden drei Überlaufbilder erzeugt. Das Bildobjekt der verwendeten Bildregel wird wie folgt generiert:



- ① Basisbild mit generiertem Bildobjekt
- ② Erstes Überlaufbild
- ③ Zweites Überlaufbild
- ④ Drittes Überlaufbild

Sobald ein Positionierungsfeld im Bild projiziert ist, werden im Bild enthaltene Projektierungen zu Überlaufbildern beim Generieren nicht mehr ausgewertet.

Hinweis

Objekte in ein Überlaufbild kopieren oder verschieben

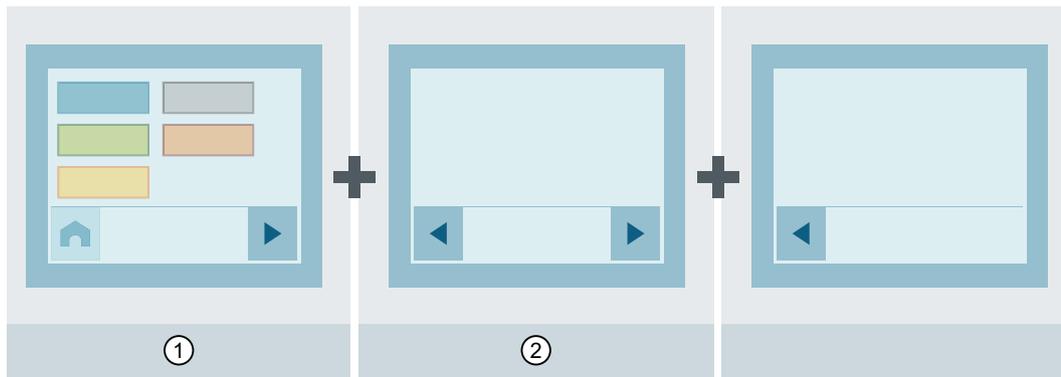
Wenn Sie Überlaufbilder mit einer Bitmaske projektieren, beachten Sie Folgendes:

Wenn Sie generierte Anzeige- und Bedienobjekte in ein Überlauf- oder Basisbild kopieren oder verschieben, werden diese Objekte bei einer erneuten Generierung wie manuell angelegte Objekte behandelt. Das Anzeige- und Bedienobjekt wird bei der nächsten Generierung neu angelegt.

Navigationsschaltflächen

Wenn SiVArC Überlaufbilder generiert, werden Navigationsschaltflächen für Bildwechsel zum vorherigen und nächsten Bild generiert.

Wenn Sie die generierten Anzeige- und Bedienobjekte frei positionieren, wird das ursprüngliche Bild der Generiervorlage als Basisbild generiert. Über eine Navigationsschaltfläche ist das Basisbild mit dem ersten Überlaufbild verbunden.



① Generiertes Bild der Kopiervorlage (Basisbild)

② Erstes Überlaufbild mit automatisch generierten Navigationsflächen mit projizierten Bildwechsell

Um auf Navigationsflächen zu verzichten, deaktivieren Sie die Auswahl "Navigationsflächen" in der Generiervorlage des Bildes.

Hinweis

Für die Navigationsflächen können Sie Kopiervorlagen in der Bibliothek ablegen.

Nähere Informationen finden Sie im Abschnitt "Generiervorlagen in SiVArc (Seite 93)"

Pop-up-Bilder

Für Pop-up-Bilder werden keine Überlaufbilder generiert. Wenn mehr Anzeige- und Bedienobjekte generiert werden, als positioniert werden können, wird eine Fehlermeldung ausgegeben. Anzeige- und Bedienobjekte, welche keinen Platz mehr auf dem Pop-up-Bild haben, werden nicht generiert.

Siehe auch

Freie Positionierung (Seite 72)

Beispiel: Einsatz von Überlaufmechanismen (Seite 86)

Überlaufbilder ohne Bildobjekte projizieren (Seite 84)

6.1.3 Unterstützte Geräte

Übersicht

SiVArc kann mit folgenden Geräten verwendet werden:

- PLCs
 - SIMATIC S7-1200
 - SIMATIC S7-1500
 - SIMATIC S7-1500 Software-Controller
 - ET 200SP CPU
- Geräte-Proxies
Geräte-Proxies werden nur für die Generierung externer Variablen verwendet.
- HMI-Geräte
 - HMI-Geräte mit WinCC RT Professional
 - HMI-Geräte mit WinCC RT Advanced
 - Comfort Panels
 - Mobile Panels der 2. Generation
 - Basic Panels
 - WinCC Comfort Unified/ Unified Scada RT

6.1.4 Planen eines Layouts

Einleitung

Welche Positionierungsmethode Sie wählen, ist abhängig von der Größe eines Automatisierungsprojekts und den Anforderungen im Unternehmen.

Folgende Projektbeispiele erläutern das Wirkprinzip und den Zweck der einzelnen Positionierungsmethoden.

Standardisierte Großprojekte

In diesem Fall sind eigene Positionierungsschemata empfehlenswert. Sie erreichen damit eine genaue Platzierung und nutzen den Anzeigebereich Ihrer verschiedenen Bediengeräte optimal aus. Jedem Bediengerät können Sie ein eigenes Positionierungsschema zuordnen. Um einen Wechsel in standardisierter Form umzusetzen, können Sie z. B. vertikale und horizontale Ausrichtungen von Bediengeräten als eigene Positionierungsschemata einrichten.

Auch wenn viele Objekte generiert werden, bleibt die Platzierung doch handhabbar, weil jedes Objekt die Platzierungsinformation über die Bildregel erhält. Wenn der Anzeigebereich nicht ausreicht, werden automatisch Überlaufbilder mit dem gleichen Layout erzeugt. Durch die Modularisierung der Platzierungsbereiche über kombinierte Positionierungsschemata erreichen Sie eine hohe Wiederverwendbarkeit Ihrer Vorlagen.

Kleinere, individuelle Projekte

Mit der freien Positionierung haben Sie einen geringen Planungsaufwand für das Layout. In einem kleinen Projekt sind manuelle Änderungen am Layout weniger aufwändig.

Wenn Sie frei positionieren, haben Sie auch die Möglichkeit, mit wenig Generiervorlagen für Bilder zu arbeiten und Ihre Visualisierung über Überlaufbilder zu erweitern. Überlaufbilder bestücken Sie auch über die freie Positionierung individuell mit Objekten.

Wenn Sie ein besonderes Konzept zur Verteilung der Objekte auf die Überlaufbilder wünschen, steuern Sie die Überlaufbilder über eine Bitmaske. Da Sie die Bitmaske in der Steuerung hinterlegen können, erhalten Sie einen direkten Bezug von der Steuerung zum Bildaufbau. Dieser Bezug unterstützt Sie z. B. bei der Fehlersuche.

Entwicklungs- und Testphase

In diesem Fall hat die freie Positionierung mehr Vorteile. Sie bleiben lange flexibel in der Positionierung. Da Sie die generierten Objekte individuell verschieben, können Sie im Projekt viele unterschiedliche Bediengeräte verwenden. Überlaufbilder stellen die Übersichtlichkeit Ihrer Anlagenbilder sicher. Die Bildnavigation der Überlaufbilder wird automatisch generiert.

Ein Bediengerätewechsel erfordert eine erneute erste Generierung mit manueller Positionierung.

Standard-Anlagenbilder

Die feste Positionierung ist abhängig vom Bediengerät. Daher eignet sich die feste Positionierung nur in Projekten mit vielen gleichen Bediengeräten oder für den oberen linken Bereich eines Bediengeräts. Da die feste Positionierung von einem Positionierungsschema ignoriert wird, verwenden Sie die feste Positionierung nur in Kombination mit der freien Positionierung.

Wenn in einem Projekt keine Änderungen oder Bediengerätewechsel erwartet werden, ist die feste Positionierung geeignet.

Planung der Bildebenen

Die Positionierung innerhalb der Bildebenen steuern Sie für die generierten Anzeige- und Bedienobjekte wie in WinCC über die WinCC-Eigenschaften der Generiervorlagen. Nur innerhalb einer Ebene ist die von SiVArc definierte Tiefenstaffelung relevant.

Siehe auch

Übersicht zur Positionierung von generierten Objekten (Seite 59)

6.1.5 Benutzerdefiniertes Positionierungsschema anlegen

Voraussetzung

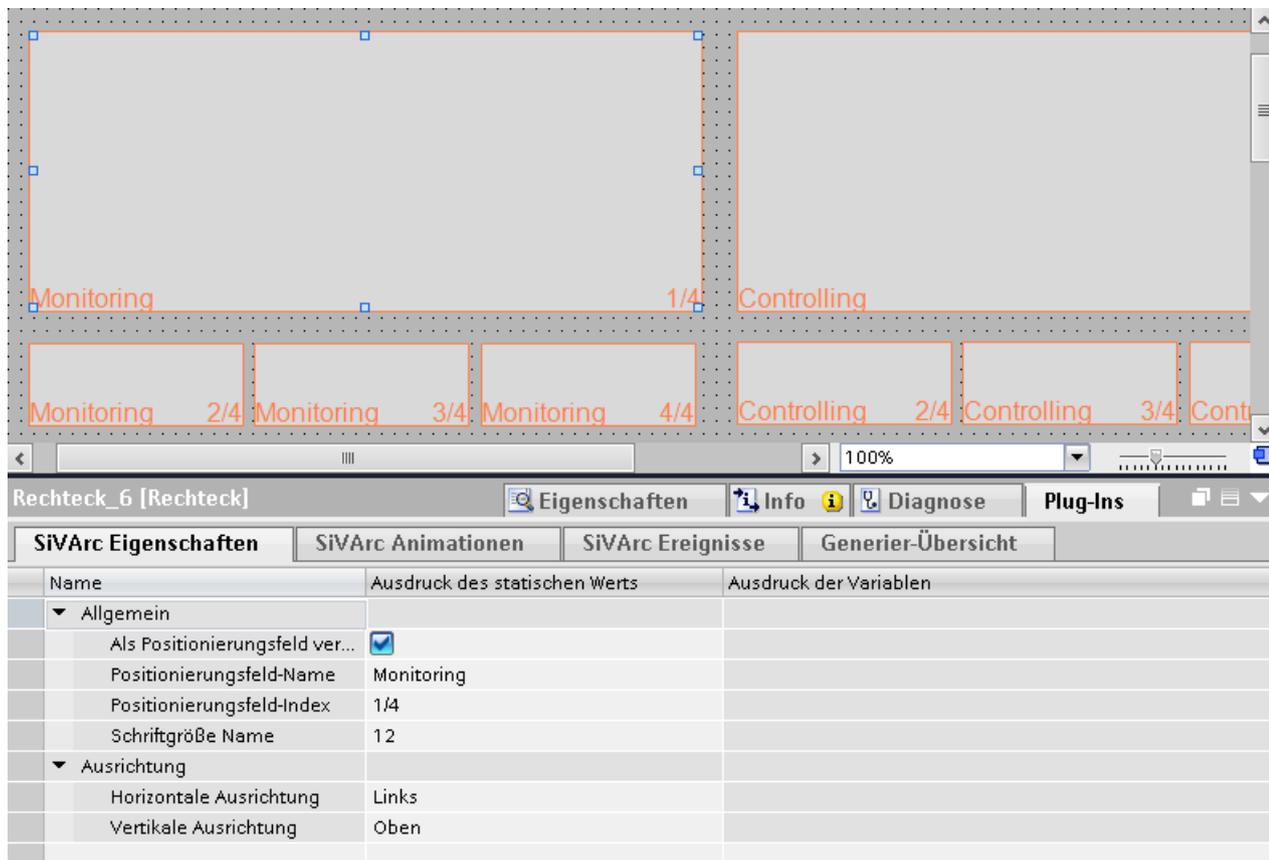
- Der Editor "Bilder" ist geöffnet.
- Das Bild "Overview" ist angelegt.

Vorgehen

Um ein Positionierungsschema anzulegen, gehen Sie folgendermaßen vor:

1. Fügen Sie aus dem Werkzeugfenster aus der Gruppe "Basisobjekte" mehrere Rechtecke in das Bild ein.
Achten Sie darauf, dass die Rechtecke für die generierten Anzeige- und Bedienobjekte groß genug sind. Sonst überlappen sich die Anzeige- und Bedienobjekte im generierten Bild.
2. Wählen Sie in den SiVArc-Eigenschaften der Rechtecke "SiVArc-Eigenschaften > Allgemein > "Als Positionierungsfeld verwenden".

3. Definieren Sie Bereiche im Bild.
 - Geben Sie den Positionierungsfeldern, die zu einer logischen Einheit gehören, den gleichen Namen, z. B. "Monitoring" und "Controlling".
 - Ändern Sie dazu unter "Allgemein > Positionierungsfeld-Name" den Namen des Positionierungsfeldes.
 - Richten Sie unter "Allgemein > Schriftgröße Name" die Schriftgröße ein.
 - Legen Sie in den WinCC-Eigenschaften jeweils für ein Positionierungsfeld unter "Eigenschaften > Eigenschaften > Gestaltung" die Rahmen- und Schriftfarbe der Positionierungsfelder fest.



4. Ändern Sie bei Bedarf unter "Allgemein > Positionierungsfeld-Index" die Reihenfolge der Befüllung der Felder.
Die Positionierungsfelder werden mit Namen und Index dargestellt.
5. Legen Sie das Bild "Overview" in der Bibliothek als Kopiervorlage ab.
6. Löschen Sie das Bild "Overview" in der Projektnavigation.

Indexreihenfolge

Die Indexvergabe folgt der zeitlichen Reihenfolge, in der Sie die Indizes editieren. Wenn Sie z. B. ein Positionierungsfeld nachträglich einer anderen logischen Einheit zuordnen, erhält das Feld die letzte Indexnummer dieser Einheit, unabhängig von der Anordnung im Bild.

Die Indexreihenfolge passt sich nach jeder Änderung automatisch neu an.

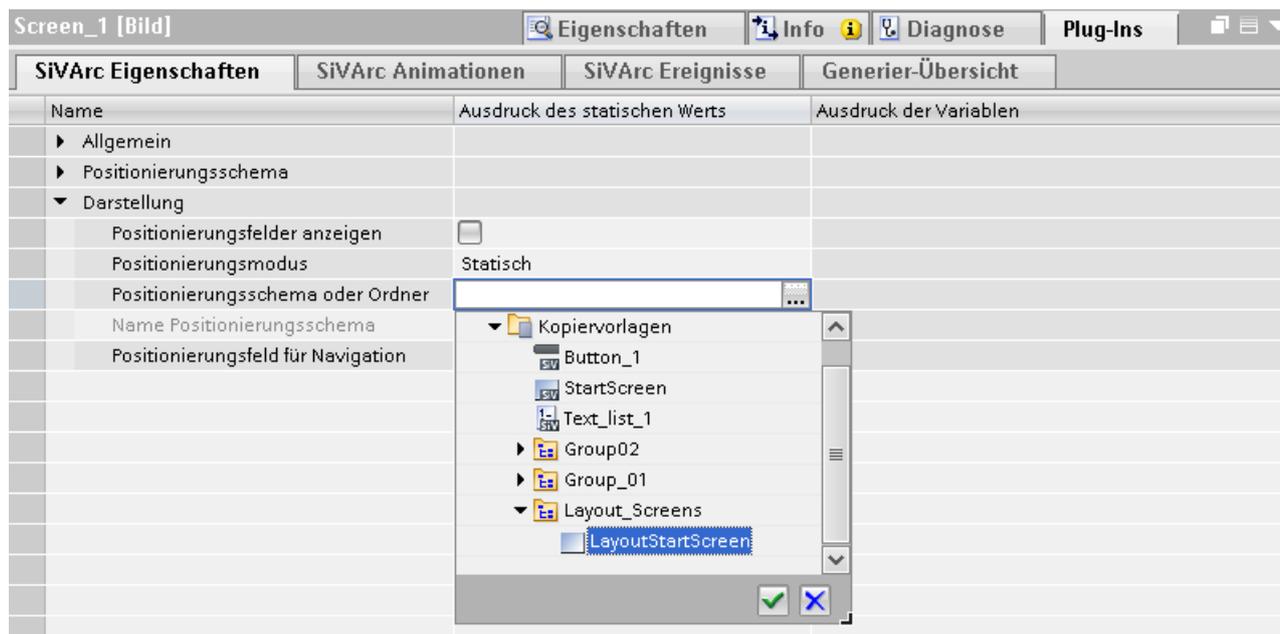
Ergebnis

Das Positionierungsschema ist angelegt. Sie können das Schema einer Generiervorlage eines Bildes hinterlegen oder das Schema als Bild in einer Bildregel verwenden.

Positionierungsschema einer Generiervorlage fest zuordnen

Um ein Positionierungsschema in einer Generiervorlage zu verwenden, gehen Sie folgendermaßen vor:

1. Erzeugen Sie aus der Generiervorlage, in die Sie das neue Positionierungsschema hinterlegen wollen, ein neues Bild.
2. Wählen Sie unter "Positionierungsmodus" die Option "Statisch".
3. Wählen Sie unter "Positionierungsschema oder Ordner" das gewünschte Positionierungsschema.



4. Löschen Sie die Generiervorlage in der Bibliothek.
5. Legen Sie das bearbeitete Bild als Generiervorlage in der Bibliothek ab.
6. Löschen Sie das Bild in der Projektnavigation.

Wenn Sie die Generiervorlage in einer Bildregel verwenden, geben Sie in der Bildregel auch das Positionierungsfeld an. SiVArc generiert das Bildobjekt in dieses Positionierungsfeld in das Feld

mit dem Index 1. Das nächste generierte Objekt wird in das Feld mit dem Index 2 generiert und so weiter.

Hinweis

Ebenenzuordnung

Wenn Sie einer Kopiervorlage eine feste Ebene vorgegeben haben und beim Generieren ein eigenes Positionierungsschema verwendet haben, wird das HMI-Objekt in die im Positionierungsschema vorgegebene Ebene generiert.

Positionierungsfelder im generierten Bild anzeigen

Um im generierten Bild die Positionierungsfelder anzuzeigen, aktivieren Sie in den SiVArc-Eigenschaften des Bildes "SiVArc-Eigenschaften > Darstellung > Positionierungsfelder anzeigen".

Siehe auch

Positionierung nach definierten Schemata (Seite 61)

6.1.6 Überlaufbilder ohne Bildobjekte projektieren

Einleitung

Sie können Überlaufbilder im SiVArc-Projekt mit und ohne Bildobjekte einrichten. Wenn Sie Überlaufbilder ohne Bildobjekte projektieren, verschieben Sie die generierten Anzeige- und Bedienobjekte nach dem ersten Generieren auf die Überlaufbilder. Diese Position ist dann für alle nachfolgenden Generierungen gesetzt.

Vorgehen

Um Überlaufbilder ohne Bildobjekte zu projektieren, gehen Sie folgendermaßen vor:

1. Geben Sie im Inspektorfenster unter "Plug-Ins > SiVArc-Eigenschaften > Allgemein" zu "Anzahl Überlaufbilder" die gewünschte Anzahl an Bildern ein.

Hinweis

Die Überlaufbilder werden für jede Instanz dieser Kopiervorlage generiert.

Um das Generieren von Überlaufbildern einzuschränken, formulieren Sie unter "Anzahl Überlaufbilder" eine Bedingung.

2. Deaktivieren Sie die Option "Anzahl Überlaufbilder als Bitmaske auswerten".
3. Aktivieren Sie bei Bedarf die Generierung von Navigationsschaltflächen.
4. Definieren Sie eine oder mehrere Bildregeln.
5. Starten Sie die Generierung.

SiVArC generiert alle Bildobjekte in das generierte Basisbild. Nach dem erstmaligen Generieren verschieben Sie die generierten Bildobjekte auf die gewünschten Positionen in die Überlaufbilder. Bei jeder weiteren Generierung werden die geänderten Positionen der Bildobjekte beibehalten.

Hinweis**Generierte Anzeige- und Bedienobjekte in ein Überlaufbild kopieren**

Wenn Sie die Anzahl Überlaufbilder als Dezimalzahl angeben, beachten Sie Folgendes:

Wenn Sie mit SiVArC generierte Objekte von einem Basisbild in ein Überlaufbild manuell kopieren, bleibt diese Änderung bei einer erneuten Generierung erhalten. Die Kopie wird zusammen mit dem HMI-Objekt in dem Basisbild wie ein generiertes Objekt von SiVArC behandelt und hat Bezug zu SiVArC.

Voraussetzung: Der Name der Kopie muss mit dem Namen des Originals übereinstimmen.

Überlaufbilder und Positionierungsfelder

Sobald ein Positionierungsfeld im Bild projiziert ist, haben die Eigenschaften "Anzahl Überlaufbilder" und "Anzahl Überlaufbilder als Bitmaske auswerten" keine Funktion mehr.

Siehe auch

Beispiel: Einsatz von Überlaufmechanismen (Seite 86)

Überlaufmechanismen (Seite 75)

6.1.7 Beispiel: Einsatz eines Layouts mit freier Positionierung**Beispielsszenario**

Ein Leiterplattenhersteller modernisiert seine Anlage und entscheidet sich unter anderem für eine neue Bediengerätegeneration mit größerem Anzeigebereich.

Anforderung

Das Ingenieurbüro, welches die Anpassungen in der Visualisierung übernimmt, entwickelt für die neuen Bediengeräte passende Positionierungsschemata für die von SiVArC generierten Anzeige- und Bedienobjekte.

Umsetzungskonzept

Um die Positionierungsschemata für SiVArC zu entwickeln, generierte der Visualisierungingenieur das bestehende Projekt erstmalig ohne Positionierungsschema.

Anschließend ordnet der Projektteur die generierten Anzeige- und Bedienobjekte auf den größeren Bildern der Bediengeräte neu an. Bei jeder weiteren Generierung bleibt die Anordnung erhalten und wird vom Projektteur weiter angepasst.

Erst nachdem die Anordnung der generierten Anzeige- und Bedienobjekte optimiert ist, erstellt der Projektteur auf dieser Basis die Positionierungsschemata für die neuen Bediengeräte.

Siehe auch

Freie Positionierung (Seite 72)

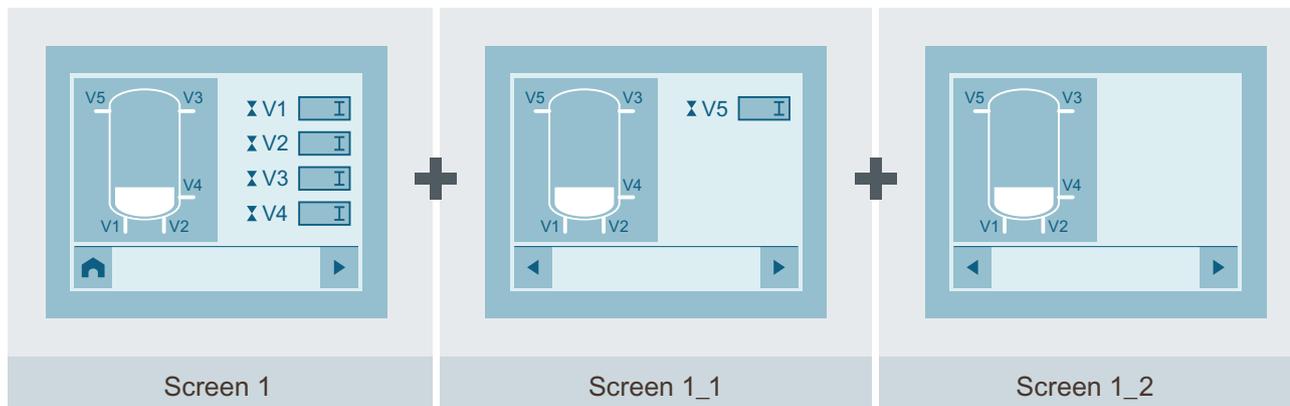
6.1.8 Beispiel: Einsatz von Überlaufmechanismen

Beispielsszenario Überlaufbild

In einer bestehenden Anlage wird ein Ventil ergänzt und mit einer Statusausgabe visualisiert. Die Anzeige des dazugehörigen Bediengeräts ist zu klein für weitere Anzeige- und Bedienobjekte.

Umsetzungskonzept

Das zusätzliche Anzeigeobjekt wird ohne weitere Projektierungen auf das erste Überlaufbild generiert.



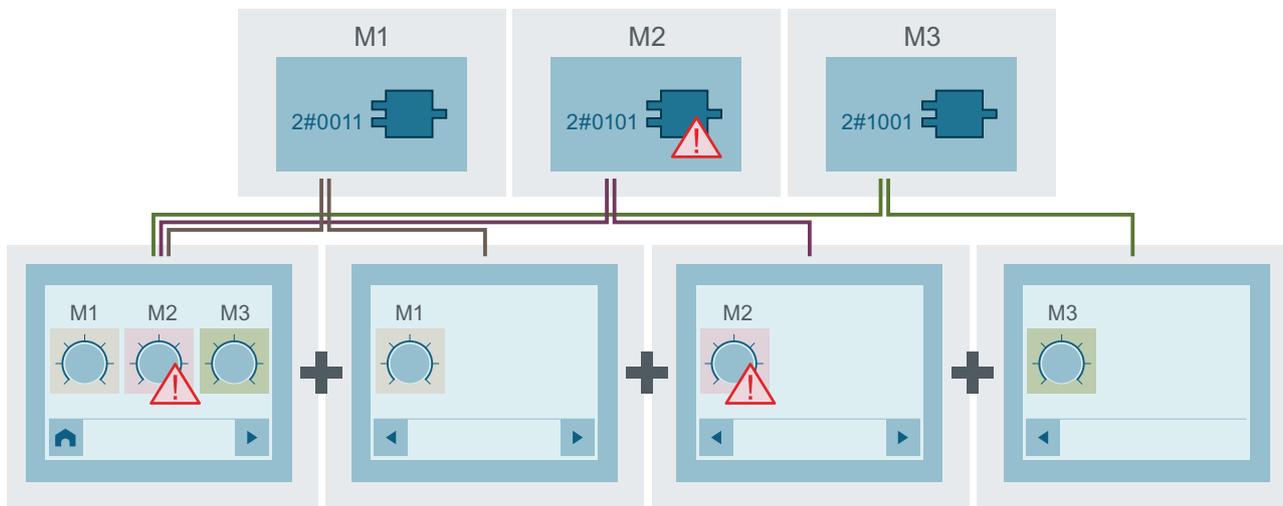
Beispielsszenario Überlaufbild mit Bitmaske

Wenn in einer Anlage ein Fehler auftritt, soll nach Eintritt einer Meldung die Navigation zur Fehlerquelle sichergestellt sein.

Umsetzungskonzept

In der Steuerung ist an einem Eingang eine Bitmaske für die Generierung von Überlaufbildern hinterlegt. Die Verteilung der generierten Objekte auf die generierten Überlaufbilder wird von der Bitmaske festgelegt. Für jedes mit 1 belegte Bit wird das Objekt einer Bildregel generiert.

Wenn an einem Bit ein Fehler auftritt, dann ist er im Prozessbild im Überlaufbild der gleichen Bitposition enthalten.



Anzahl Überlaufbilder als Bitmaske auswerten

Um Überlaufbilder mit Bitmaske zu projektieren, gehen Sie folgendermaßen vor:

1. Tragen Sie im Inspektorfenster unter "Plug-Ins > SiVArc-Eigenschaften > Allgemein" zu "Anzahl Überlaufbilder" die gewünschte Bitmaske ein, z. B. 11 (2#1011).
oder
Wählen Sie im Inspektorfenster unter "Plug-Ins > SiVArc-Eigenschaften > Allgemein" zu "Anzahl Überlaufbilder" den Bausteineingang, an dem die Bitmaske für Überlaufbilder gesetzt ist, z. B. Block.Parameters("OVERFLOW_PIC").Value.
2. Aktivieren Sie die Option "Anzahl Überlaufbilder als Bitmaske auswerten".
3. Aktivieren Sie bei Bedarf die Generierung von Navigationsschaltflächen.
4. Definieren Sie eine oder mehrere Bildregeln.
5. Starten Sie die Generierung.

Wenn Sie eine Bitmaske als Zahl eingetragen haben, werden in diesem Beispiel beim Generieren drei Überlaufbilder erzeugt. In das erste und das dritte Überlaufbild wurde das Bildobjekt der verwendeten Bildregel generiert und in das Basisbild.

Wenn Sie den Bausteineingang gewählt haben, wird der Wert am Parameter verarbeitet. Wenn kein gültiger Wert gesetzt ist, wird nur am Basisbild das Bildobjekt der verwendeten Bildregel generiert und eine Fehlermeldung ausgegeben.

Bitmaske für Überlaufbilder als Variable projektieren

Um Überlaufbilder mit einer Bitmaske zu projektieren, die in einer Variablen gespeichert ist, gehen Sie folgendermaßen vor:

1. Tragen Sie im Inspektorfenster unter "Plug-Ins > SiVArC-Eigenschaften > Allgemein" zu "Anzahl Überlaufbilder" den Namen der SiVArC-Variablen ein, die für die Bitmaske für Überlaufbilder definiert wurde, z. B. "SiVArCVariable".
2. Aktivieren Sie die Option "Anzahl Überlaufbilder als Bitmaske auswerten".
3. Aktivieren Sie bei Bedarf die Generierung von Navigationsschaltflächen.
4. Definieren Sie eine oder mehrere Bildregeln.
5. Starten Sie die Generierung.

Beim Generieren wird der aktuelle Wert der gewählten Variablen verarbeitet. Wenn keine Variable angelegt ist, generiert SiVArC das Bildobjekt der verwendeten Bildregel in das Basisbild.

Siehe auch

Überlaufmechanismen (Seite 75)

Überlaufbilder ohne Bildobjekte projektieren (Seite 84)

6.1.9 Beispiel: Einsatz eines dynamischen Layouts

Beispielsszenario

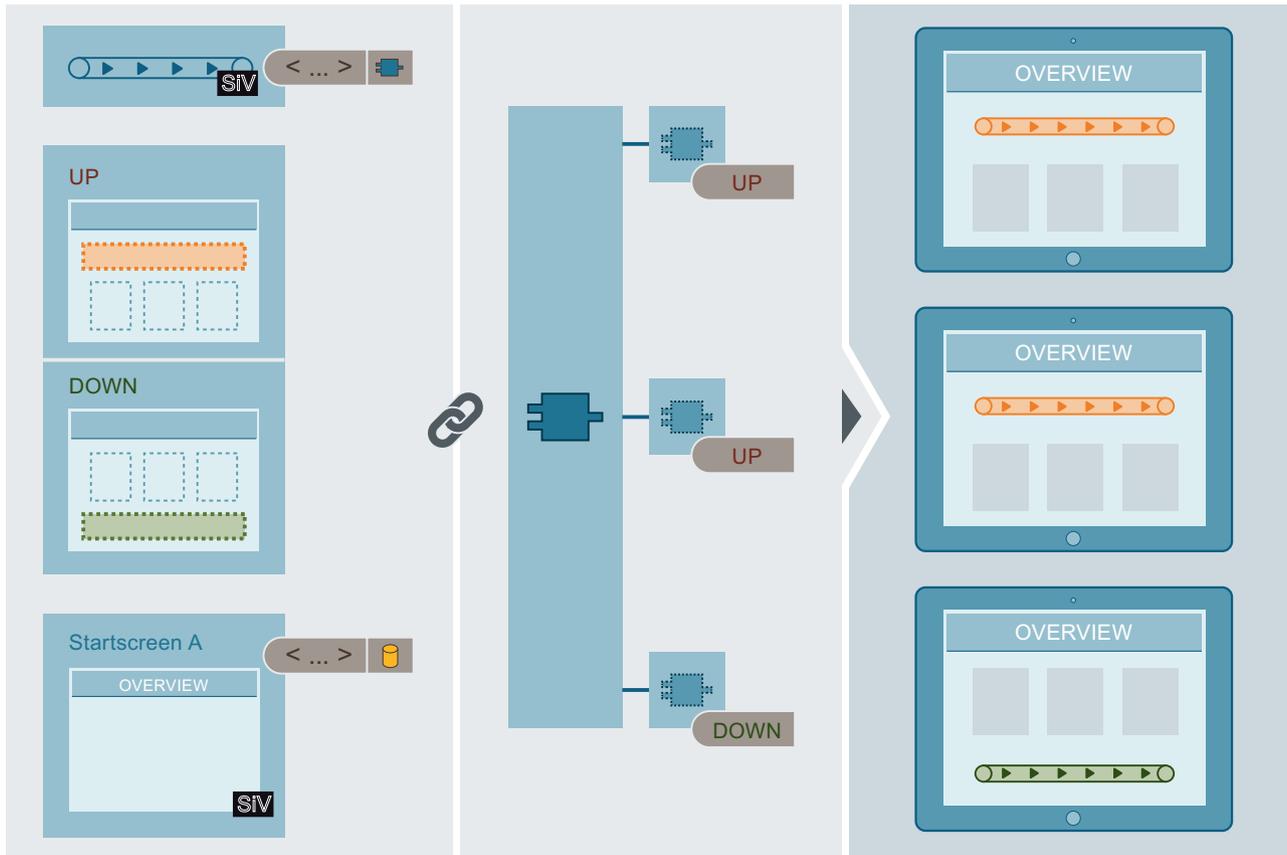
Zwei Förderbänder erfordern eine unterschiedliche Bildanordnung, haben aber die gleichen Bedienobjekte.

Anforderung

Die Anzahl Bildregeln soll reduziert werden.

Umsetzungskonzept

Zwei Positionierungsschemata werden erstellt und in einem Bibliotheksordner abgelegt. Am Netzwerk ist eine SiVArc-Variable definiert. Die SiVArc-Variable steuert, wo das Band im Bild angeordnet werden soll. Beim Auswerten der Bildregel wird das passende Positionierungsschema gewählt. Das eine Band wird im oberen Bildbereich angeordnet, das andere Band wird im unteren Bildbereich dargestellt.

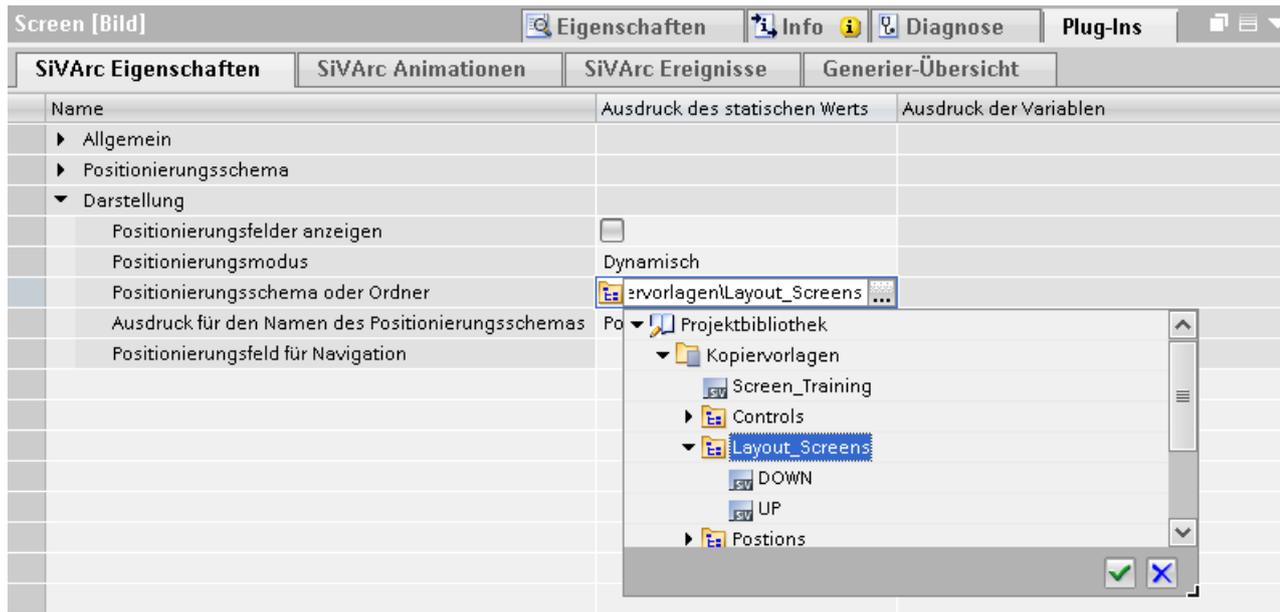


Dynamisches Positionierungsschema einrichten

Wenn Sie einem Bild abhängig von bestimmten Bedingungen ein Positionierungsschema zuordnen wollen, ordnen Sie der Generiervorlage einen Ordner mit Positionierungsschemata zu. Dann vergeben Sie einen SiVArc-Ausdruck, der den Namen eines im gewählten Ordner enthaltenen Positionierungsschemas zurückgibt.

1. Erstellen Sie in einem Bibliotheksordner mehrere Positionierungsschemata.
2. Benennen Sie den Ordner z. B. mit "Layout_Screens".
3. Öffnen Sie die Generiervorlage des Bildes, der Sie ein dynamisches Positionierungsschema zuordnen wollen.
4. Wählen Sie in den SiVArc-Eigenschaften unter "Positionierungsmodus" den Modus "Dynamisch".

5. Wählen Sie unter "Positionierungsschema oder Ordner" den Ordner "Layout_Screens".



6. Projektieren Sie unter "Ausdruck für den Namen des Positionierungsschemas" einen SiVArC-Ausdruck, der den Namen eines im Ordner enthaltenen Positionierungsschemas zurückgibt. Sie können z. B. im Anwenderprogramm eine SiVArC-Variablen definieren und als Bedingung verwenden. Sie ordnen dann der Variablen den Namen des für diesen Programmbaustein gewünschten Positionierungsschemas zu.
7. Legen Sie das bearbeitete Bild als Generiervorlage in der Bibliothek ab.
8. Löschen Sie das Bild in der Projektnavigation.

Hinweis

Wenn Sie für das Generieren von mehreren Bildelementen eines Bildes den Positionierungsmodus "Dynamisch" wählen, werden nicht alle dynamisch zugeordneten Positionierungsfelder dargestellt.

Auch wenn Sie in den SiVArC-Eigenschaften des Bildes "SiVArC -Eigenschaften > Darstellung > Positionierungsfelder anzeigen" aktivieren, wird nur das Positionierungsfeld für das zuerst generierte Bildelement dargestellt.

Siehe auch

Positionierung nach definierten Schemata (Seite 61)

Bildlegende (Seite 279)

6.1.10 Beispiel: Einsatz eines kombinierten Layouts

Beispielszenario

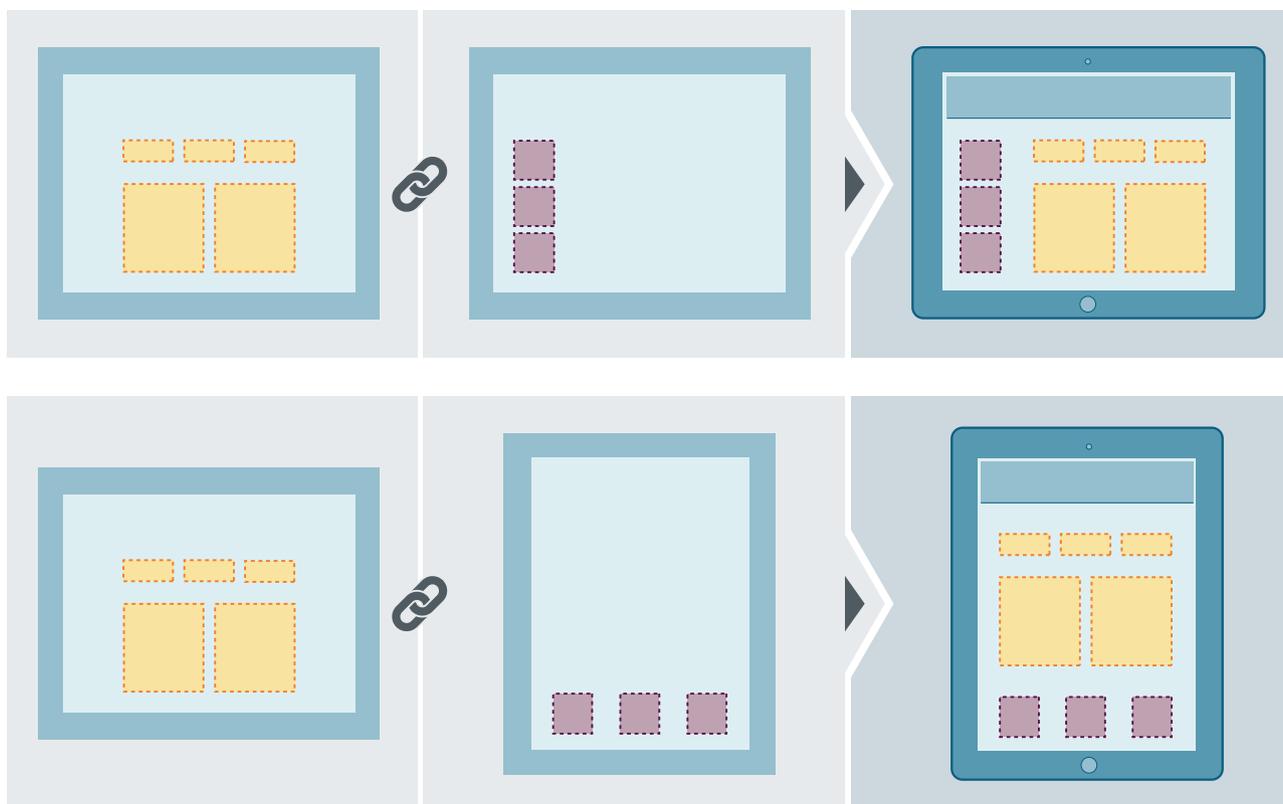
Die Übersichtsbilder für die Fertigungsstrecke für Platzieren, Löten und Verpacken sind ähnlich aufgebaut und unterscheiden sich dadurch, dass zwei unterschiedliche Bediengeräte eingesetzt werden (Querformat und Hochformat). Das bedingt eine unterschiedliche Aufteilung des Bildschirms.

Anforderung

Das Layout der Anlagenbilder soll mit einem hohen Wiederverwendungsgrad der bestehenden Positionierungsschemata umgesetzt werden.

Umsetzungskonzept

Das Positionierungsschema für die mittleren Bildobjekte der Fertigungsstrecken wird mit anderen Positionierungsschemata kombiniert. Auf diese Weise werden die Anzeige- und Bedienobjekte auf dem vertikal ausgerichteten Bediengerät unten angeordnet. Auf dem horizontal ausgerichteten Gerät werden die Objekte seitlich angeordnet.



Siehe auch

Positionierung nach definierten Schemata (Seite 61)

6.1.11 Beispiel: Einsatz generierter Bildnavigation

Beispielsszenario

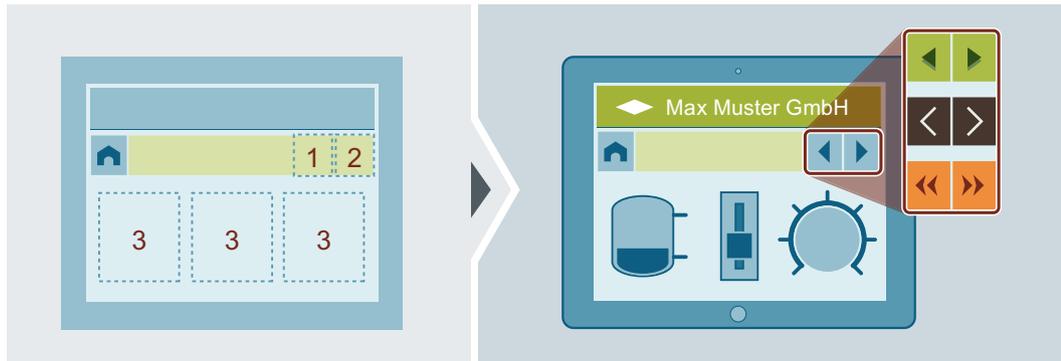
In einem vorhandenen Visualisierungsprojekt weicht die Bildnavigation ab von den neuen Vorgaben zum Corporate Design und zur Standardisierung im Unternehmen.

Anforderung

Die Projektierung der Bildnavigation soll effizienter und ansprechender umgesetzt werden.

Umsetzungskonzept

Das beauftragte Ingenieurbüro beschließt, die neue Bildnavigation über benutzerdefinierte Positionierungsschemata umzusetzen. Auf diese Weise werden Bedienobjekte des Corporate Designs nach firmenspezifischer Anordnung verwendet.

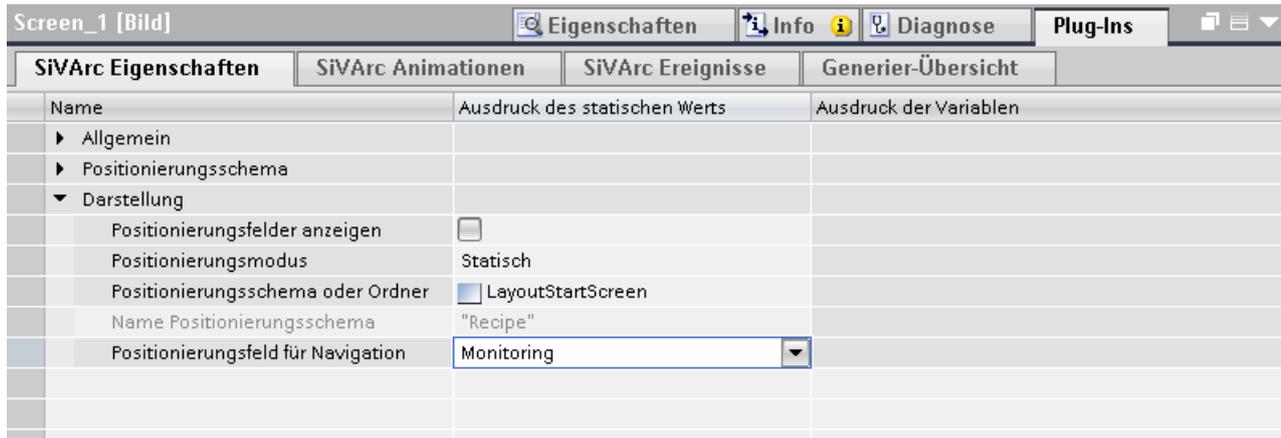


In den Überlaufbildern wird die automatische Bildnavigation aktiviert. Definierte Anlagenteile werden so in einem einzigen Bild mit Überlaufbildern dargestellt. Überlaufbilder vermindern die Anzahl der Generiervorlagen für Bilder und vereinheitlichen die Darstellung. Außerdem ist dadurch sichergestellt, dass alle Objekte in einem sichtbaren Bereich generiert werden.

Für die automatische Bildnavigation werden auch firmenspezifische Bedienobjekte hinterlegt.

Positionierungsfeld für Navigationsschaltflächen verwenden

Um die Bildnavigation über benutzerdefinierte Positionierungsschemata umzusetzen, wählen Sie in den SiVArc-Eigenschaften des Bildes unter "SiVArc-Eigenschaften > Darstellung > Positionierungsfeld für Navigation" das Positionierungsfeld für Navigationsschaltflächen, z. B. "Monitoring".



Wenn beim Generieren mit dieser Generiervorlage Überlaufbilder entstehen, werden im Positionierungsfeld "Monitoring" die Navigationsschaltflächen platziert.

Automatische Bildnavigation in Überlaufbilder aktivieren

Um die automatische Bildnavigation in Überlaufbildern zu aktivieren, aktivieren Sie in der Generiervorlage des Bildes die Option "Navigationsschaltflächen".

Siehe auch

Positionierung nach definierten Schemata (Seite 61)

6.2 Erstellen von Generiervorlagen

6.2.1 Generiervorlagen in SiVArc

Definition

Generiervorlagen sind HMI-Objekte aus der Bibliothek, die nicht nur mit festgelegten definierten WinCC-Eigenschaften, sondern auch mit SiVArc-Eigenschaften projiziert werden. Eine SiVArc-Eigenschaft ist eine Objekteigenschaft, die zunächst als Variable/Ausdruck zugeordnet wird. Entsprechend dem SiVArc-Mechanismus werden die SiVArc-Eigenschaften erst bei der Generierung mit Texten wie dem Objektnamen, Beschriftungen oder einer Variablenbezeichnung gefüllt.

Funktionsprinzip

SiVArC-Eigenschaften können statisch oder dynamisch sein. Auf der Registerkarte "SiVArC-Eigenschaften" projizieren Sie die Eigenschaften einer Generiervorlage.

The screenshot shows the SIMATIC Manager software interface. The main workspace displays a grid with two text objects labeled "Text" and a larger gray rectangular area. The bottom panel shows the "Schaltfläche_2 [Schaltfläche]" context menu with the "Eigenschaften" (Properties) option selected. The "SiVArC Eigenschaften" (SiVArC Properties) dialog box is open, showing a table with columns for "Name", "Ausdruck des statischen Werts" (Static value expression), and "Ausdruck der Variablen" (Variable expression).

Name	Ausdruck des statischen Werts	Ausdruck der Variablen
▼ Allgemein		
Text AUS	Block.DB.SymbolicName	
▼ Verschiedenes		
Name	<input type="text"/>	
Ebene		
Text Tooltip		
▼ Position		
X - Position		
Y - Position		

Die Registerkarte enthält drei Spalten:

- Name
In dieser Spalte sind die verfügbaren Eigenschaften aufgeführt.
- Ausdruck für den statischen Wert
In dieser Spalte projektieren Sie eine Eigenschaft mit einem festen Wert oder einem SiVArc-Ausdruck, der eine Zeichenkette oder eine Zahl zurückgibt.
Feste Werte werden beim Generieren der Visualisierung in jede Instanz dieser Kopiervorlage eingetragen. Achten Sie z. B. bei der Eigenschaft "Name" darauf, dass die Eindeutigkeit des Objektnamens auch bei Mehrfachverwendung in einem Bild gewährleistet ist.
- Ausdruck der Variablen
In dieser Spalte projektieren Sie eine Eigenschaft mit einem Variablennamen oder einem SiVArc-Ausdruck, der einen Variablennamen zurückgibt.

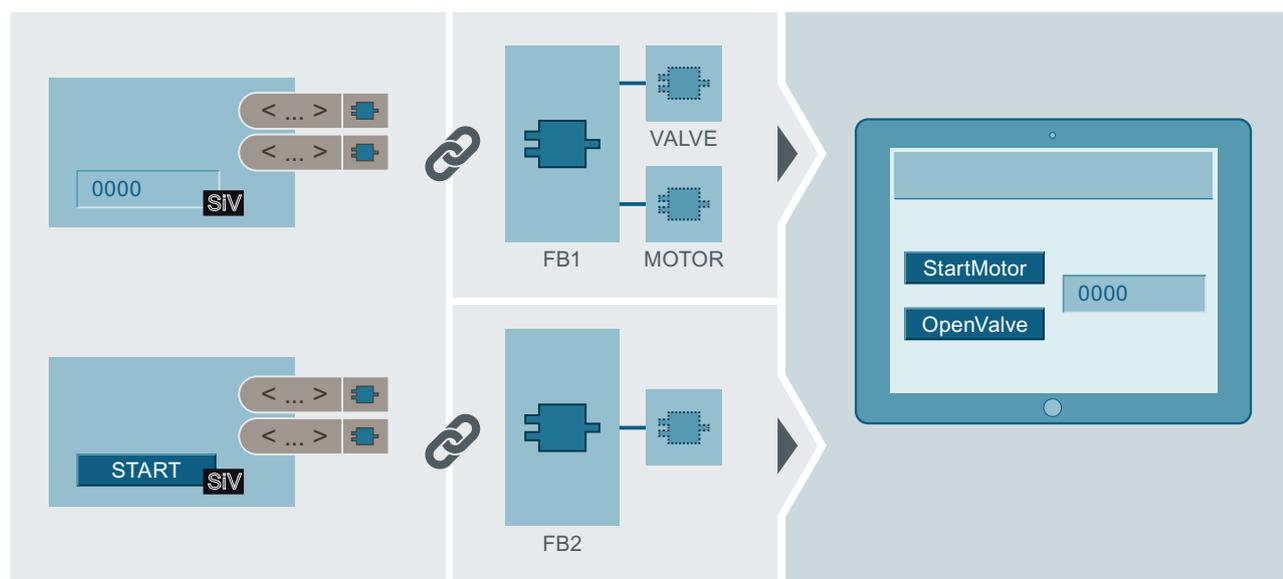
Anschließend legen Sie die projektierte Generiervorlage in der Projektbibliothek ab und verschalten sie später an einen Funktionsbaustein. Dabei erstellen Sie für jede Generiervorlage eine separate Verschaltung. Die SiVArc-Eigenschaften werden bei der Generierung der Visualisierung ausgewertet.

Generiervorlagen für Bilder und Bildobjekte

Pro HMI-Objekt wird eine Generiervorlage generiert.

Die Registerkarte "SiVArc-Eigenschaften" ist nur für von SiVArc unterstützte Objekte verfügbar.

Die folgende Abbildung zeigt schematisch das Generieren von Bildobjekten aus Generiervorlagen, die auf eine Anweisung aus STEP 7 verweisen. Die SiVArc-Eigenschaften werden bei der Generierung der HMI-Objekte ausgewertet. Objekteigenschaften wie "Beschriftung" oder "Name" werden generiert.



SiV	SiVArc-Generiervorlage
	SiVArc-Eigenschaft mit referenzierter Textquelle
	Funktionsbaustein
	Instanzierter Funktionsbaustein

Bildobjekte mit Bildvorlagen generieren

SiVArc unterstützt das Generieren von Bildobjekten aus Bildvorlagen und die Erstellung von Vorlagen. Die Vorlagen können Sie mit dem **Plug-ins**-Editor erstellen. In den Vorlagenbildern können Sie folgende Aktionen ausführen:

- Vollständige Generierung und Zusammenführen von Szenarien wie in normalen Bildern
- Projektieren von Regeln durch Auswahl von Vorlagen als Kopiervorlagen
- Aktualisieren der Bildvorlage ausgehend von der projizierten Vorlageneigenschaft

Hinweis

Unterstützung des Vorlagenbilds auf allen anderen Geräten, außer auf Professional-Geräten.

Typen als Generiervorlage verwenden

Beachten Sie beim Generieren von Bildtypinstanzen und Bildbausteininstanzen Folgendes:

- Wenn Sie für die SiVArc-Generierung Bildtypen verwenden, werden alle Instanzen im Projekt aktualisiert, auch diejenigen, die nicht von SiVArc erstellt wurden.
- Wenn Sie die Verbindung einer generierten Instanz mit einem Bildtyp entfernen und das Bild ändern, wird die Änderung bei der nächsten Generierung trotzdem durch eine neue Instanz des Bildtyps überschrieben.
- Wenn Sie für die SiVArc-Generierung Bildtypen der globalen Bibliothek verwenden, wird der Bildtyp der Projektbibliothek hinzugefügt.
- Bei der Generierung aktualisiert SiVArc den verwendeten Bildtyp auf die aktuellste Typversion, die im Projekt und in den Bibliotheken vorhanden ist.

Hinweis

Bildtypen als Generiervorlage verwenden

Ein Bildtyp wird im Editor "Bildregeln" nur als Bildobjekt verwendet. Der Bildtyp wird daher immer in einem Bildfenster angezeigt.

Generiervorlagen für Textlisten

Bei SiVArc werden mehrsprachige Textlisteneinträge direkt im Anwenderprogramm erstellt. Dies sind z. B. Statustexte für Funktionsbausteine oder Schnittstellenbeschreibungen für Bausteinparameter. Bei der Generierung verschalten Sie die Texte mit den entsprechenden Anzeige- und Bedienobjekten. Hierdurch haben Sie die Möglichkeit, beschreibende Texte für Ihr Projekt einzugeben.

Sie können Textlisteneinträge in mehrsprachigem Format am Baustein ablegen oder sie von der Symboltabelle eines Bausteinparameters ableiten.

- Textlisteneinträge am Baustein
Sie können Textlisteneinträge im Netzwerk oder Programmbaustein erstellen. Um das richtige Netzwerk auszuwählen, klicken Sie im Inspektorfenster in einem beliebigen Bereich des Netzwerks auf einen Programmbaustein. Für die Textlisteneinträge können Sie auch SiVArc-Ausdrücke verwenden.
Über Namensgleichheit sind die Textdefinition und der Textlisteneintrag in der Textlisten-Kopiervorlage miteinander verknüpft.
- Textlisteneinträge am Bausteinparameter
Für einzelne Parameter in der Symboltabelle legen Sie einen Kommentar an, der von SiVArc zum Textlisteneintrag verarbeitet wird.

Hinweis

Textquellen aus STEP 7 verwenden

Innerhalb einer Textliste wird nur eine Textquelle verarbeitet. Verwenden Sie daher für eine Textliste entweder Texte vom Baustein **oder** Texte aus einer Symboltabelle.

Generiervorlagen für automatisch generierte Objekte

Die folgenden Objekte werden automatisch mit SiVArC generiert:

- Bildfenster zur Anzeige eines Bilds im Bild
- Navigationsschaltflächen für Überlaufbilder

Die automatisch generierten Objekte können mithilfe von Generiervorlagen angepasst werden.

Hierzu legen Sie die angepassten Objekte unter "Kopiervorlagen" in der Projektbibliothek ab.

Beachten Sie beim Ablegen der angepassten Objekte die folgenden Vorgaben:

- Die Generiervorlage für das Bildfenster muss unter dem Namen "DefaultScreenWindowControl" abgelegt werden.
- Die Generiervorlagen für die Navigationsschaltflächen müssen in einer Bibliothek unter den Namen "DefaultScreenWindowControl" und "PrevButton" abgelegt werden. Sie können diese Schaltflächen individuell projektieren.

Wenn Sie die Generiervorlagen nicht anpassen, werden bei einer Generierung die Standardvorlagen aus der Werkzeugpalette verwendet.

Generiervorlagen für Positionierungsschemas

Um anzugeben, an welcher Position im generierten Bild ein Bildobjekt platziert werden soll, geben Sie eine Positionierungsfeldgruppe an, die Sie in den Bildregeln dem Bildobjekt zuordnen.

Generiervorlagen für Bilder und Bildobjekte unter Verwendung von Kopierregeln

In SiVArC können Sie HMI-Bilder auf der Basis von Vorlagen mithilfe von Kopierregeln generieren. Bei der Generierung von Bildvorlagen mithilfe von Kopierregeln wird die Möglichkeit genutzt, dass einzelne oder mehrere Bildvorlagen mehreren in einem Projekt projektieren Bediengeräten zugleich zugeordnet werden können. Die Bildvorlagen werden automatisch mithilfe von Kopierregeln aus der Projektbibliothek oder der globalen Bibliothek in Bediengeräte kopiert. Im Rahmen der SiVArC-Generierung werden die Bildvorlagen für Bediengeräte unter "Bildverwaltung" im Ordner "Vorlagen" bereitgestellt.

Generierung von HMI-Bildern unter Verwendung der Eigenschaft für Bildvorlagen

In SiVArC können Sie HMI-Bilder mit Bildvorlagen unter Verwendung der "SiVArC-Eigenschaften" unter "Plug-Ins" projektieren. Hierbei muss Folgendes gewährleistet sein:

- Der im Inspektorfenster des HMI-Bilds eingegebene Name "Bildvorlage" muss dem Namen der Bildvorlage ähnlich sein.
- Der für "Bildvorlage" aufgelöste Ausdruck muss mit dem Namen der vorhandenen Bildvorlage im Projekt übereinstimmen.

Nach Projektierung des HMI-Bilds mit der Bildvorlage wird das HMI-Bild im Ordner "Kopiervorlagen" unter "Bibliotheken" abgelegt. Eine Bildregel wird erstellt, die auf die vorhandene Bildvorlage zurückgreift, und wird bei der SiVArC-Generierung auf die HMI-Bilder

angewandt. SiVArc generiert die HMI-Bilder mit demselben Namen wie die projektierte Bildvorlage.

Hinweis

Bildvorlagen, die Kopiervorlagen und eine SiVArc-Eigenschaft verwenden, werden von den Geräten der SIMATIC-Panel- und der RT-Advanced-Reihe unterstützt.

Siehe auch

Anforderungen an eine Generiervorlage (Seite 112)

Beispiel: Generierung einer Bildvorlage mithilfe der Kopierregel (Seite 134)

Beispiel: Generierung eines HMI-Bilds mithilfe von Vorlageneigenschaften (Seite 135)

6.2.2 Unterstützte HMI-Objekte

Mit Steuerungsdaten generierbare HMI-Objekte

Abhängig davon, für welches Bediengerät HMI-Objekte zu generieren sind, generiert SiVArc folgende HMI-Objekte:

HMI-Objekt	Basic Panels	Comfort Panels / Mobile Panels / 2nd Generation RT Advanced	RT Professional	WinC Comfort Unified/ Unified SCADA RT
Externe Variable ¹	x	x	x	---
Folgende Kopiervorlagen in einer Bibliothek:				
Balken	x	x	x	x
Bild ¹	x	x	x	x
Bildfenster	---	---	x	---
Bildvorlage	---	x	---	---
EA-Feld	x	x	x	x
Grafisches EA-Feld	x	x	x	---
GRAPH-Übersicht	---	x	x	---
Kurvenanzeige	x	x	---	---
f(x)-Kurvenanzeige	---	x	x	---
f(t)-Kurvenanzeige	---	---	x	---
PLC-Codeanzeige	---	x	x	---
Pop-up-Bild ¹	---	x	---	---
ProDiag-Übersicht	---	x	x	---
Rundschaftfläche	---	---	x	---
Schalter	---	x	---	x
Schaltfläche	x	x	x	x
Schieberegler	---	x	x	x

HMI-Objekt	Basic Panels	Comfort Panels / Mobile Panels / 2nd Generation RT Advanced	RT Professional	WinC Comfort Unified/ Unified SCADA RT
Symbolisches EA-Feld	x	x	x	x
Textfeld	x	x	x	x
Textlisten	x	x	x	x
Zeigerinstrument	---	x	x	x
Folgende Typen in einer Bibliothek:				
Bild als Bildfenster ¹	---	---	x	---
Meldungen	x	x	x	---
Bildbausteine	---	x	x	x
Ereignisse	x	x	x	x
Animation	x	x	x	-

¹: Strukturierte Ablage möglich

Hinweis

Je nach verwendetem Bediengerät können die Eigenschaften der von SiVArc unterstützten HMI-Objekte variieren.

Ohne Steuerungsdaten generierbare HMI-Objekte

SiVArc generiert oder instanziiert folgende Objekte aus Typen oder Kopiervorlagen einer Bibliothek:

HMI-Objekt	Basic Panels	Comfort Panels/ Mobile Panels 2nd Generation RT Advanced	RT Professional	WinCC Unified SCADA RT / Uni- fied Comfort Pa- nel
Bild	x	x	x	
Variablen				x
Interne Variable	x	x	x	x
Variablen-tabelle	x	x	x	x
Skripte				
C-Skript	---	---	x	
VB-Skript	x	x	x	x
Textliste	x	x	x	x

Eigenschaften mit geräteabhängigen Maximalwerten

Die Maximalwerte für einzelne Eigenschaften sind beim Generieren der Visualisierung für die folgenden Bediengeräte begrenzt:

Eigenschaft	Basic Panels	Comfort Panels	Mobile Panels / 2nd Generation	WinCC Unified Comfort Panel	WinCC Unified SCADA RT
Text Off (Länge)	320	500	500	1000	1000
ToolTip (Länge)	320	1000	1000	320	320
Textfeld (Text)	320	32767	32767	30000	30000
Textlisteneintrag (Text)	320	320	320	320	255

6.2.3 Quellen für Texte

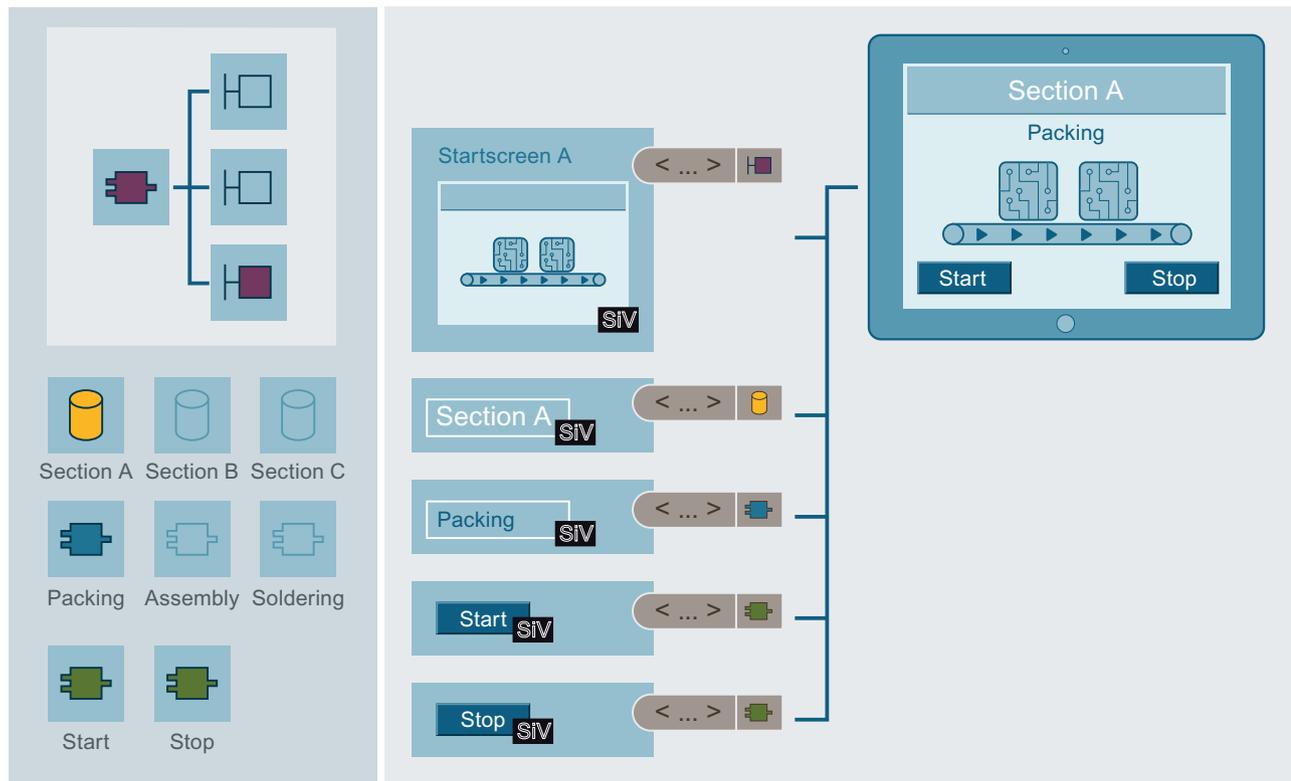
6.2.3.1 Übersicht zu Textquellen im SiVArc-Projekt

Definition

Mit SiVArc können Sie auf Texte von STEP 7 und anderen TIA Portal-Editoren für die Visualisierung zugreifen. Im Gegensatz zur üblichen WinCC-Konfiguration werden diese Texte vom Steuerungsprogrammierer erstellt. Diese Texte können mehrfach in der Visualisierung mit SiVArc verwendet werden.

Die folgende Abbildung zeigt, wie ein Bild mit SiVArc aufgebaut wird:

- In STEP 7 stehen verschiedene Textquellen zur Verfügung, zum Beispiel Netzwerke, Datenbausteine oder Funktionsbausteine.
- Ein Bild besteht aus mehreren Generiervorlagen. Die SiVArc-Eigenschaften der Generiervorlagen greifen auf Textquellen zu.
- Während der Generierung verarbeitet SiVArc die referenzierten Textquellen und füllt die SiVArc-Eigenschaften der HMI-Objekte aus.



-  SiVArc-Generiervorlage
-  SiVArc-Eigenschaft mit referenzierter Textquelle
-  Main [OB1]
-  Netzwerk im Anwenderprogramm
-  Datenbaustein
-  Prozessbaustein
-  Standardbaustein

Vorteile

SiVArc generiert verschiedene Texte, je nachdem welcher Funktionsbaustein mit einer Generiervorlage verknüpft wird. Eine Generiervorlage kann daher an verschiedenen Stellen verwendet werden. Das WinCC-Projekt kann auf einfache Weise angepasst werden. Die Konsistenz der Texte wird vom Anwenderprogramm in die Visualisierung übertragen.

String-Funktionen

Um die Wiederverwendbarkeit von Generiervorlagen zu maximieren bzw. Texte für die Anzeige zu optimieren, stellt SiVArc mehrere String-Funktionen bereit, wie "Split", "Contains" oder "Trim".

Textquellen aus STEP 7

Eine SiVArc-Eigenschaft kann sich auf die folgenden Texte von STEP 7 beziehen:

- Netzwerk
 - SiVArc-Texte und SiVArc-Variablen
 - Netzwerktitel
 - Netzwerkkommentar
- Datenbaustein
 - Symbolischer Name
 - Speicherpfad in der Projektnavigation
 - Kommentar
 - Bausteinnummer
 - TagPrefix
 - Typ (IDB, MDB)
- Funktionsbaustein
 - Kommentar
 - Parameterwert
 - Speicherpfad in der Projektnavigation
 - Bausteinnummer
 - Titel
 - Typversion eines Bibliothekstyps
- Main block (OB)
 - Symbolischer Name
 - Symbolische Adresse
 - Version
 - Ordnerpfad
 - Typ
 - Nummer
- Peripheriegerät
 - Artikelnummer
 - Invariantentyp
 - Name

- PLC-Variablen
 - HMI-Präfix
- HMI-Gerät
 - GeräteName/Auflösung/Typ
- HMI-Anwendung
 - Name
 - Typ

Textquellen aus Hardwaredaten

Eine SiVArC-Eigenschaft kann auf die folgenden Eigenschaften eines HMI-Geräts zugreifen:

- Runtime-Software
 - Name
 - Typ
- HMI-Gerät
 - Name
 - Typ

Textquellen aus Bibliotheken

Eine SiVArC-Eigenschaft kann auf die folgenden Eigenschaften von Bibliotheksobjekten zugreifen:

- Name
- Speicherpfad in der Bibliothek

Siehe auch

Einfluss des Anwenderprogramms auf eine Generiervorlage (Seite 121)

SiVArC-Ausdruck (Seite 109)

6.2.3.2 SiVArC-Texte

Einleitung

Mit SiVArC können Sie zur Generierung Ihrer Visualisierung Texte als Textlisteneinträge definieren. Diese Funktionalität ist ins Anwenderprogramm in STEP 7 integriert und steht an jedem Netzwerk und Baustein zur Verfügung.

Um SiVArC-Texte zu definieren, wählen Sie im Inspektorfenster des gewünschten Netzwerktitels oder Bausteintitels die Registerkarte "Plug-Ins".

Definition

SiVArc-Texte sind Textlisteneinträge, die in STEP 7 erstellt werden. Bei der Generierung der Visualisierung werden daraus Textlisteneinträge im HMI generiert. Über den Namen sind Textdefinition und Textlisteneintrag miteinander verknüpft.

Wenn der Programmbaustein in einer Textlistenregel verwendet wird, werden in einer Textliste die SiVArc-Texte als Textlisteneinträge generiert.

The screenshot shows the SIMATIC Manager interface. The main window displays a ladder logic network (Netzwerk 1) with a function block (FB2) named "Controller_01". The function block has two inputs: "Motor" and "Valve", both set to "false". The function block has three outputs: "Motor_1", "Valve_1", and "Tagname". The function block is connected to a data block (%DB2) named "Controller_01_DB_1".

The bottom panel shows the "SiVArc" tab with the "Textdefinitionen" section. The table below shows the text definitions for the network.

Name	Text in der aktuellen Bearbeitungssprache
Netzwerk 1	
Info	
Warning	PID Compact 1: Door is open
<Neuen Text hinzufügen>	

Sie können SiVArc-Texte statisch oder dynamisch festlegen:

- Statisch: Sie vergeben einen Text als Textdefinition. Diesen Text können Sie auch mehrsprachig projektieren.
- Dynamisch: Sie legen einen SiVArc-Ausdruck als Textdefinition fest.

Wenn Sie einen Text und einen SiVArc-Ausdruck angeben, wird der SiVArc-Ausdruck verwendet.

Siehe auch

SiVArc-Variablen (Seite 110)

Einfluss der Mehrsprachigkeit auf eine Generiervorlage (Seite 124)

6.2.4 Unterstützte Objekte im Anwenderprogramm

Programmbausteine

SiVArc unterstützt die folgenden Programmbausteine:

- Funktionsbaustein (FB)
- Funktion (FC)
- Datenbaustein (DB)
 - Globale DBs
 - Instanz-DBs

FBs und FCs werden im Anwenderprogramm aufgerufen. In Bildregeln werden nur FBs und FCs verwendet. Sie können FBs und FCs auch als Kopiervorlagen und Typen aus der Bibliothek verwenden.

Sprachen für Programmbausteine

SiVArc unterstützt bei Programmbausteinen die folgenden Programmiersprachen:

- AWL
- FUP
- KOP
- SCL

Technologieobjekte

SiVArc unterstützt die folgenden Technologieobjekte:

- PID-Regelung:
 - Kompakt-PID
 - Basis-PID-Funktionen

Siehe auch

Unterstützte Datentypen für PLC-Variablen (Seite 275)

6.2.5 SiVArc-Scripting

Definition

SiVArc Scripting ist eine Skript-Sprache, die von VBS abgeleitet ist und die ausschließlich im TIA Portal für die SiVArc-Option verwendet wird.

SiVArc Scripting kann auf Textquellen im TIA Portal zugreifen. Dabei verknüpft SiVArc Scripting das Anwenderprogramm und die Visualisierung im TIA Portal.

Sie verwenden SiVArc Scripting für den Aufbau von SiVArc-Ausdrücken in den Generiervorlagen. Während der Generierung wertet SiVArc alle SiVArc-Ausdrücke aus. Auf diese Weise werden zahlreiche konsistente HMI-Objekte aus einer Vorlage generiert.

Editor "SiVArc-Ausdrücke"

Wenn Sie in eine Tabellenzeile eines SiVArc-Editors klicken, um einen SiVArc-Ausdruck zu programmieren, wird ein mehrzeiliger Editor geöffnet. Der Editor für SiVArc-Ausdrücke unterstützt Sie mit verschiedenen Funktionalitäten:

- **Autovervollständigen**
Wenn Sie einen Buchstaben oder ein Zeichen eingeben, schlägt der Editor für SiVArc-Ausdrücke mögliche Operatoren, SiVArc-Objekteigenschaften, Eigenschaften und Funktionen vor, die mit diesem Buchstaben beginnen oder zu diesem Zeichen passen. Die eingegebenen Ausdrücke werden auch automatisch im Editor gespeichert.
- **Hervorhebung**
Schlüsselwörter im SiVArc-Ausdruckseditor werden mit unterschiedlichen Farben markiert. Unbekannte Wörter werden als solche gekennzeichnet. Die Tabelle zeigt die voreingestellten Farben für die wichtigsten Einträge. Die Voreinstellung können Sie unter "Optionen > Allgemein > Skript / Texteditoren" ändern.

Farbe	Bedeutung	Beispiel
Blau	Operatoren And, Or, Xor, Not Bool If-Funktion	And True If
Schwarz	Andere Operatoren Zeichen Andere Funktionen	+ , TrailNum
Dunkelcyan	String	"SG_NR"
Rot	Unbekannte Elemente	\$

- **Fehleranzeige**
Im Editor "SiVArc-Ausdrücke" werden Fehler im Skript markiert und die Fehlerursachen als Tooltips angezeigt.

Sie können bereits erstellte SiVArc-Ausdrücke ändern – wählen Sie dazu den Ausdruck aus und verwenden Sie die Befehle im Kontextmenü.

Sie können einen oder mehrere Ausdrücke kopieren oder ausschneiden und in die Registerkarte SiVArc-Eigenschaften eines anderen HMI-Objekts einfügen.

Im SiVArC-Plug-in-Editor können Sie Skripte verschieben (Drag-and-drop).

Formulierungsregeln

Für die Formulierung von SiVArC-Ausdrücken sind die folgenden Regeln zu beachten:

- Ein leerer SiVArC-Ausdruck gibt eine leere Zeichenkette zurück.
- String-Konstanten sind mit Anführungszeichen zu kennzeichnen
- In String-Konstanten sind grundsätzlich alle Zeichen erlaubt.
- Wenn Sie einen String in Anführungszeichen oder Backslashes verwenden, setzen Sie einen Backslash als Escape-Zeichen voran:
 - `\"`
 - `\\.`
- Ein Zeilenumbruch in einer String-Konstante wird mit `\n` angezeigt.
- Für den absoluten Aufruf eines Programmbausteins sind nur die folgenden Schlüsselwörter (SiVArC-Objekte) erlaubt.
 - `Block`
 - `StructureBlock`
 - `ModuleBlock`
 - `SubModuleBlock`

Um die Eigenschaften des Programmbausteins festzulegen, verknüpfen Sie ein SiVArC-Objekt durch einen Punkt mit den SiVArC-Objekteigenschaften, z.
 B. `ModuleBlock.SymbolicName` für den Zugriff auf den symbolischen Namen.

Dateneingabe als Binärcode

Um Daten im Binärcode einzugeben, verwenden Sie Präfix "2#", z. B. `2#00000101`, um anzuzeigen, dass Bit 0 und Bit 2 einer Variablen gesetzt sind.

Beachten Sie bei Verwendung von Binärcodes Folgendes:

- Verwenden Sie, falls erforderlich, alle Operatoren mit dem Binärcode, z. B. `2#1010 + 2#1111 = 25`
- Verwenden Sie, falls erforderlich, Binärcode und SiVArC-Variablen in einem Ausdruck, z. B. `VAR_1 Or 2#11100 = 29`
- Verwenden Sie, falls erforderlich, Binärcode und andere Konstantenwerte, z. B. `25 * 2#11100 = 700`
- Ein Binärcode kann bis zu 32 Bits enthalten.
- Sie können die binäre Formatierung auch mit der Funktion "Format" einstellen. Verwenden Sie dafür "b" als zweiten Operanden.

Weitere Informationen

Ausführliche Informationen über den Aufbau von SiVArC-Ausdrücken finden Sie in der Referenz.

Siehe auch

Referenz (Seite 237)

6.2.6 SiVArc-Ausdruck

6.2.6.1 Übersicht zu SiVArc-Ausdrücken

Definition

Ein SiVArc-Ausdruck ist eine Funktion, die einen Text zurückgibt. Mit diesen Texten werden ausgewählte Eigenschaften generierter HMI-Objekte befüllt.

Der SiVArc-Ausdruck greift über SiVArc-Objekte auf Textquellen zu. Die SiVArc-Objekte adressieren Bausteine im Programmaufruf in STEP 7, in den Bediengerät- oder Bibliotheksdaten.

Im Gegensatz zum ES- oder Runtime-Scripting verbindet der SiVArc-Ausdruck Daten und Strukturen aus anderen Editoren des TIA Portals dauerhaft mit der WinCC-Projektierung. Veränderungen und Anpassungen im Anwenderprogramm, in der Bibliothek oder an den Bediengeräte wirken direkt auf die Visualisierung ein.

Syntax-Elemente eines SiVArc-Ausdrucks

Der SiVArc-Ausdruck ist gemäß SiVArc-Scripting aufgebaut.

Folgende Syntax-Elemente sind in einem SiVArc-Ausdruck möglich:

- SiVArc-Objekte
- SiVArc-Objekteigenschaften
- SiVArc-Variablen
- Boolesche Werte `True / False`
- Strings
- Zahlen
- Operatoren
- Vordefinierte Funktionen
- If-Bedingungen

Konfigurationen mit SiVArc-Ausdrücken

Sie verwenden SiVArc-Ausdrücke für folgende Konfigurationen:

- Bedingungen für die Generierung von HMI-Objekten formulieren

Hinweis

Achten Sie auf die korrekte Schreibweise bei der Formulierung von Namen oder Adressierungen in Bedingungen.

Eine Fehlermeldung wird erst bei der Generierung ausgegeben.

- Dynamisches Generieren von Eigenschaften, Ereignissen und Animationen für die Visualisierung
- Ablageort und Ablagestruktur von externen Variablen konfigurieren
- Ablagestrukturen von generierten HMI-Objekten konfigurieren

Siehe auch

SiVArc-Variablen (Seite 110)

SiVArc-Objekteigenschaften (Seite 259)

Funktion "Format" (Seite 262)

Aufbau von SiVArc-Ausdrücken (Seite 119)

6.2.6.2 SiVArc-Variablen

Definition

SiVArc-Variablen sind benutzerdefinierte Variablen. Für den Organisationsbaustein "Main (OB1)" und für jedes Netzwerk können Sie mehrere Variablen anlegen.

Sie definieren den Variablennamen und den gewünschten Wert.

Mit einer SiVArc-Variablen hinterlegen Sie im Anwenderprogramm instanzspezifische Informationen zu einem Programmbaustein. SiVArc-Variablen verwenden Sie in SiVArc-Ausdrücken und Bedingungen.

Hinweis

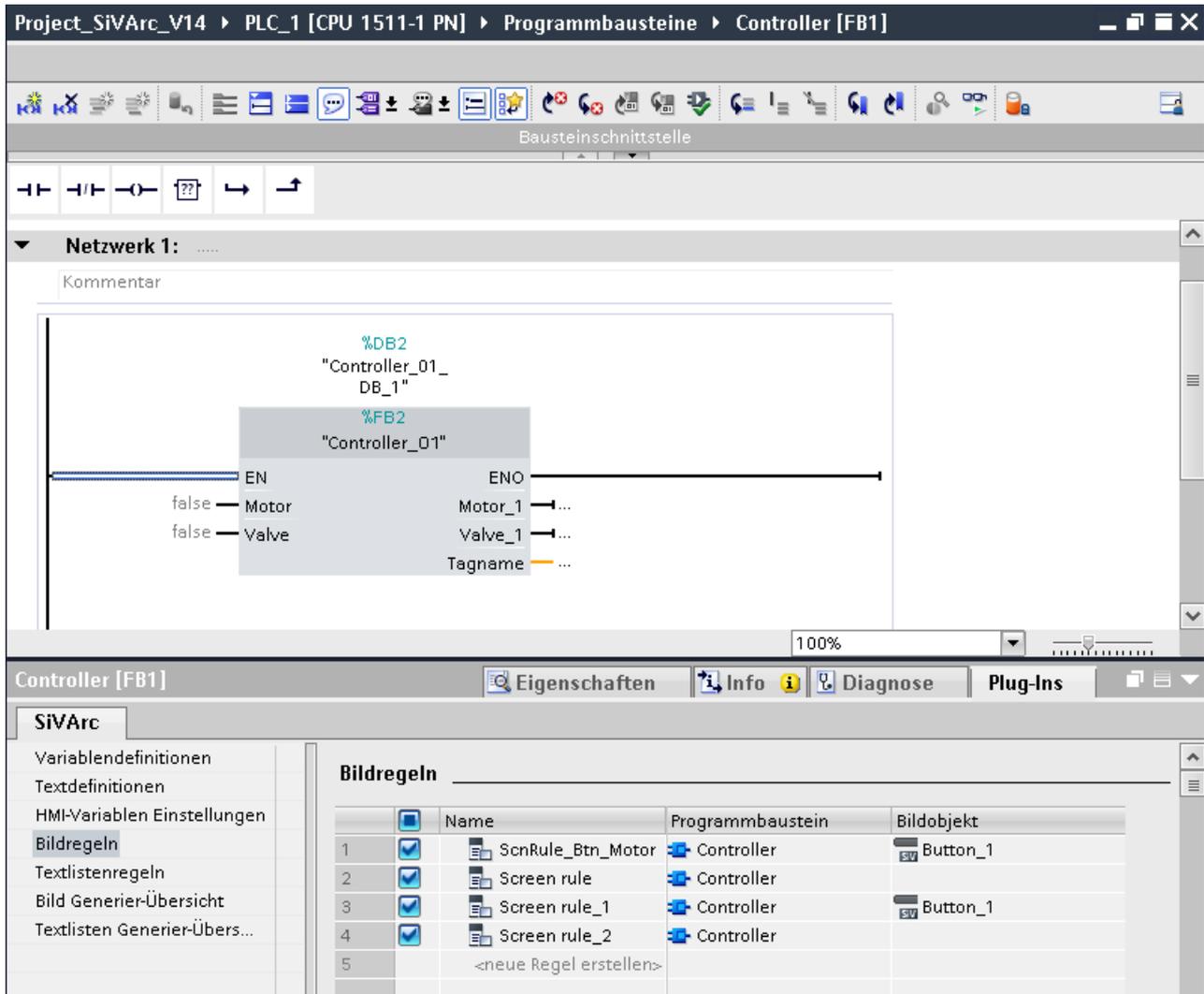
Einsatz von SiVArc-Variablen in Bildregeln

Wenn Sie SiVArc-Variablen in Bildregeln zum Auswerten von instanzspezifischen Informationen verwenden, verwenden Sie die Funktion `IsDefined("Variablenname")`. Damit fragen Sie ab, ob eine SiVArc-Variable vorhanden ist. Damit vermeiden Sie einen Generierfehler auf Grund von einer nicht vorhandenen Variablen.

Anlegen und Verwenden einer SiVArc-Variablen

Eine SiVArc-Variable legen Sie im Netzwerk oder Programmbaustein an.

Um das richtige Netzwerk auszuwählen, klicken Sie im Inspektorfenster eines Programmbausteins auf einen beliebigen Bereich im Netzwerk.



Anhand der Darstellung im Inspektorfenster entscheiden Sie, ob die Variable in diesem Netzwerk angelegt werden soll.

Sie verwenden SiVArc-Variablen wie folgt:

- Am Netzwerk
Die Variablendefinition ist in diesem Netzwerk gültig.
- Am Programmbaustein
Die Variablendefinition ist in allen Netzwerken in diesem Programmbaustein gültig. Über die Variablen adressieren Sie alle Programmbausteine, die aus dem entsprechenden Programmbaustein aufgerufen werden.
Wenn Sie SiVArc-Variablen in einem Programmbaustein verwenden, muss die SiVArc-Variablen im aufrufenden Baustein stehen.
Beispiel:
In FB1 ist eine SiVArc-Variablen definiert. FB1 ruft FB2 auf. Um den Zugriff auf die SiVArc-Variablen zu ermöglichen, definieren Sie eine Bildregel für FB2.

Hinweis**Priorisierung der SiVArc-Variablen**

Wenn Sie mehrere SiVArc-Variablen mit demselben Namen verwenden, wird der Eintrag verwendet, den SiVArc zuletzt gefunden hat. Wenn z. B. eine SiVArc-Variable für einen Netzwerk- und für einen Programmbausteinkommentar denselben Namen hat, verwendet SiVArc den Variablenwert für den Netzwerkkommentar.

Siehe auch

Übersicht zu SiVArc-Ausdrücken (Seite 109)

Editor "Bildregeln" (Seite 27)

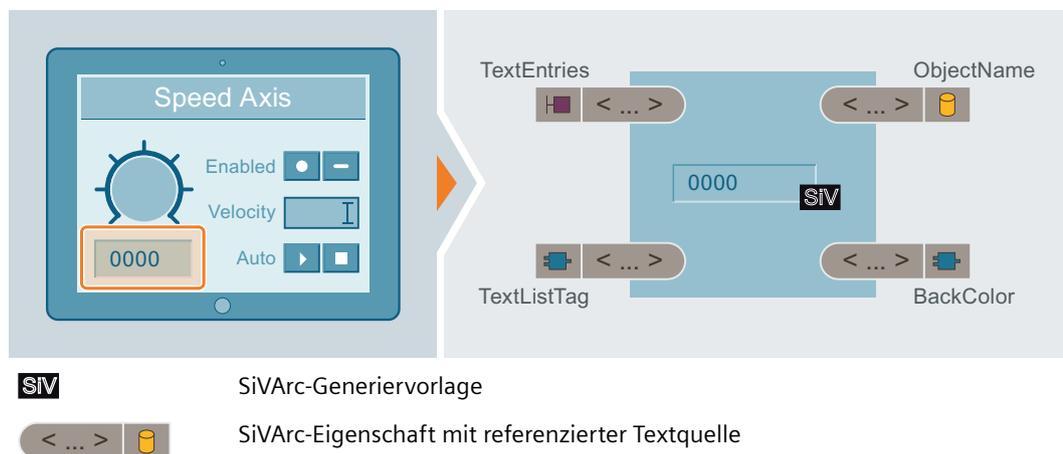
SiVArc-Texte (Seite 104)

6.2.7 Anforderungen an eine Generiervorlage**Optimierte Wiederverwendbarkeit**

Die wichtigste Anforderung an eine Generiervorlage ist eine hohe Wiederverwendbarkeit. Die beste Wiederverwendbarkeit erreichen Sie über das Typ-Instanz-Konzept aus der Bibliothek.

Wenn z. B. im Anwenderprogramm Bausteine über die Bibliothek firmenweit zur Verfügung gestellt und verwendet werden, ist es sinnvoll, dem Bausteintyp aus der Bibliothek bereits ein Set an Generiervorlagen zuzuordnen.

Diese Generiervorlagen parametrieren Sie dann mit SiVArc-Eigenschaften so, dass jede Instanz im Anwenderprogramm und jede Instanz des Bibliothekstyps in der Projektnavigation mit diesem Set an Generiervorlagen visualisiert werden kann. Dabei sollten möglichst viele Varianten möglich sein.



Generiervorlagen als Typen

Einzelne Generiervorlagen können selbst auch als Typ angelegt werden und behalten so den direkten Bezug zu den generierten HMI-Objekten.

Da die aus Typen erzeugten HMI-Objekte Typ-Instanzen sind, gelten dafür die Regeln des Typ/Instanz-Konzepts der Bibliotheken.

Hinweis

Typ-Version

SiVArc verwendet ausschließlich die aktuellste Version eines Typs. Wenn die Instanzen des FC- oder FB-Typs im Projekt nicht aktualisiert sind, bricht SiVArc die Generierung ab.

Aktualisieren Sie vor jeder SiVArc-Generierung in Ihrem Projekt alle Typen.

Wenn Sie Typen als Generiervorlagen verwenden, bewerten Sie genau die Aufgabe und die Anwendungsart eines HMI-Objekts. Für Navigationsschaltflächen sind z. B. Kopiervorlagen besser geeignet.

Regeln zum Verwenden von Typen in Bildregeln

Wenn Sie Typen verwenden, gelten folgende Regeln:

- Wenn ein Typ aus der globalen Bibliothek verwendet wird, erzeugt SiVArc beim Generieren eine Kopie des Typs in der Projektbibliothek.
- Sobald am Typ SiVArc-Ausdrücke bearbeitet werden, ist eine erneute SiVArc-Generierung erforderlich.
- Sonstige Änderungen am Typ werden automatisch in den verwendeten Instanzen aktualisiert, auch in von SiVArc generierten Instanzen des Typs.

Hinweis

Parallele Verwendung von Typen und Instanzen

Wenn Sie für eine im Projekt vorhandene Instanz eines Typs und für den Typ selbst Bildregeln definieren, verarbeitet SiVArc den Typ doppelt.

Stellen Sie sicher, dass SiVArc entweder die Instanz oder den Typ verarbeitet.

Geräteabhängigkeit

Die Verfügbarkeit von Bildobjekten und Anzeigegrößen sind abhängig vom Bediengerät. Beachten Sie bei der Erstellung von Generiervorlagen für welche Geräte die Generiervorlage verwendbar ist. Erstellen Sie unterschiedliche Positionierungsschemata, die Sie mit Ihrer Generiervorlage verknüpfen, um die Anordnung der generierten HMI-Objekte für unterschiedliche Bediengeräte steuern zu können.

Außerdem können Sie Überlaufbilder in der Generiervorlage einrichten, um die korrekte Positionierung für unterschiedlich große Bediengeräte sicherzustellen.

Parametrierung der Generiervorlage

Um eine Generiervorlage möglichst oft einsetzen zu können, muss sie konsistent in der Benennung und Beschriftung sein. Möglichst viele Eigenschaften sollten mit den passenden Verwendungsstellen im Anwenderprogramm verknüpft sein.

Außerdem berücksichtigt eine Generiervorlage im Idealfall auch die Ablagestrukturen im Projekt und die Mehrsprachigkeit eines Projekts. Dazu verwenden Sie strukturierende SiVArc-Objekteigenschaften wie z. B. "`<Object>.FolderPath`" und mehrsprachig projektierbare Ausdrücke wie z. B. "`DB.Comment`".

Dynamische Größenanpassungen

Ändern Sie Größe von Bildfenstern, Bildbausteinen und Textfeldern immer manuell.

Die dynamische Größenanpassung wird von SiVArc zwar unterstützt, kann jedoch zu unerwünschten Effekten führen, z. B. zu einer Überlappung der Bildobjekte.

Ändern von Generiervorlagen

Um Änderungen und Optimierungen an einer Generiervorlage beim nächsten Generieren zu übernehmen, legen Sie die geänderte Generiervorlage wieder in der Bibliothek ab. In den Bildregeln werden die Namen der Generiervorlage referenziert. Eine aktualisierte Generiervorlage muss daher in der Bibliothek unter dem gleichen Namen wie die ursprüngliche Generiervorlage abgelegt sein. Sonst ist die zugehörige Bildregel ungültig.

Vorteile

Die Pflege und Optimierung der Generiervorlagen unterstützen Sie dabei, effizient mit SiVArc zu arbeiten. Ihr SiVArc-Projekt bleibt dadurch agil und leicht anpassbar an andere STEP 7-Anwenderprogramm, die mit standardisierten Strukturen und Namenskonzepten arbeiten.

Auf diese Weise setzen Sie z. B. schon parallel zur Erstellung des Anwenderprogramms Ihr SiVArc-Projekt auf. Die Standardisierung in Ihrem Unternehmen kann mit SiVArc aufgebaut und gepflegt werden, auch für mehrsprachige Projekte.

Auf der anderen Seite sind die Generiervorlagen individuell anpassbar, um auch weniger standardisiert Projekte mit SiVArc umzusetzen.

Siehe auch

Generiervorlagen in SiVArc (Seite 93)

6.2.8 Parametrierungskonzepte

6.2.8.1 Beispiel: Unified-Bildbausteine generieren

Bildbausteine in Unified-Bildern

SiVArc unterstützt die folgenden Bildbausteineigenschaften eines Unified-Geräts:

- Variablenschnittstelle – Sie können hier den Namen der Variablenschnittstelle, die (von Variablen unterstützten) Datentypen, Benutzerdatentypen (von Variablen unterstützte UDT) konfigurieren
- Eigenschaftenschnittstelle – Sie können die Eigenschaften der Schnittstelle und den Datentyp (Farbe, Ressourcenliste) konfigurieren

Voraussetzung

- Die Generiervorlage für den Bildbaustein ist in der Bibliothek gespeichert
- Das Unified-Bild ist in einer Bildregel konfiguriert

Vorgehensweise

1. Bildbaustein für ein Unified-Gerät erstellen. Weitere Informationen zu Bildbausteinen finden Sie unter Bildbausteine generieren. (Seite 149)
2. Fügen Sie im Unified-Bildbaustein eine Schaltfläche hinzu. Die Plug-in-Eigenschaften einer Schaltfläche werden automatisch aktualisiert.
3. Sie können die Variablen- und die Eigenschaftenschnittstelle für die Bildbausteine konfigurieren, wobei alle Schritte im Plug-in-Eigenschaftenbereich übernommen werden.
4. Konfigurieren Sie die Bildregel mit dem Bildobjekt als Unified-Bildbaustein.
5. Bei der SiVArc-Generierung wird der Unified-Bildbaustein als Bildbaustein-Container generiert. Der Container wird mit dem Bildbaustein verknüpft und bezieht sich auf den zuletzt verfügbaren Bildbaustein in der Bibliothek. Die beim Konfigurieren der Bildbausteine eingestellten Schnittstelleneigenschaften stehen in den Eigenschaften des generierten Unified-Bildbausteins zur Verfügung.

Siehe auch

Beispiel: Bildbausteine mit Animationen generieren (Seite 149)

6.2.8.2 Beispiel: Erstellen eines Parametrierungskonzepts

Einleitung

Um mit SiVArc möglichst viele HMI-Objekte automatisiert zu generieren, gibt es unterschiedliche Ansätze und Möglichkeiten.

Beispielsszenario

Ein Ingenieurbüro wird beauftragt, Generiervorlagen auf Basis eines fertigen Anwenderprogramms abzuleiten. Das Projekt ist sehr umfangreich und weist einen hohen Standardisierungsgrad auf.

Um Anpassungs- und Erweiterungsaufwände minimieren, entscheidet sich das Ingenieurbüro nach einer Analyse des Projekts dafür, möglichst viele Texte aus dem modularen Anwenderprogramm für die Visualisierung abzuleiten und mit SiVArc zu arbeiten.

Weil das Anwenderprogramm so modular aufgebaut und standardisiert ist, kann die Anzahl von SiVArc-Projektierungen minimiert werden:

- minimale Anzahl von Generiervorlagen
- minimale Anzahl von SiVArc-Regeln

Wenn die nächste Erweiterung des Projekts ansteht, kann das Ingenieurbüro mit wenigen Anpassungen im SiVArc-Projekt die erweiterte Visualisierung generieren.

Vorteile

Durch den strukturierten Aufbau der Ausdrücke und die konsequente Zuordnung von Anweisungen und HMI-Objekten entsteht ein übersichtliches und durchschaubares HMI-Projekt. Änderungen in der Anlage oder im Anwenderprogramm können schnell und sicher umgesetzt werden. SiVArc vereinfacht damit wiederkehrende Aufgaben. Fehler können auf diese Weise vermieden werden.

Des Weiteren können Sie Unternehmensstandards einfacher umsetzen.

6.2.8.3 Zuordnung Baustein und Generiervorlage

Einleitung

Die Standardisierung des Anwenderprogramms kann in der SiVArc-Projektierung abgebildet werden. Je besser das Anwenderprogramm aufgebaut ist hinsichtlich Struktur, Modularität und Standardisierung, desto höher ist die Anwendungsqualität der SiVArc-Projektierung.

Aus den Modulen des Anwenderprogramms leitet sich der optimale Grundbezug zwischen einem HMI-Objekt und einem Funktionsbaustein ab.

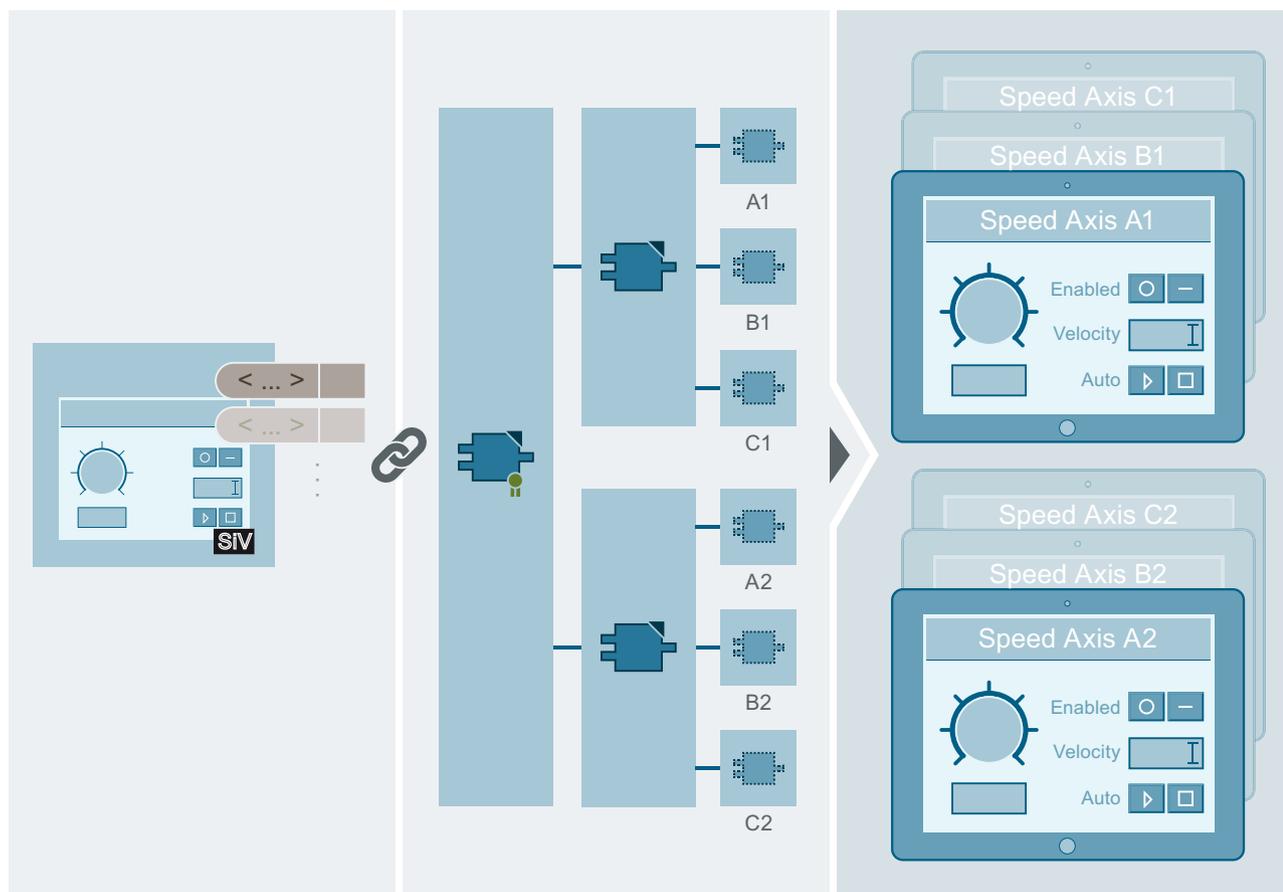
Beispielsszenario

Das Anwenderprogramm verwendet zur Steuerung aller virtuellen Drehzahlachsen die gleiche Anweisung. Diese Anweisung ist als Typ in der Bibliothek abgelegt. In STEP 7 werden zwei Typ-Instanzen verwendet und im Anwenderprogramm instanziiert.

SiVArc-Konzept

Folgendes Konzept leitet sich aus dem Beispiel ab:

- Die Steuerung der Drehzahlachse wird in einem Bildbaustein abgebildet.
- Dieser Bildbaustein wird als Generiervorlage für SiVArc parametrierung. Bezeichnungen und Verschaltungen sind über SiVArc-Eigenschaften parametrierung.
- Die SiVArc-Eigenschaften greifen auf Texte zu, die an der jeweiligen Verwendungsstelle im Anwenderprogramm erst definiert werden, z. B. die Bezeichnung der Drehzahlachse im Netzwerk-Titel.
- In der Bildregel ist der Bibliotheks-Typ der Anweisungen mit der Bildbaustein-Generiervorlage verknüpft.
- Während der Generierung durchläuft SiVArc alle Instanzen des Bausteintyps und alle seine instanziierten Aufrufe im Anwenderprogramm.



SIV

SiVArc-Generiervorlage



Bibliotheks-Typ der Anweisung



Instanz des Bibliotheks-Typs in STEP 7



Instanzierte Typ-Instanz in einem Netzwerk im Main OB

Ergebnis

Für jeden Aufruf einer Bibliothekstyp-Instanz wird ein Bildbaustein mit den Texten aus dem Anwenderprogramm generiert.

Mit nur einer Bildregel und einer Generiervorlage können alle Drehzahlachsen im Projekt visualisiert werden. Der Bezug zwischen dem HMI-Objekt und der Struktur des Anwenderprogramms ist optimal umgesetzt.

6.2.8.4 Aufbau von SiVArc-Ausdrücken

Konzept der SiVArc-Ausdrücke in WinCC

Um im WinCC-Projekt für Eindeutigkeit und Klarheit zu sorgen, werden SiVArc-Ausdrücke beispielsweise für den Zugriff auf die Datenbausteine der Anweisungsinstanzen oder Netzwerktitel verwendet. Daher ist bei der Benennung von Datenbausteinen und Netzwerktiteln stets auf Eindeutigkeit und Konsistenz zu achten.

SiVArc unterstützt symbolische Ausdrücke für Organisationsbausteine. Sie können Ausdrücke mittels relativer Adressierung für einen Organisationsbaustein festlegen. Die Punkte im Ausdruck "...SymbolicName" bedeuten, dass in der Aufrufhierarchie die Punkte zum Hauptbaustein führen und somit der Ausdruck bis zum Hauptbaustein aufgelöst wird. Wenn die Punkte im Ausdruck nicht zu einem Wert führen, wird ein Fehler angezeigt.

SiVArc unterstützt die absolute Adressierung für einen Organisationsbaustein.

Die folgende Tabelle zeigt beispielsweise, wie der symbolische Name "SG01_FB" eines Funktionsbausteins im generierten HMI-Objekt mit SiVArc-Ausdrücken verwendet wird.

SiVArc-Ausdruck	Ergebnis
"MyBlock"	MyBlock
"My\"Block"	My"Block
Block.SymbolicName	SG01_FB
"MyBlock_"&Block.SymbolicName	MyBlock_SG01_FB
"MyBlock_"&Block.SymbolicName&"_An"	MyBlock_SG01_FB_An

Beispiel: Eindeutige HMI-Objektnamen

Für die eindeutige Zuordnung generierter HMI-Objekte zur Visualisierung eines Prozesses, der mehrfach in einem Projekt verwendet wird, zu einem Prozess, sind die generierten Objekte mit dem Pfadaufruf zu benennen.

Beispiel für ein Textfeld, das ein Förderband bezeichnet:

- Objektname: Productionline_Instance_1_Dispatchunit_Instance_1_Conveyor
- Text: Dispatching

Hierfür sind im SiVArc-Ausdruck SiVArc-Objekte (Schlüsselwörter) zu verwenden, die eine Anweisung der ersten drei Aufrufebenen in der Aufrufhierarchie adressieren.

Beispiel: SiVArc-Ausdruck für den Objektnamen des Textfelds:

- `ModuleBlock.DB.SymbolicName&"_"&SubModuleBlock.DB.SymbolicName&"_Conveyor`

Die Benennungsregeln für jede Ebene der Aufrufhierarchie sind im Anwenderprogramm festgelegt. Für jede Prozessanzeige werden verschiedene HMI-Objekte zur Verfügung gestellt.

SiVArc-Objekt	Funktionstyp	Name der Anweisung	Symbolname des Datenbausteins	Name der Generierungsvorlage
StructureBlock	Hauptfunktion	"Plantsection"	"Plantsection_1_Instance_1_DB"	Label 01
ModuleBlock	Unterstützungsfunktion	"ProductionLine"	"ProductionLine_Instance_1_DB"	Label 02
SubModuleBlock	Standardfunktion	"DispatchUnit"	"DispatchUnit_Instance_1_DB"	Label 03
Block	Referenzierte Anweisung	"Initialize"	"Initialize_Instance_1_DB"	Button

- Prinzip des SiVArc-Ausdrucks für den Objektname eines generierten Textfelds

```
ModuleBlock.DB.SymbolicName&"_"&SubModuleBlock.DB.SymbolicName&"_<
Name_Generiervorlage>
```

- Generierter Objektname
ProductionLine_Instance_1_DispatchUnit_Instance_1_Label_03
- Generiertes Label:
DispatchUnit
- SiVArc-Ausdruck für das Label
Split("SubModuleBlock.DB.SymbolicName", "_" (1)

Beispiel: Eindeutige Triggervariablen

Um Triggervariablen im HMI-Objekt eindeutig zu verschalten, stellen Sie sicher, dass die Namen der PLC-Variablen und die Runtime-Einstellungen für die Synchronisierung der Variablen konsistent sind.

Der Variablenname aus dem symbolischen Namen des DB und der Name der PLC-Variablen werden in WinCC gebildet:

- PLC-Variablenname im DB der zweiten Aufrufebene
Activate
- Symbolischer Name des DB
Plantsection01
- Generierter HMI-Variablenname
Plantsection01_Activate

Die relevante Anweisung findet sich auf der zweiten Aufrufebene.

- SiVArc-Ausdruck des Variablennamens
StructureBlock.DB.SymbolicName&_Activate

Beispiel für Labels

- Ist für ein Label keine Eindeutigkeit verlangt, halten Sie den Namen der Anweisung möglichst kurz und knapp, damit er auf einer Schaltfläche angezeigt werden kann, zum Beispiel:

- Stop
- Activate

Im SiVArc-Ausdruck in der Generierungsvorlage weisen Sie das Label direkt über den Namen der Anweisung zu:

- SiVArc-Ausdruck: `Block.Title`
- Ist ein kurzer, knapper Name nicht möglich, verwenden Sie String-Funktionen:
 - Name des relevanten Datenbausteins: `Plantsection1_DB`
 - SiVArc-Ausdruck: `Split (StructureBlock.DB.SymbolicName, "_", 0)`
 - Generiertes Label: "Plantsection1"

Siehe auch

Übersicht zu SiVArc-Ausdrücken (Seite 109)

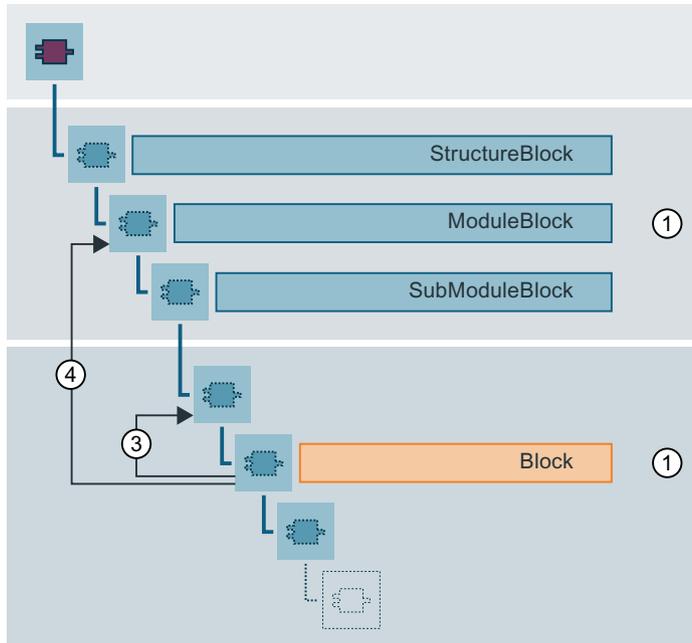
6.2.9 Einfluss des Anwenderprogramms auf eine Generiervorlage

Einleitung

Wenn Sie mit SiVArc HMI-Objekte generieren, wertet SiVArc alle Aufrufe von Programmbausteinen im Anwenderprogramm aus. Das Anwenderprogramm wird von oben nach unten durchlaufen. Wenn in einem Programmbaustein weitere Programmbausteine aufgerufen werden, bearbeitet SiVArc zuerst die Programmbausteine der untergeordneten Hierarchieebenen.

Adressierung von Programmbaustein-Eigenschaften

Die folgende Abbildung zeigt den Zusammenhang zwischen der Aufrufhierarchie von Programmbausteinen und dem Zugriff auf die Eigenschaften von Programmbausteinen:



Die Bausteine der ersten drei Aufrufebenen werden über SiVArC-Objekte abgebildet. Über die SiVArC-Objekte adressieren Sie diese Bausteine absolut.

Das SiVArC-Objekt `Block` stellt immer den Programmbaustein dar, der gerade von SiVArC bearbeitet wird - unabhängig von dessen Position innerhalb der Aufrufhierarchie.

Aus einer unteren Hierarchieebene adressieren Sie Programmbausteine aus jeder darüber liegenden Hierarchieebene. Die Adressierungsmethode ist von der aktuellen Position in der Aufrufhierarchie abhängig. In dieser Abbildung bearbeitet SiVArC gerade einen Programmbaustein in der fünften Hierarchieebene.

① Einen übergeordneten Baustein ohne SiVArC-Objekt erreichen Sie ausschließlich über dessen relative Adressierung.

Ausgehend vom gerade von SiVArC ausgewerteten Baustein, stellen Sie pro Hierarchieebene einen Punkt "." voran:

In diesem Beispiel adressieren Sie den Namen des übergeordneten Bausteins wie folgt:

.Name

② Einen übergeordneten Baustein mit SiVArC-Objekt erreichen Sie entweder über dessen relative oder absolute Adressierung:

In diesem Beispiel adressieren Sie den Baustein der zweiten Hierarchieebene wie folgt:

- Relativ: `..Name`
- Absolut: `ModuleBlock.Name`

Hinweis**Arbeiten im Editor "SiVArc-Ausdrücke"**

Die relative Adressierung wird vom Editor "SiVArc-Ausdrücke" nicht unterstützt. Um einen Baustein relativ zu adressieren, geben Sie die Adresse direkt in das Eingabefeld der SiVArc-Eigenschaft ein.

Beispiel

In einer 8-stufigen Aufrufhierarchie adressieren Sie aus einem FB der Hierarchieebene 8 wie folgt:

- Baustein aus den Aufrufebenen 1 - 3 adressieren Sie mit einem SiVArc-Objekt oder relativ, z. B. `StructureBlock.Version` oder `.....Version`
- Bausteine aus den Aufrufebenen 4 - 7 adressieren Sie relativ, z. B. `...Version` (5. Hierarchieebene)

Um die Eigenschaften eines Programmbausteins zu adressieren, verwenden Sie SiVArc-Objekteigenschaften.

Beispiele für den Zugriff auf Programmbausteine

Die folgenden Beispiele zeigen Ihnen, wie die Eigenschaften eines Programmbausteins in der jeweiligen Aufrufhierarchie adressiert werden:

Beispiel	Standardaufruf	Relativer Zugriff auf den aufrufenden Baustein	Absoluter Zugriff auf den übergeordneten Baustein in Aufrufebene 1
Zugriff auf Bausteinnamen	<code>Block.Name</code>	<code>.Name</code>	<code>StructureBlock.Name</code>
Zugriff auf symbolischen Namen des DBs	<code>Block.DB.SymbolicName</code>	<code>.DB.SymbolicName</code>	<code>StructureBlock.DB.SymbolicName</code>
Zugriff auf den Wert eines Baustein-Parameters	<code>Block.Parameters("<Name Parameter>").Value</code>	<code>.Parameters("<Name Parameter>").Value</code>	<code>StructureBlock.Parameters("<Name Parameter>").Value</code>
Zugriff auf den Kommentar einer Variable, die dem Bausteinparameter zugewiesen ist.	<code>Block.Parameters(<Name Parameter>).AssignedTag.Comment</code>	<code>.Parameters(<Name Parameter>).AssignedTag.Comment</code>	<code>StructureBlock.Parameters(<Name Parameter>).AssignedTag.Comment</code>
Zugriff auf den Pfad des adressierten Bausteins	<code>Block.FolderPath</code> <code>ModuleBlock.FolderPath</code>	<code>.FolderPath</code> Bildet die Aufrufhierarchie ab	<code>StructureBlock.FolderPath</code>
Zugriff auf den Pfad des Instanz-DBs des adressierten Bausteins Der Instanz-DB kann eine Einzelinstanz oder Multiinstanz sein.	<code>Block.DB.FolderPath</code> Hinweis: Sie referenzieren mit <code>DB.FolderPath</code> ausschließlich Bausteine, die über einen DB verfügen.	<code>.DB.FolderPath</code>	<code>StructureBlock.DB.FolderPathTagNaming</code> <code>.SeparatorChar</code>

Wenn Sie den Standardaufruf mit dem SiVArC-Objekt `Block` verwenden, wird der Programmbaustein adressiert, der gerade in einem SiVArC-Ausdruck bearbeitet wird.

Siehe auch

SiVArC-Objekteigenschaften (Seite 259)

Übersicht zu Textquellen im SiVArC-Projekt (Seite 101)

6.2.10 Einfluss der Mehrsprachigkeit auf eine Generiervorlage

Projektsprachen und Runtime-Sprachen

Bereits in der Generiervorlage können Sie das mehrsprachige Projektieren über die Funktionalität von SiVArC optimieren und effizient umsetzen.

Wenn Sie z. B. in einer Generiervorlage mehrsprachige Eigenschaften mit mehrsprachigen SiVArC-Eigenschaften projektieren, wird für jede Runtime-Sprache eine entsprechende Zeichenkette generiert.

Die Spracheinstellungen des TIA Portals und die Generiervorlage mit mehrsprachigen SiVArC-Objekten definieren, welche mehrsprachigen Texte in den generierten Objekten erzeugt werden.

Spracheinstellungen des TIA Portals

Ein SiVArC-Projekt können Sie in allen Projektsprachen generieren. Dazu aktivieren Sie die gewünschten Projektsprachen als Runtime-Sprachen.

SiVArC verwendet die Standard-Generiersprache, wenn mehrsprachige SiVArC-Objekteigenschaften an einer einsprachigen Eigenschaft gesetzt sind. Die Standard-Generiersprache ist abhängig vom Bediengerät:

- Bediengerät mit RT Advanced
Runtime-Sprache, die unter "Runtime-Einstellungen > Sprache & Schriftart > Runtime-Sprache und Schriftauswahl" an oberster Stelle steht
- Bediengerät mit RT Professional
Runtime-Sprache, die unter "Runtime-Einstellungen > Sprache & Schriftart > Runtime-Sprache und Schriftauswahl" als "Runtime-Sprache für nicht-multilinguale Objekte" gesetzt ist

Wenn eine Projektsprache nicht als Runtime-Sprache gesetzt ist, werden die mehrsprachigen Eigenschaften im Projekt für diese Projektsprache mit dem Wert aus der Kopiervorlage generiert. Der SiVArC-Ausdruck für diese Eigenschaft wird in dieser Projektsprache nicht ausgewertet.

Wenn ein Wert in einer mehrsprachigen Variablen nicht gesetzt ist, wird als Eigenschaftswert für diese Sprache der leere String generiert.

Mehrsprachige SiVArc-Objekte

Bei der Projektierung eines mehrsprachigen SiVArc-Projekts arbeiten Sie mit folgenden SiVArc-Objekten:

- Mehrsprachige Eigenschaften
- Mehrsprachige SiVArc-Objekteigenschaften
- SiVArc-Texte für Textlisteneinträge

Mehrsprachige WinCC-Eigenschaften

SiVArc unterstützt folgende mehrsprachige Eigenschaften.

Die Ausdrücke dieser Eigenschaften werden von SiVArc für jede Runtime-Sprache einzeln ausgewertet. Wenn ein Ausdruck mehrsprachige SiVArc-Objekteigenschaften enthält, ergeben sich dabei für die jeweiligen Runtime-Sprachen unterschiedliche Werte.

HMI-Objekt	Eigenschaft
Balken	Titel Tooltip Einheit
Bild	Anzeigename
Bildfenster	Titel
Textfeld	Text
E/A-Feld	Hilfetext
Grafisches E/A-Feld	Tooltip
Schalter	Titel TextAUS TextEIN Tooltip
Rundschaftfläche	Text Tooltip
Schaltfläche	TextAUS
Zeigerinstrument	Titel Einheit

Für alle anderen Eigenschaften, für die Sie einen SiVArc-Ausdruck verwenden können, wird der Ausdruck in der Standard-Generiersprache ausgewertet.

Mehrsprachige SiVArc-Objekteigenschaften

Folgende SiVArc-Objekteigenschaften sind mehrsprachig projektierbar:

- Title
- SymbolComment
- DB.Comment

- NetworkTitle
- NetworkComment

Verwendung von SiVArC-Ausdrücken im mehrsprachigen Kontext

In SiVArC-Ausdrücken verwenden Sie mehrsprachige und einsprachige SiVArC-Objekteigenschaften. Welche SiVArC-Objekteigenschaften mehrsprachig sind, ist in der Referenz beschrieben. Sie verwenden SiVArC-Objekteigenschaften in mehrsprachigen und einsprachigen SiVArC-Eigenschaften. Die SiVArC-Ausdrücke werden in diesem Zusammenhang wie folgt ausgewertet:

	Einsprachige SiVArC-Objekteigenschaften	Mehrsprachige SiVArC-Objekteigenschaften
Einsprachige Eigenschaft	Für jede Runtime-Sprache wird die gleiche Zeichenkette generiert.	Die Variable wird in der Standard-Generiersprache ausgewertet. Die Standard-Generiersprache legen Sie in den Runtime-Einstellungen des Bediengeräts fest.
Mehrsprachige Eigenschaft	Für jede Runtime-Sprache wird die gleiche Zeichenkette generiert.	Die Variable wird für alle projektierten Runtime-Sprachen ausgewertet. Für jede Runtime-Sprache wird die projektierte Zeichenkette generiert.

Siehe auch

- Title (Seite 257)
- SymbolComment (Seite 255)
- Comment (Seite 250)
- DB (Seite 240)
- NetworkTitle (Seite 254)
- NetworkComment (Seite 253)
- SiVArC-Texte (Seite 104)

6.2.11 Ablagestrategien für generierte Objekte

Übersicht

SiVArC bietet für Bilder und Variablen die Option zum Festlegen von Regeln für die Speicherstruktur der Generiervorlagen mit Hilfe von SiVArC-Ausdrücken in den Bildregeln oder in der Generiervorlage.

Hierfür gibt es verschiedene Speicherstrategien:

- Abbildung der Speicherstruktur im Projektbaum in STEP 7
- Abbildung der Speicherstruktur in der Projektbibliothek
- Individuelle Speicherstruktur

Die SiVArc-Speicherstrategien basieren auf den generierten HMI-Objekten im Projektbaum unter den HMI-Geräten in den Bereichen von Bildern und Variablen.

Die strukturierte Speicherung von SiVArc-Regeln stellt Ihnen die Funktionen von SiVArc-Editoren zur Verfügung.

Anwendungsbeispiel

Die Bausteine werden im Projektbaum gespeichert, zum Beispiel nach Funktion. Diese Speicherform kann für die betreffenden Bilder automatisch angelegt werden. Konfigurieren Sie in den Generiervorlagen der Bilder einen SiVArc-Ausdruck, der auf den Pfad der Bausteine im Projektbaum verweist.

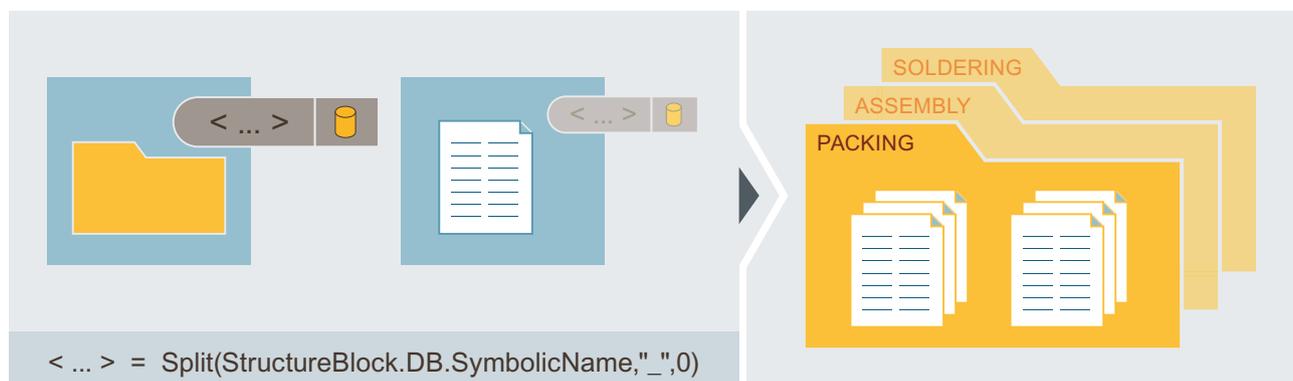
Vorteil

Die SiVArc-Speicherstrategien erhöhen die Durchgängigkeit und Standardisierung Ihres Visualisierungsprojekts. Wenn eine andere Speicherstrategie erforderlich ist, können Sie Ihr Projekt mit geringem Aufwand neu strukturieren.

Regeln für die Speicherung generierter Variablen

Für die Speicherung von Variablen sind die folgenden Strategien möglich:

- Abbildung der Speicherstruktur im Projektbaum in STEP 7
Sie können die SiVArc-Ausdrücke `HmiTag.DB.SymbolicName` und `HmiTag.DB.FolderPath` für den Editor "Variablenregeln" verwenden, um die Variablen tabellen auf der Grundlage des Steuerungsprogramms mit nur einer Variablenregel zu strukturieren.
Das Projekt wird einmalig auf der Steuerungsseite strukturiert.
- Individuelle Speicherstruktur
Den Zielordner und die Variablen tabellen definieren Sie individuell mit den Spalten "Variablen gruppe" und "Variablen tabelle".



Doppelklicken Sie in der Projektnavigation auf "Gemeinsame Daten > SiVArc > Variablenregeln", um den Editor "Variablenregeln" zu öffnen.

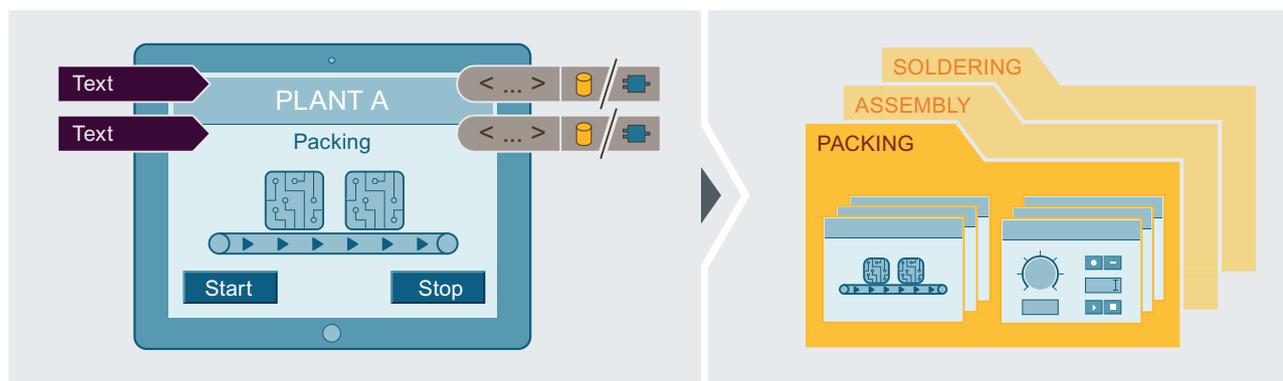
Project_Sivarc > Common data > SiVArC > Tag rules

	Name	Index	Tag group hierarchy	Tag table	Condition
1	Tag rule	0	HmiTag.DB.FolderPath	HmiTag.DB.SymbolicName	
2	<create new rule>				

Regeln für die Speicherung generierter Bilder

Für die Speicherung von Bildern sind die folgenden Strategien möglich:

- Abbildung der Speicherstruktur im Projektbaum in STEP 7
- Abbildung der Speicherstruktur in der Projektbibliothek
- Individuell



Sie steuern die Speicherung von Bildern im Projektbaum mit den Eigenschaften "Name" und "Bildgruppe" in der Generiervorlage eines Bilds. Wenn Sie einen Text unter "Bildgruppe" angeben, wird im Projektbaum eine Gruppe mit diesem Namen angelegt. In dieser Gruppe werden die anhand der Generiervorlage erstellten Bilder gespeichert.

Plantsection1 [Bild]

Eigenschaften Info Diagnose Plug-Ins

SiVArC Eigenschaften	SiVArC Animationen	SiVArC Ereignisse	Generier-Übersicht
Name	Ausdruck des statischen Werts		Ausdruck der Variablen
▼ Allgemein			
Name	Split(StructureBlock.DB.SymbolicName,"_",0)		
Anzeigename			
Kommentar			
Bildgruppe	Split(StructureBlock.SymbolicName,"_",0)		
Anzahl Überlaufbilder			
Anzahl Überlaufbilder als Bitmaske auswerten	<input type="checkbox"/>		
Navigationsschaltfläche	<input checked="" type="checkbox"/>		
Navigationsschaltfläche "Weiter"			
Navigationsschaltfläche "Zurück"			
▶ Bild als Inhalt des Bildfensters			
▶ Positionierungsschema			
▶ Darstellung			

Sie können die Speicherstruktur und die Benennung der generierten Objekte mit der Bibliothek in der Generiervorlage eines Bildtyps synchronisieren.

Verwenden Sie hierzu die SiVArc-Ausdrücke "LibraryObject.FolderPath" und "LibraryObject.Name".

SiVArc-Objekteigenschaft	Referenziertes Objekt	SiVArc-Eigenschaft
LibraryObject.FolderPath	Speicherpfad des Bildtyps in der Bibliothek	Bildgruppe: Der Speicherpfad für die Bibliothek wird im Projektbaum generiert. Name*: Das generierte Bild wird nach dem Ordner benannt, in dem der Bildtyp gespeichert ist.
LibraryObject.Name	Name des Bildtyps in der Bibliothek	Name: Das Bild wird nach dem Bildtyp benannt. Bildgruppe: Das Bild wird in einem Ordner mit dem Namen des Bildtyps im Projektbaum gespeichert.

*) Verwenden Sie `LibraryObject.FolderPath` nur dann als "Name" für die SiVArc-Eigenschaft, wenn der Bildtyp in der Bibliothek auf lediglich einer Hierarchieebene gespeichert ist. Wenn Sie eine mehrere Ebenen umfassende Speicherhierarchie verwenden möchten, können Sie den Ausdruck und `LibraryObject.FolderPath` durch den umgekehrten Schrägstrich ersetzen.

Alternativ können Sie Speicherordner und Bildnamen individuell festlegen.

Siehe auch

Variablenregel anlegen (Seite 186)

Generiervorlage für ein Bild erstellen (Seite 160)

Name (Seite 253)

FolderPath (Seite 251)

LibraryObject (Seite 243)

HMITag (Seite 242)

6.2.12 Beispiel: Erreichen von hoher Flexibilität

Beispielsszenario

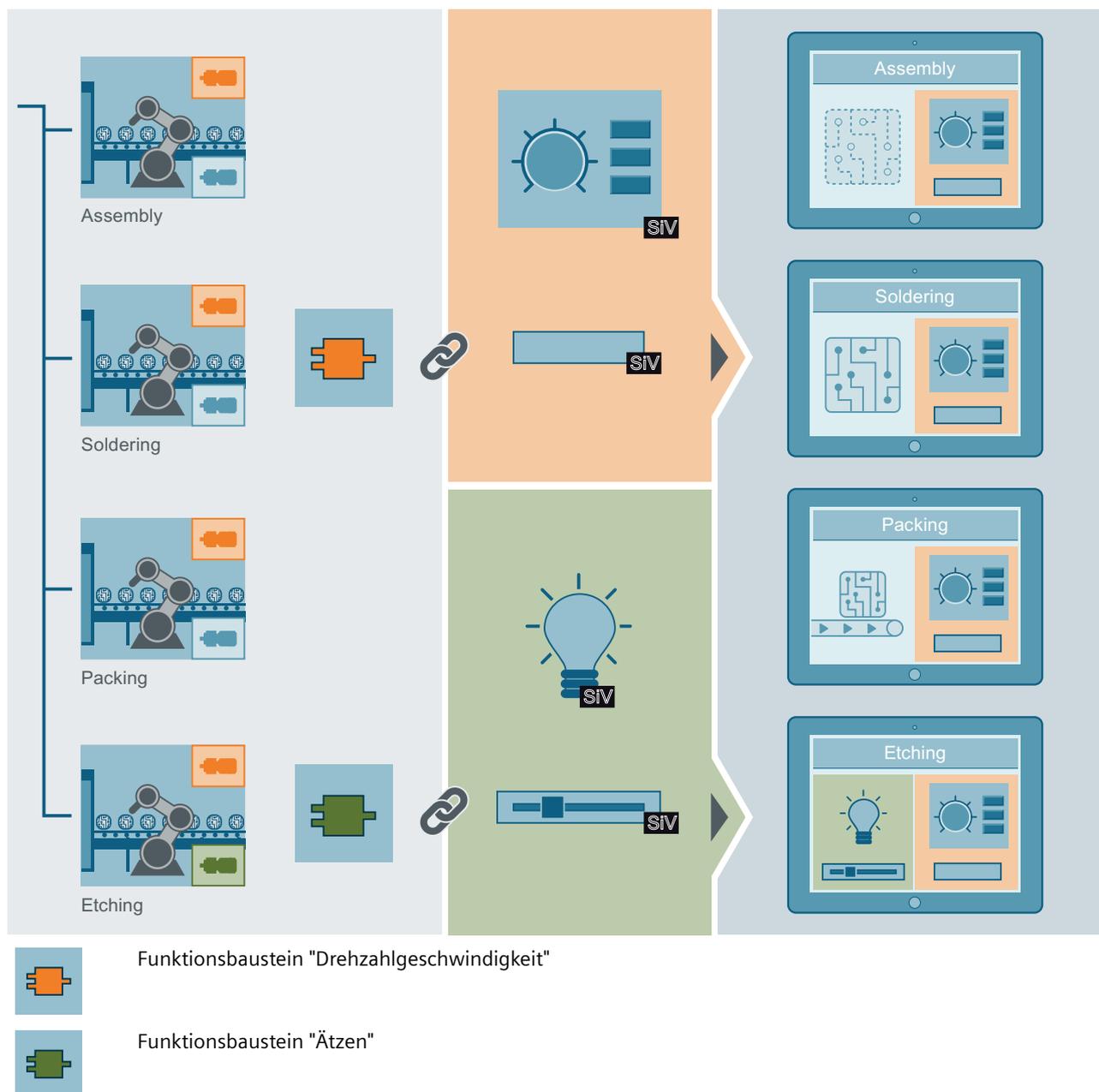
In einer Leiterplattenfabrik existieren die Anlagenbereiche "Bestücken", "Löten" und "Packen". Ein neuer Typ Leiterplatten erfordert die Planung und Ergänzung einer weiteren Fertigungsstufe "Ätzen". Die Anlage besteht zum großen Teil aus Standardbausteinen.

Nachdem der neue Anlagenbereich erstellt ist, werden die Module getestet und für die Bedienung optimiert.

Umsetzungskonzept

Ein Großteil der Funktionen aus dem Anlagenbereich "Löten" wird im Anlagenbereich "Ätzen" wiederverwendet und daher ohne weitere SiVArC-Projektierungen generiert

Für den Anlagenbereich "Ätzen" sind zusätzliche Funktionen notwendig. Diese Funktionen sind Standardfunktionen, denen bereits Generiervorlagen in Bildregeln zugeordnet sind. Der Ingenieur bildet mit den für die Fertigungsstufe "Ätzen" zusätzlich benötigten Regeln eine Gruppe und aktiviert dafür die relevanten Bediengeräte.



Um die Anlage zu testen, aktiviert der Projektteur gesammelt eine Bildregelgruppe und deaktiviert die für den Test nicht benötigten Module. Der Visualisierungsingenieur testet die

daraus generierte Bedienoberfläche. Abhängig vom Testergebnis werden Generiervorlagen und Regeln über Bedingungen oder Modifikationen der SiVArc-Eigenschaften optimiert.

6.2.13 Beispiel: Erreichen von hoher Wiederverwendbarkeit

Beispielsszenario

Ein Ingenieurbüro erhält von einem Neukunden den Auftrag, eine Standardanlage für die Leiterplattenherstellung zu projektieren.

Im Büro ist bereits ein optimiertes SiVArc-Projekt für die Leiterplattenherstellung vorhanden und soll für den Neukunden wiederverwendet werden.

Dabei soll anlagenweit auf größtmögliche Konsistenz bei der Bedienung und Darstellung gleicher Funktionen geachtet werden, z. B. folgende Funktionen:

- ON/OFF
- Fahren in Grundstellung
- Status anzeigen

Umsetzungslösung

Im standardisierten Anwenderprogramm sind viele Funktionsbausteine Standardfunktionen. Systemblöcke sind als Bibliothekstypen angelegt. Daher kann das Ingenieurbüro für jede Standardfunktion ein vollständiges, bereits vorhandenes Set an Generiervorlagen einrichten.

Die bestehenden Generiervorlagen für Standardfunktionen greifen über SiVArc-Ausdrücke auf die Textquellen direkt am Standardbaustein zu. Die Aufrufhierarchie wird nicht berücksichtigt. Die Triggervariablen sind über den Namen des Datenbausteins des Systemblocks eindeutig referenziert. Jede Wiederverwendung des Typs erzeugt in der Visualisierung die dazugehörigen Bedienobjekte auf Basis des gleichen Sets an Generiervorlagen. Anpassungen sind daher nicht erforderlich.

Farbe und Formen der Generiervorlagen werden an das Bedienkonzept angepasst und über eine projektabhängige Bibliothek zur Verfügung gestellt.

Das neue Corporate Design für die Bedienbilder wird über die Positionierungsschemata den Generiervorlagen zugeschaltet.

6.2.14 Beispiel: Generiervorlage für Bildfenster erstellen

Beispielsszenario

Zu Schulungszwecken müssen an einem redundant ausgelegten Arbeitsplatz mehrere vorhandene Bediengeräte dupliziert werden.

Ziel

An der redundanten Bedienstation müssen verschiedene Bildfenster mit den entsprechenden Startbildern generiert werden. Der Name des Bildfensters weist auf den Namen des Programmbausteins hin, der darin visualisiert wird.

Bildfenster in SiVArc

Bildfenster werden nicht direkt als Bildobjekte generiert. Ein Bildfenster wird implizit erzeugt, wenn ein Bild als Bildobjekt angegeben wird.

Wenn sich im Projekt eine Generiervorlage "DefaultScreenWindowControl" befindet, generiert SiVArc Bildfenster anhand dieser Vorlage. Wenn diese Vorlage nicht verfügbar ist, erstellt SiVArc eine Kopie des Bildfensters aus "Werkzeuge".

Voraussetzung

- Ein leeres Bedienerbild des redundanten Arbeitsplatzes wird als Generiervorlage für das Bildfenster "Screen_Training" abgelegt. Die SiVArc-Eigenschaft "Name" der Generiervorlage wird mit dem SiVArc-Ausdruck `"Block.DB.SymbolicName&"_SWC"` projiziert.
- Der Programmbaustein "Plantsection_Soldering" ist im Anwenderprogramm enthalten und wird wiederholt im OB1 aufgerufen.

Vorgehensweise

Um eine Kopie einer Generiervorlage für ein Bildfenster anzulegen, gehen Sie wie folgt vor:

1. Öffnen Sie die Generiervorlage "Screen_Training" aus der Bibliothek.
2. Projektieren Sie unter "Plug-Ins > SiVArc-Eigenschaften" den Namen des Bildfensters mit dem SiVArc-Ausdruck `"Block.DB.SymbolicName&"_SWC"`. Dabei verweist der Abschnitt `Block.DB.SymbolicName` je nach Art des Bausteinaufrufs auf einen der folgenden Namen:
 - Global: Symbolischer Name des Instanz-Datenbausteins
 - Local: Name der Bausteininstanz

Durch den Abschnitt `&"-SWC"` wird das Suffix für "Screen Window Control" zum Namen hinzugefügt.

3. Projektieren Sie im Inspektorfenster unter "Plug-Ins > SiVArc-Eigenschaften > Bild als Inhalt des Bildfensters" die gewünschten Eigenschaften:
 - Geben Sie unter "Name des Bildfensters" einen eindeutigen Namen oder einen SiVArc-Ausdruck für das Bildfenster ein, das auf dem Zielbild generiert werden soll.
 - Geben Sie unter "Variablenpräfix" den Namen der Variablen ein, die einen Anwenderdatentyp verwendet. Verwenden Sie bei Bedarf einen SiVArc-Ausdruck.

Screen_Training [Bild]			
SiVArc Eigenschaften		SiVArc Animationen	SiVArc Ereignisse
Name		Ausdruck des statischen Werts	Ausdruck der Variablen
▼ Allgemein			
Name		Block.DB.SymbolicName&"_SWC"	
Anzeigename		Soldering	
Kommentar			
Bildgruppe			
Anzahl Überlaufbilder			
Anzahl Überlaufbilder als Bitmaske auswerten	<input type="checkbox"/>		
Navigationsschaltfläche	<input checked="" type="checkbox"/>		
Navigationsschaltfläche "Weiter"			
Navigationsschaltfläche "Zurück"			
▼ Bild als Inhalt des Bildfensters			
Name des Bildfensters		Training_SWC_01	
Variablen-Präfix oder Prozessvariable		Soldering_	
Titel			
Zusätzliche Bilder generieren	<input type="checkbox"/>		
▶ Position			
▶ Positionierungsschema			
▶ Darstellung			

4. Geben Sie eine neue Bildregel mit dem Namen "Station_Training" ein.
5. Geben Sie einen Kommentar "Nur zu Schulungszwecken" ein.
6. Wählen Sie den zentralen Programmbaustein "Plantsection_Soldering" aus.
7. Wählen Sie unter "Bildobjekt" die Generiervorlage "Startscreen" aus.
8. Wählen Sie unter "Bild" die Generiervorlage "Screen_Training" aus.

GettingStartedSiVArcV2.0_Complete_V14_1 ▶ Gemeinsame Daten ▶ SiVArc ▶ Bildregeln						
	Name	Programmbaustein	Bildobjekt	Bild	Kommentar	
1	Plantsection_Title	Plantsection_Soldering	Plantsection_Title	Plantscreen		
2	Plantsection_Stat...	Plantsection_Soldering	PlantStatus_Symb...	Plantscreen		
3	Productionline_Title	Productionline	Productionline_title	Plantscreen		
4	Conveyor	Conveyor	Conveyor	Plantscreen		
5	Productionline_Po...	Productionline	Position_IO	Plantscreen		
6	Processing_Unit	Processing	ProcessingUnit	Plantscreen		
7	Activate_Btn	Activate	Function_Activate	Plantscreen		
8	Stop_Btn	Stop	Function_Stop	Plantscreen		
9	Station_Training	Plantsection_Solder...	Startscreen	Screen_Training	Training_Only	
10	<neue Regel erstellen>					

Ergebnis

Ein Bild "Screen_Training" wird generiert. Es enthält für jeden Aufruf des Programmbausteins "Plantsection_Soldering" ein Bildfenster. Das Startbild der Anlage "Soldering" ist im Bildfenster enthalten.

Der Name des Bildfensters enthält einen Bezug zu dem im Fenster visualisierten Programmbaustein sowie das Suffix SWC, beispielsweise "Plantsection_Soldering_Instance01_SWC".

Bildfenster für mehrere Bilder

Um in einem generierten Bildfenster weitere Bilder anzuzeigen, beispielsweise Diagnosebilder, platzieren Sie die gewünschte Generiertvorlage im selben Bibliotheksordner, z. B. "Training_Screens". Projektieren Sie außerdem die folgenden SiVArC -Eigenschaften:

- Projektieren Sie in dem Bild, das Sie als "Bildobjekt" ausgewählt haben, die SiVArC-Eigenschaft "Zusätzliche Bilder generieren".
- Projektieren Sie in dem Bild, das Sie als "Bildobjekt" ausgewählt haben, die SiVArC-Eigenschaft "Bild im Bildfenster".

Siehe auch

Beispiel: Generierung einer Bildvorlage mithilfe der Kopierregel (Seite 134)

6.2.15 Beispiel: Generierung einer Bildvorlage mithilfe der Kopierregel

Beispiel: Generierung einer Bildvorlage mithilfe der Kopierregel

Wir empfehlen, zu Schulungszwecken die Teilemontage in einer Automobilfertigungsanlage zu programmieren, wobei "PLC" den Hauptbaustein als "Assemble_body", "Production_time" und "Production_order" als Funktionsbausteine enthält.

Ziel

In einer Automobilproduktionsanlage erhält ein Ingenieurbüro den Auftrag, bei der SiVArC-Generierung eine einzige Bildvorlage auf alle Bediengeräte anzuwenden. Mithilfe der Kopierregel werden von der Bildvorlage mehrere Kopien erstellt und mit benutzerdefinierten SiVArC-Ausdrücken auf alle in den PLCs projektierten Bausteine angewendet. Auf diese Weise optimiert die Kopierregel über Bildvorlagen die Abläufe und steigert die Produktivität einer Produktionsanlage.

Gehen Sie wie unten beschrieben vor, um mithilfe von Kopierregeln Bildvorlagen mit dem Namen "Production_order" zu erstellen.

Voraussetzung

- Mit PLC-Bausteinen projektierte Bediengeräte.
- Projektierte Bildvorlage.

Vorgehensweise

Gehen Sie wie folgt vor, um mithilfe von Kopierregeln Bildvorlagen anzulegen:

1. Erstellen Sie unter "Vorlagen > Neue Vorlage hinzufügen" eine neue Vorlage.
2. Projektieren Sie im Fenster "Eigenschaften" der Bildvorlage die Vorlage namens "Production_order".
3. Fügen Sie unter "SiVArc > Kopierregeln" die Bildvorlage in den Ordner "Kopiervorlagen" ein.
4. Klicken Sie auf "Neue Regel erstellen".
5. Geben Sie den Namen der Kopierregel ein.
6. Suchen Sie unter "Bibliotheksobjekt" die Vorlage im Ordner "Kopiervorlagen". Bibliotheksobjekte unterstützen IntelliSense.
7. Generieren Sie auf dem Bediengerät die SiVArc-Visualisierung.

Ergebnis

Eine Bildvorlage wird im Ordner "Vorlagen" mit dem Namen "Production_order" erstellt. Der Name der vorhandenen Bildvorlage erhält das Suffix "_Renamed".

Siehe auch

Generiervorlagen in SiVArc (Seite 93)

6.2.16 Beispiel: Generierung eines HMI-Bilds mithilfe von Vorlageneigenschaften

Beispiel: Beispielszenario für die Generierung eines HMI-Bilds mithilfe von Vorlageneigenschaften

Wir empfehlen, zu Schulungszwecken die Teilemontage in einer Automobilproduktionsanlage zu programmieren, wobei "PLC" den Hauptbaustein als "Assemble_body", Funktionsbausteine als "Production_time", "Production_order" und die Bildvorlage als "Production_order" enthält.

Ziel

Wenn in der Automobilproduktionsanlage mehrere HMI-Bilder mithilfe von Vorlagen generiert werden müssen, wird hierzu eine Vorlageneigenschaft verwendet. Mithilfe der Eigenschaft "SiVArc" werden die Vorlagen hinzugefügt und auf alle Bausteine angewandt, die in den PLCs mit benutzerdefinierten SiVArc-Ausdrücken erstellt wurden. Im Ergebnis profitieren Sie von erhöhter Produktivität und Skalierbarkeit über mehrere Bediengeräte hinweg.

Um HMI-Bilder mithilfe der Vorlageneigenschaft "Production_order" zu generieren, gehen Sie wie folgt vor.

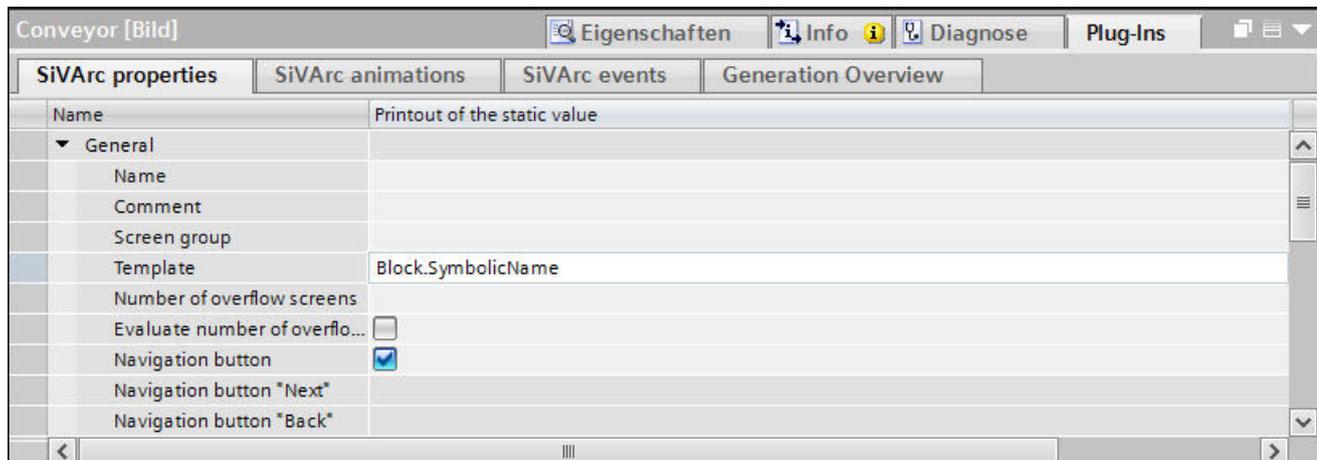
Voraussetzung

- Mit PLC-Bausteinen projektierte Bediengeräte.
- Projektierte Bildvorlage.

Vorgehensweise

Um HMI-Bilder mithilfe von Vorlageneigenschaften zu generieren, gehen Sie wie folgt vor:

1. Wählen Sie "Bilder > Neues Bild hinzufügen > Conveyor".
2. Projektieren Sie unter "Plug-Ins > SiVArC-Eigenschaften" den Namen des Bilds mit dem SiVArC-Ausdruck "Block.SymbolicName".
3. Dabei verweist der Abschnitt `Block.SymbolicName` je nach Art des Bausteinaufrufs auf einen der folgenden Namen:
 - Global: Symbolischer Name des Bausteins
 - Local: Name der Bausteininstanz
4. Im Inspektorfenster unter "Plug-Ins > SiVArC-Eigenschaften > Vorlage" können Sie entweder den Vorlagennamen eingeben oder einen SiVArC-Ausdruck so projektieren, dass der Ausdruck in einen in der Projektbibliothek vorhandenen Vorlagennamen aufgelöst wird. Der aufgelöste Name des Ausdrucks erscheint im Zielbild.



5. Platzieren Sie das projektierte HMI-Bild im Ordner "Kopiervorlagen".
6. Klicken Sie unter "SiVArC > Bildregeln" auf "Neue Regel erstellen". Geben Sie als Namen der Bildregel "Förderbandregel" ein.
7. Suchen Sie unter "Bibliotheksobjekt" einen PLC-Baustein sowie die Bildvorlage im Ordner "Kopiervorlagen". Das Bibliotheksobjekt und der Programmbaustein unterstützen IntelliSense.
8. Generieren Sie auf dem Bediengerät die SiVArC-Visualisierung.

Ergebnis

Ein HMI-Bild "Conveyor" wird mit dem Namen "Production order" als Vorlagenname erstellt. Der Name der bisherigen Bildvorlage erhält das Suffix "_Renamed".

Der folgende Screenshot zeigt den Vorlagennamen, der nach der SiVArc-Generierung automatisch angewendet wird.



Siehe auch

Generiervorlagen in SiVArc (Seite 93)

6.2.17 Beispiel: Generiervorlage mit Animation erstellen

Beispielsszenario

Wenn ein Roboter in Grundstellung fährt, soll grundsätzlich der Roboter im Bild mit Farbwechsel blinken.

Ziel

Die Generiervorlage für den Roboter ist ein Grafisches E/A-Feld, das mit einer Gestaltungsanimation parametrisiert ist. Die Statusvorgaben folgen einem SiVArc-Ausdruck.

SiVArc-Animationen

SiVArc unterstützt folgende Arten von Animationen:

- Animationen mit Variablenanbindung (nur verfügbar in WinCC Runtime Professional für S7-GRAPH-Übersicht)
- Animationen der Kategorie "Anzeige"

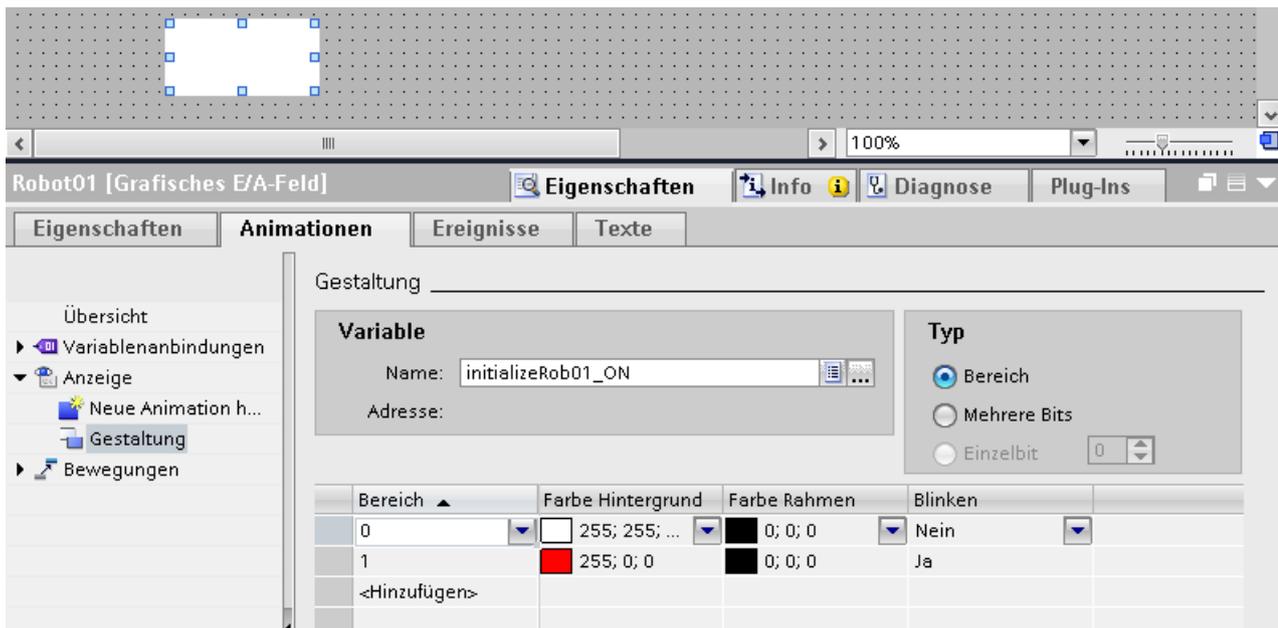
Für diese Animationen definieren Sie über einen SiVArc-Ausdruck die Prozessvariablen, welche die Animation in Runtime auslöst.

Voraussetzung

- Ein Grafisches E/A-Feld "Robot01" ist als Generiervorlage für die Darstellung des Roboters projektiert.
- Die PLC-Variablen sind mit den HMI-Variablen synchronisiert.
- Die Variable "initializeRob01_ON" enthält die Statusinformation der Fahrt in die Grundstellung und ist mit der externen Variable "Soldering_Instance_01_initializePosRob01" verbunden.
- Der Programmbaustein "Rob01" ist im Anwenderprogramm enthalten.
- Eine Bildregel ist erstellt, die das Grafische E/A-Feld "Robot01" mit dem Programmbaustein "Rob01" verknüpft.

Vorgehen

1. Öffnen Sie die Generiervorlage des Grafischen E/A-Feldes.
2. Projektieren Sie eine Gestaltungsanimation.



- Wählen Sie den Typ "Bereich"
 - Wählen Sie für den Bereich "1" Rot als Hintergrundfarbe und die Option "Blinken"
3. Öffnen Sie unter Eigenschaften > PlugIns" die Registerkarte "SiVArC-Animationen".
 4. Parametrieren Sie für die Animation "Gestaltung" unter "Ausdruck der Variablen" den SiVArC-Ausdruck "StructureBlock.DB.SymbolicName&"_initialPosRob01""
 5. Überschreiben Sie in der Bibliothek die bestehende Generiervorlage.

Ergebnis

Beim Generieren wird für jede Instanz des Programmbausteins "Rob01" das Grafische E/A-Feld "Robot01" erstellt. Für jedes Grafische E/A-Feld wurde die Animation projektiert. Wenn in Runtime der Roboter in Grundstellung fährt, blinkt der Roboter im Bild in Rot.

6.2.18 Beispiel: Generiervorlage mit Ereignisprojektierung erstellen

Beispielszenario

Die Schaltfläche "Activate" soll das Fahren in die Grundstellung des Fräs-/Löt- oder Positionierungsroboters triggern.

Ziel

In der Generiervorlage der Schaltfläche "Activate" ist das Ereignis "Klicken" mit der Systemfunktion "SetzeBit" parametrieret.

Der eindeutige Parameter "Variable" für die Systemfunktion wird beim Generieren aus den Textquellen aus STEP 7 zusammengesetzt.

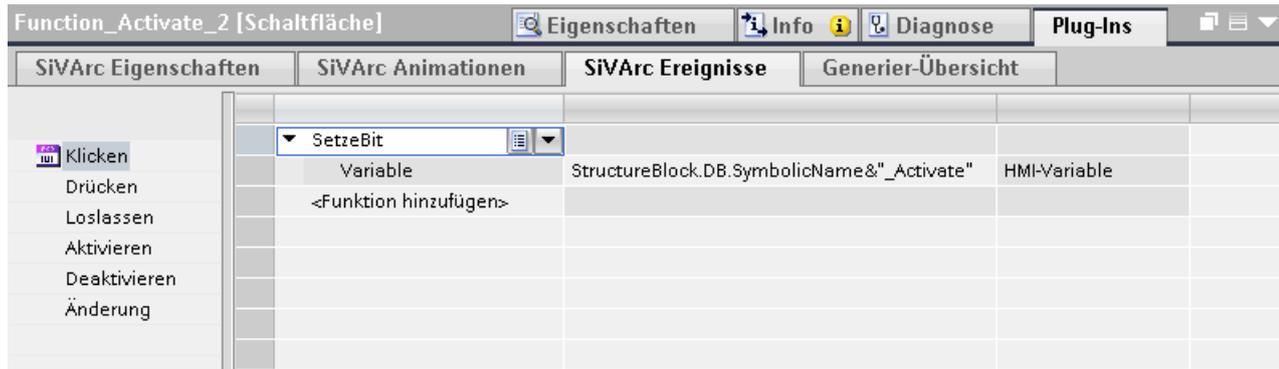
Voraussetzung

- Die Generiervorlage der Schaltfläche "Activate" mit der Systemfunktion "SetzeBit" ist parametrieret.
- Eine Bildregel ist angelegt, in der die Generiervorlage "Activate" mit dem relevanten Funktionsbaustein verknüpft ist.
Der Funktionsbaustein befindet sich in unserem Beispiel in der zweiten Ebene der Aufrufhierarchie und wird über das SiVArc-Objekt "StructureBlock" adressiert.

Vorgehensweise

Um eine Generiervorlage mit Ereignisprojektierung zu erstellen, gehen Sie folgendermaßen vor:

1. Öffnen Sie die Generiervorlage Schaltfläche "Activate" in WinCC.
2. Projektieren Sie im Inspektorfenster unter "Plug-Ins > SiVArC-Ereignisse > Klicken" als Variable den SiVArC-Ausdruck "StructureBlock.DB.SymbolicName&"_Activate".



3. Überschreiben Sie in der Bibliothek die bestehende Generiervorlage.

Ergebnis

Für jeden Aufruf des relevanten Funktionsbausteins wird eine Schaltfläche generiert, welche die Funktion auslösen und beenden kann. Die Variablen sind bereits für alle Instanzen verschaltet.

Ereignisse für WinCC Unified-Geräte

- Ereignisse unterstützen Skriptfunktionen für Bilder und Bildobjekte. Die Parameter von Skriptfunktionen müssen nicht zwingend konfiguriert werden. Bei nicht konfiguriertem Funktionsparameter ist der Parametertyp auf "Keine" voreingestellt.
- Sie können die "SetLanguage"-Parameter mittels "LCID-Parameter" mit Sprachenkennung oder Sprachennotation konfigurieren. Beispiel: Sprachenkennung "1033"; Sprachennotation "en-US".
- Bei Parametern vom Typ "Auswahltyp" müssen Sie die unterstützten Werte manuell konfigurieren. Andernfalls zeigt SiVArC eine Fehlermeldung an.
- Bei ungültigen Werten zeigt das System eine Warnmeldung an und am Engineering-System werden keine Änderungen vorgenommen.

6.2.19 Beispiel: Generiervorlage mit Skriptprojektierung erstellen

Beispielsszenario

In einem SiVArC-Musterprojekt sollen Temperaturmesswerte grundsätzlich zusätzlich zu den Werten in Fahrenheit auch in Grad Celsius ausgegeben werden können.

Umsetzungskonzept

Um ein zusätzliches Anzeigeobjekt entsprechend umzuschalten, wird eine Schaltfläche mit dem entsprechenden Skript in jedes Projekt generiert.

Wenn in einem Projekt die Umrechnung nicht erforderlich ist, deaktiviert der SiVArc-Projekteur die entsprechende Bildregel im nächsten Generat oder schränkt die Bildregel auf eine Auswahl von Bediengeräten ein.

Verfügbarkeit von Systemfunktionen und Skripten

Wenn Sie Skripte an Ereignisse anbinden, müssen diese Skripte auf jedem Zielgerät vorhanden sein. Wenn auf dem Zielgerät das projektierte Skript im Editor "Skripte" fehlt, wird das Anzeige- und Bedienobjekt ohne diese Skriptanbindung generiert.

SiVArc unterstützt das Parametrieren von Systemfunktionen und Skripten mit SiVArc-Ausdrücken an allen Ereignissen von Bildern und Bildobjekten. SiVArc unterstützt Systemfunktionen aus folgenden Kategorien:

- Berechnung
- Bitbearbeitung
- Bilder

SiVArc unterstützt für Bildbausteine eine begrenzte Auswahl an SiVArc-Ereignissen und Systemfunktionen. Eine Übersicht der unterstützten Systemfunktionen finden Sie im Abschnitt "Referenz".

Hinweis

Geräteabhängigkeit

Die Anzahl und Art von Ereignissen an einem Anzeige- und Bedienobjekt ist abhängig vom projektierten Bediengerät.

Weitere Informationen zur Geräteabhängigkeit von Ereignissen finden Sie in der Online-Hilfe des TIA Portals im Abschnitt "Mit Systemfunktionen und Runtime Scripting arbeiten" in der Referenz.

Voraussetzung

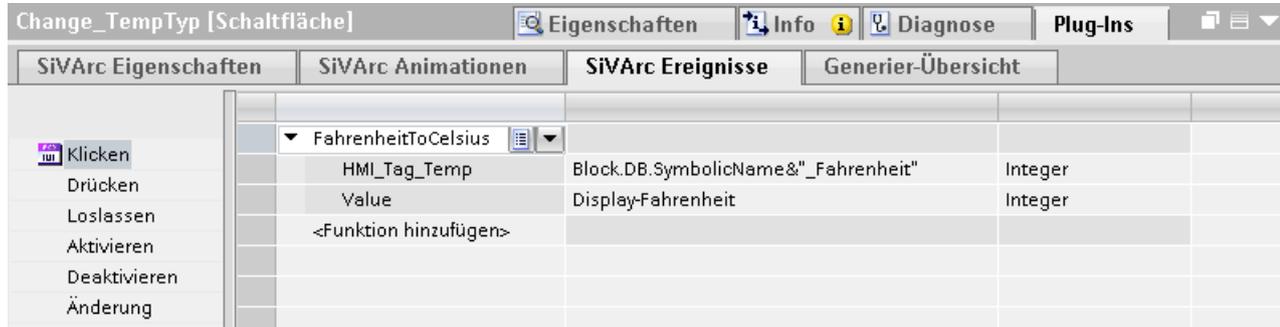
- Ein Skript "FahrenheitToCelsius" ist programmiert, das Fahrenheit in Celsius umrechnet und in einem E/A-Feld ausgibt.
- Das Skript verfügt über die Parameter "HMI_Tag_Temp" und "Value".
- Das Skript ist auf allen Zielgeräten angelegt.
- Eine Schaltfläche ist als Generiervorlage "Change_TempTyp" erstellt und über eine Bildregel mit der relevanten Funktion für die Temperaturmessung verknüpft.

Vorgehen

1. Öffnen Sie die Generiervorlage der Schaltfläche "Change_TempTyp" in WinCC.
2. Projektieren Sie im Inspektorfenster unter "Plug-Ins > SiVArc Ereignisse" an das Ereignis "Klicken" das Skript "FahrenheitToCelsius".

6.2 Erstellen von Generiervorlagen

3. Parametrieren Sie den Parameter "HMI_Tag_Temp" mit dem SiVArC-Ausdruck "Block.DB.SymbolicName&"_Fahrenheit".
4. Parametrieren Sie den Parameter "Value" mit dem Namen des Ausgabefeldes "Display_Fahrenheit".



5. Überschreiben Sie in der Bibliothek die bestehende Generiervorlage "Change_TempTyp".
6. Legen Sie für das E/A-Feld "Display_Fahrenheit" eine Bildregel an.

Ergebnis

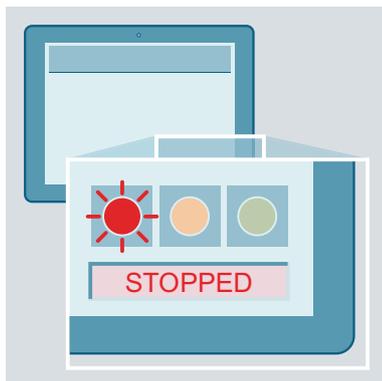
Bei der Generierung der Visualisierung wird die Schaltfläche "Change_TempTyp" und das E/A-Feld "Display_Fahrenheit" für jeden Aufruf des relevanten Funktionsbausteins erzeugt.

An die Schaltfläche "Change_TempTyp" ist das Skript "FahrenheitToCelsius" verschaltet. In Runtime wird der umgerechnete Wert aus der jeweiligen Variablen "Fahrenheit" jeder Instanz der Funktion im E/A-Feld "Display_Fahrenheit" angezeigt.

6.2.20 Beispiel: Generiervorlagen für Textlisten erstellen

Beispielszenario

Der Anlagenstatus wird mit einer Verkehrsampel angezeigt. Jeder Farbe ist ein Statustext zugeordnet, der in einem symbolischen E/A-Feld neben der Verkehrsampel angezeigt wird.



Ziel

Eine Generierungsvorlage für eine Textliste wird in der Bibliothek bereitgestellt. Die erforderlichen Textdefinitionen sind im Anwenderprogramm im Netzwerk beizubehalten.

Der Generierungsvorlage für die Textliste wird die dynamische Triggervariable zugeordnet. Der Datenbaustein für die relevante Funktion heißt "Plantsection1_DB". Der Name der Textliste sollte sich auf den ersten Teil des symbolischen Bausteinnamens beziehen: "Plantsection1_Textlist". Bei der Funktion "Split" wird "_DB" in einem SiVArc-Ausdruck für den Namen der Textliste verkürzt.

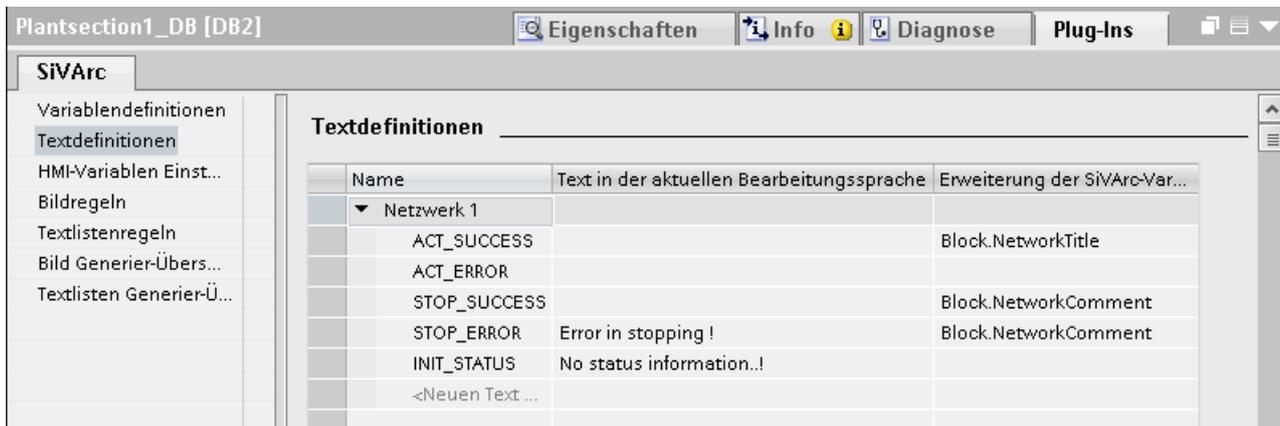
Voraussetzung

- Die Generierungsvorlage "Textlist_State" wurde in der Bibliothek gespeichert.
- Es wurde eine Textlistenregel erstellt, um die Generierungsvorlage "Textlist_State" mit dem Funktionsbaustein "Plantsection" zu verknüpfen.

Textlisteneinträge definieren

Um Textdefinitionen zu erstellen, gehen sie wie folgt vor:

1. Wählen Sie das gewünschte Netzwerk in dem Anwenderprogramm, für das die Textlisteneinträge definiert werden sollen.
2. Wählen Sie unter "Plug-ins > SiVArc" Kategorie "Textdefinitionen". Die folgenden Textdefinitionen werden im Anwenderprogramm im Netzwerk der ersten Instanz von "Plantsection" festgelegt:



3. Geben Sie unter "Name > Netzwerk" den Namen der Textliste ein.

4. Geben Sie unter "Text in der aktuellen Bearbeitungssprache" einen statischen Textlisteneintrag ein.
Wenn kein dynamischer Text angegeben ist, generiert SiVArC den statischen Text.
5. Geben Sie unter "Erweiterung der SiVArC-Variablen" einen SiVArC-Ausdruck ein, um einen Textlisteneintrag dynamisch zuzuweisen. Während der Generierung im Beispiel verwendet SiVArC den Netzwerktitel (`Block.NetworkTitle`) und den Netzwerkcommentar (`Block.NetworkComment`) des Funktionsbausteins, der in der Textlistenregel verknüpft ist.
Sie können die Textdefinition mit einem SiVArC-Ausdruck projektieren, der bei Platzierung in der Kopiervorlage in die Textlistenamen aufgelöst wird. Die Generierung von Textlisten mit den gleichen Namen wie die Standardtextlisten ist nicht möglich.

Vorgehensweise

Um eine Generierungsvorlage für eine Textliste zu erstellen, gehen sie wie folgt vor:

1. Öffnen Sie in der Bibliothek die "Textlist_State" Generierungsvorlage.
2. Wählen Sie als Textlistentyp "Area".
3. Öffnen Sie die Textlisteneinträge für die Textliste.
4. Kopieren Sie die Namen der Textdefinitionen aus dem Anwenderprogramm in die aufeinanderfolgenden Werte in der Spalte "Name".
5. Geben Sie Standardtextlisteneinträge ein.
6. Projektieren Sie im Inspektor-Fenster unter "Plug-Ins > SiVArC-Eigenschaften" den Namen der Textliste mit dem SiVArC-Ausdruck
"`Split(StructureBlock.DB.SymbolicName,"_",0)&"_Textlist"`.

Einträge in Textliste				
	Standard	Wert ▲	Name	Text
	<input type="radio"/>	0 - 1	ACT_SUCCESS	RUN
	<input type="radio"/>	2 - 3	ACT_ERROR	ERROR
	<input type="radio"/>	4 - 5	STOP_SUCCESS	STOP
	<input type="radio"/>	6 - 7	STOP_ERROR	STOP ERROR
	<input type="radio"/>	8 - 9	INIT_STATUS	START
		<Hinzufügen>		

Textlist_State [Textliste] Eigenschaften Info Diagnose Plug-Ins

SiVArC Eigenschaften

Name	Ausdruck des statischen Werts	Ausdruck der Variablen
▼ Allgemein		
Name	Split(StructureBlock.DB.SymbolicName,"_",0)&"_Textlist"	
Kommentar		
▼ Bausteinpara...	<input type="checkbox"/>	
Bausteinp...		
Eingang/A...	Eingabe	
<Neuen Date...		

7. Überschreiben Sie die vorhandene Generierungsvorlage in der Bibliothek.

Ergebnis

Während der Generierung wird die Textliste für die erste Instanz des Funktionsbausteins "Plantsection" erstellt. Der Name der Textliste "Plantsection1_Textlist" wird mit der "Split"-Funktion und dem Namen des Datenbausteins generiert.

Um auch die Textliste für alle anderen Verwendungen des Bausteins zu generieren, geben Sie die Textdefinitionen an allen Stellen ein, an denen der Baustein im Anwenderprogramm verwendet wird.

Wenn die Eingaben nicht ausgewertet werden können, wird eine Textliste auf Basis der SiVArc-Kopiervorlage erstellt.

Werden beim Generieren mehrere identische Namen für SiVArc-Texte erkannt, verwendet SiVArc den zuletzt erstellten SiVArc-Text.

Für Unified-Geräte können Sie die Variablen-tabelle für eine PLC mit mehrsprachigen Kommentaren projektieren und gleichzeitig eine Textliste als Kopiervorlage erstellen. Das System vergleicht die Variablen-tabelle mit den mehrsprachigen Kommentaren und mit der Variablen-tabelle im Funktionsbaustein. Die übereinstimmenden Einträge stehen in einer neu erstellten HMI-Textlistentabelle zur Verfügung.

Zur Erinnerung

- Für WinCC Unified-Geräte können Sie die Variablen-tabelle für eine PLC mit mehrsprachigen Kommentaren projektieren und gleichzeitig eine Textliste als Kopiervorlage erstellen; das System vergleicht die Variablen-tabelle mit den mehrsprachigen Kommentaren und der Variablen-tabelle im Funktionsbaustein. Die übereinstimmenden Einträge stehen in einer neu erstellten HMI-Textlistentabelle zur Verfügung.
- Wenn die PLC mit WinCC Unified-Geräten und Runtime Advanced-Geräten verbunden ist, können Sie einen Textlisteneintrag im WinCC Unified-Gerät erstellen und die Textliste in der Kopiervorlage hinzufügen. Wenn Sie bei der Erstellung von Textlistenregeln gerätespezifische Spalten für WinCC Unified-Geräte und Runtime Advanced-Geräte auswählen, werden die Textlisten bei der SiVArc-Generierung nur auf dem Unified-Gerät erstellt, während für das Runtime Advanced-Gerät ein Fehler angezeigt wird. Das gilt gleichermaßen für Kopierregeln.

6.2.21 Beispiel: Generiervorlage für eine Textliste für Bausteinparameter erstellen

Beispielsszenario

Der Anlagenzustand eines Förderbands soll im Bedienbild laufend ausgegeben werden.

Ziel

Ein E/A-Feld wird von SiVArc generiert, das mit einer Textliste verschaltet ist, die ihre Einträge direkt vom Bausteinausgang "State_A" des Funktionsbausteins "Conveyor" bezieht.

Voraussetzung

- Die Generiervorlage "Textlist_Parameter" ist in der Bibliothek abgelegt.
- Eine Textlistenregel ist erstellt, welche die Generiervorlage "Textlist_Parameter" mit dem Funktionsbaustein "Conveyor" verschaltet.
- In den Kommentaren der Variablen am Bausteinausgang "State_A" sind die Förderbandstatus enthalten:
 - OFF
 - ERROR
 - STOP
 - RUN

Vorgehen

Um eine Generiervorlage für eine Textliste für einen Bausteinparameter zu erstellen, gehen Sie folgendermaßen vor:

1. Öffnen Sie die Generiervorlage "Textlist_Parameter" aus der Bibliothek.
2. Aktivieren Sie in den SiVArc-Eigenschaften "Blockparameter und relevante PLC-Variablen verwenden".
3. Tragen Sie den Parameternamen "State_A" und E/A-Typ "Ausgabe" ein.
Um mehrere Parameter auszuwählen, verwenden Sie einen regulären Ausdruck mit Asterisken. Das System wertet dann alle Parameter aus, in dessen Namen die Zeichenkette wie angegeben enthalten ist.
4. Wählen Sie den Datentypen "BOOL" und die Anzahl der Variablen "4", die für die Textlistengenerierung verwendet werden.
Wenn Sie z. B. die Anzahl "17" wählen, werden die ersten 17 Variablen verarbeitet. Wenn es nur 15 gibt, werden nur die ersten 15 verarbeitet.
5. Überschreiben Sie in der Bibliothek die bestehende Generiervorlage.

Ergebnis

Die Variablen vom projektierten Datentyp werden von der Generierung erfasst und ausgewertet. Für vier dieser Variablen wird jeweils ein Textlisteneintrag wie folgt erstellt:

- Die Textlisteneinträge entsprechen dem jeweiligen Kommentar der Variablen.
- Die Namen der Textlisteneinträge setzen sich zusammen aus dem Parameternamen, dem Datentyp des Parameters und einer fortlaufenden Nummer, z. B. State_A_Bit_1, State_A_Bit_2, etc.

Wenn der Variablenname in der Symboltabelle nicht enthalten ist, wird die projektierte Anzahl von Textlisteneinträgen mit Wertzuordnung und Namen erstellt. Die Namen der Texteinträge sind dann vom Parameter abgeleitet. Ergänzen Sie in diesem Fall die gewünschten Textlisteneinträge in den Kommentaren der Variablen und generieren Sie ein weiteres Mal. Wenn Sie die Texteinträge manuell eintragen, bleiben die Texte nur bis zur nächsten Generierung erhalten.

6.2.22 Beispiel: Pop-up-Bilder und deren Verwendung generieren

Beispielsszenario

Auf einem Bedienbild reicht der Platz gerade noch aus für alle zur Prozesssteuerung benötigten Anzeige- und Bedienobjekte. Daher wird der Dialog zur Sprachumschaltung in ein Pop-up-Bild ausgelagert.

Umsetzungskonzept

An eine Schaltfläche wird der Aufruf eines Pop-up-Bildes projiziert, das die Einstellungen für den Sprachwechsel enthält.

Verwendung von Pop-up-Bildern in SiVArc

Sie verwenden Pop-up-Bilder mit SiVArc wie andere WinCC-Bilder. Um einem Pop-up-Bild ein eigenes Positionierungsschema zu geben, verwenden Sie Positionierungsschemata, die auf Basis eines Pop-up-Bildes erstellt sind.

Um Anzeige- und Bedienobjekte in ein Pop-up-Bild zu generieren, verwenden Sie das Pop-up-Bild in einer Bildregel als "Kopiervorlage eines Bildes".

Voraussetzung

- Folgende Generiervorlagen sind in der Bibliothek angelegt:
 - Pop-up-Bild "PopUp_ChangeLang"
 - Schaltfläche "Button_PopUp_ChangeLang"
 - Startbild "StartScreen"
- Ein Pop-up-Bild ist als Positionierungsschema "PopUp_Pos_ChangeLang" in der Bibliothek angelegt.
- Bildregeln sind für folgende HMI-Objekte angelegt:
 - Schaltfläche "Button_PopUp_ChangeLang"
 - Pop-up-Bild "Button_PopUp_ChangeLang"

Generiervorlage für das Pop-up-Bild erstellen

Um eine Generiervorlage für den Aufruf eines Pop-up-Bildes zu erstellen, gehen Sie folgendermaßen vor:

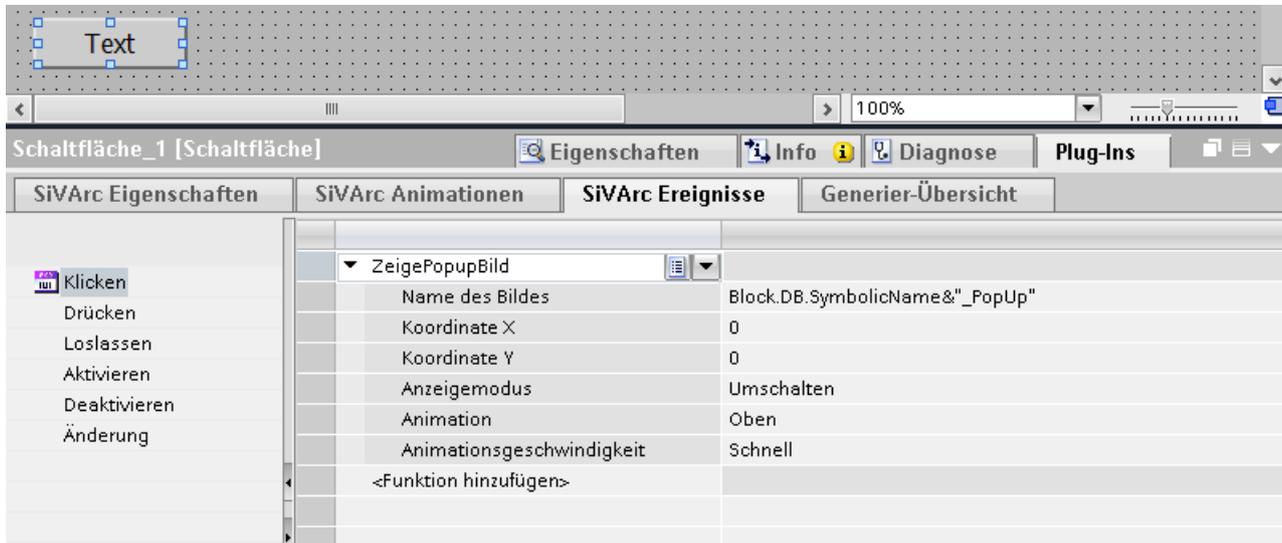
1. Öffnen Sie die Generiervorlage des Pop-up-Bildes "PopUp_ChangeLang" aus der Bibliothek.
2. Parametrieren Sie im Inspektorfenster unter "Plug-Ins > SiVArC-Eigenschaften > Allgemein" die folgenden SiVArC-Eigenschaften:
 - Um einen eindeutigen Bildnamen zu generieren, geben Sie unter "Name" einen SiVArC-Ausdruck oder eine Zeichenkette ein. Integrieren Sie z. B. für den Namen des Pop-up-Bildes den Namen des aufgerufenen Programmblocks mit `"Block.DB.SymbolicName&"_PopUp"`.
 - Wenn Sie das generierte Bild in einer Gruppe oder in der Anlagenstruktur ablegen wollen, geben Sie unter "Bildgruppe" einen SiVArC-Ausdruck ein.
 - Wählen Sie "Fest" als Modus des Positionierungsschemas und das Positionierungsschema "PopUp_Pos_ChangeLang".
3. Überschreiben Sie in der Bibliothek die bestehende Generiervorlage.

Generiervorlage für die aufrufende Schaltfläche erstellen

Um eine Generiervorlage für den Aufruf eines Pop-up-Bildes zu erstellen, gehen Sie folgendermaßen vor:

1. Öffnen Sie die Generiervorlage der aufrufenden Schaltfläche "Button_PopUp_ChangeLang" aus der Bibliothek.
2. Parametrieren Sie im Inspektorfenster unter "Plug-Ins > SiVArC-Eigenschaften" die gewünschten SiVArC-Eigenschaften. Integrieren Sie z. B. für den Namen der Schaltfläche den Namen des aufgerufenen Programmblocks mit `"Block.SymbolicName&"_ButtonPopUp"`

3. Parametrieren Sie im Inspektorfenster unter "Plug-Ins >SiVArc-Ereignisse" z. B. an das Ereignis "Klicken" die Systemfunktion "ZeigePopupBild".
 - Vergeben Sie für den Parameter "Name des Bildes" den SiVArc-Ausdruck, den Sie in der Generiervorlage des Pop-up-Bildes unter Plug-Ins >SiVArc-Eigenschaften > Allgemein > Name" konfiguriert haben: "Block.DB.SymbolicName&"_PopUp"
 - Parametrieren Sie die Koordination zur Anzeigeposition des Pop-up-Bildes mit einem Integer-Wert.
 - Wählen Sie die Anzeigewerte.



4. Überschreiben Sie in der Bibliothek die bestehende Generiervorlage.

Ergebnis

In den Bildregeln sind das Startbild der Anlage, die Schaltfläche für die Sprachumstellung und das Pop-up-Bild und der zentrale Funktionsbaustein miteinander verknüpft. Nach dem Generieren wird in das Startbild des Anlagenbereichs eine zusätzliche Schaltfläche generiert, die ein Pop-up-Fenster für die Sprachumschaltung aufruft.

6.2.23 Beispiel: Bildbausteine mit Animationen generieren

Beispielszenario

Ein Fließband einer Fertigungsanlage ist für schwere Lasten ausgelegt und wird nur für spezielle Verpackungsformate verwendet. Die Drehzahlregelung der zwei Achsen soll daher nur dann auf dem Anlagenbild angezeigt werden, wenn die Fertigungslinie in Betrieb ist.

Umsetzungskonzept

Die Visualisierung der Drehzahlregelung wird in einem Bildbaustein vorbereitet. Der Bildbaustein wird im Projekt als Generierungsvorlage zur Regelung aller drehzahlgeregelten Achsen verwendet. Der Visualisierungstechniker erstellt eine neue Generierungsvorlage mit einer Sichtbarkeitsanimation auf Grundlage des Bildbausteins.

In den Bildregeln werden Bedingungen zur Regelung verwendet, wann ein Bildbaustein mit Animation generiert wird.

Animierte Bildbausteine in SiVArC

SiVArC unterstützt die folgenden Animationen für Bildbausteine:

- Sichtbarkeit
- Bedienbarkeit
- Gestaltung

Zum Generieren von Animationen für Bildbausteine mit SiVArC konfigurieren Sie dynamische Eigenschaften für die Animation im Bildbausteintyp, der als Generierungsvorlage dient.

Voraussetzung

- Die Generierungsvorlage "fpSpeedAxis" des Bildbausteintyps wird in der Bibliothek gespeichert.
- Die Variable "Conveyor_HeavyLoad_Instance01_ReActivate" ist in der Bausteinschnittstelle des relevanten Funktionsbausteins enthalten.
- Eine Variable mit der Endung "_ReActivate" wird nur für den Schwerlastbetrieb verwendet.

Vorgehensweise

Zum Generieren von Bildbausteinen mit Animationen befolgen Sie diese Schritte:

1. Öffnen Sie die Generierungsvorlage des Bildbausteintyps "fpSpeedAxis".
2. Erstellen Sie in der Liste "Schnittstelle" für den Bildbausteintyp eine Eigenschaft mit dem Namen "Sichtbar" vom Datentyp BOOL.
3. Konfigurieren Sie die Animation "Sichtbar" in den WinCC-Animationen aller Objekte, die im Bildbausteintyp enthalten sind. Verwenden Sie dabei in jedem Fall die Schnittstelleneigenschaft "Sichtbar" als Prozessvariable.

- Konfigurieren Sie in den SiVArc-Eigenschaften des Bildbausteintyps die Schnittstelleneigenschaft Sichtbar mit dem SiVArc-Ausdruck "Block.DB.SymbolicName&"_ReActivate".

Speed axis

Set value

Enabled OFF

Actual value

-500 0 500

100%

Eigenschaften Ereignisse Variablen Skripte Textlisten Grafiklisten Texte

Enthaltene Objekte

Name Dynamis...

▶ Text field_2

▶ Text field_3

▶ Text field_4

▶ Text field_5

▶ Text field_6

ame Typ

Properties_Faceplate

▶ typeSpeedAxis typeSpeedAxi...

Visible Bool

fpSpeedAxis [Bild] [fpSpeedAxis V 2.0.1... Eigenschaften Info Diagnose Plug-Ins

SiVArc Eigenschaften		SiVArc Animationen	SiVArc Ereignisse
Name	Ausdruck des statischen ...	Ausdruck der Variablen	
▼ Verschiedenes			
Name			
Ebene			
▼ Position			
X - Position			
Y - Position			
▼ Eigenschaftenschnittstelle			
▼ Properties_Faceplate			
typeSpeedAxis			
Visible		Block.DB.SymbolicName&"_ReActivate"	

- Erstellen Sie eine neue Version des Bildbausteintyps als Generierungsvorlage.
- Verwenden Sie den Bildbausteintyp und den relevanten Programmbaustein in einer Bildregel.

Ergebnis

Wenn Sie eine Bildregel mit dieser Generierungsvorlage erstellt haben, wird der SiVArc-Ausdruck während der Generierung ausgewertet. Eine externe von SiVArc generierte Variable wird der Eigenschaft jeder generierten Instanz des Bildbausteintyps zugewiesen.

Im Beispiel wird die Animation nur auf dem Bildbaustein für das Schwerlast-Förderband verschaltet, da eine Variable mit der Endung "_ReActivate" nur dort vorhanden ist.

Zusammenführen von Schnittstelleneigenschaften für Bildbausteine

Bei der Konfiguration eines Bildbausteins mit Schnittstelleneigenschaften von Bildobjekten können Sie die für statische und dynamische Werte vorgenommenen Änderungen beibehalten. Mit dem Kontrollkästchen **Manuelle Änderungen beibehalten** im Editor **Plug-Ins** für Bildbausteine können Sie die für Schnittstelleneigenschaften vorgenommenen Änderungen für die aktuelle Bildbausteininstanz beibehalten.

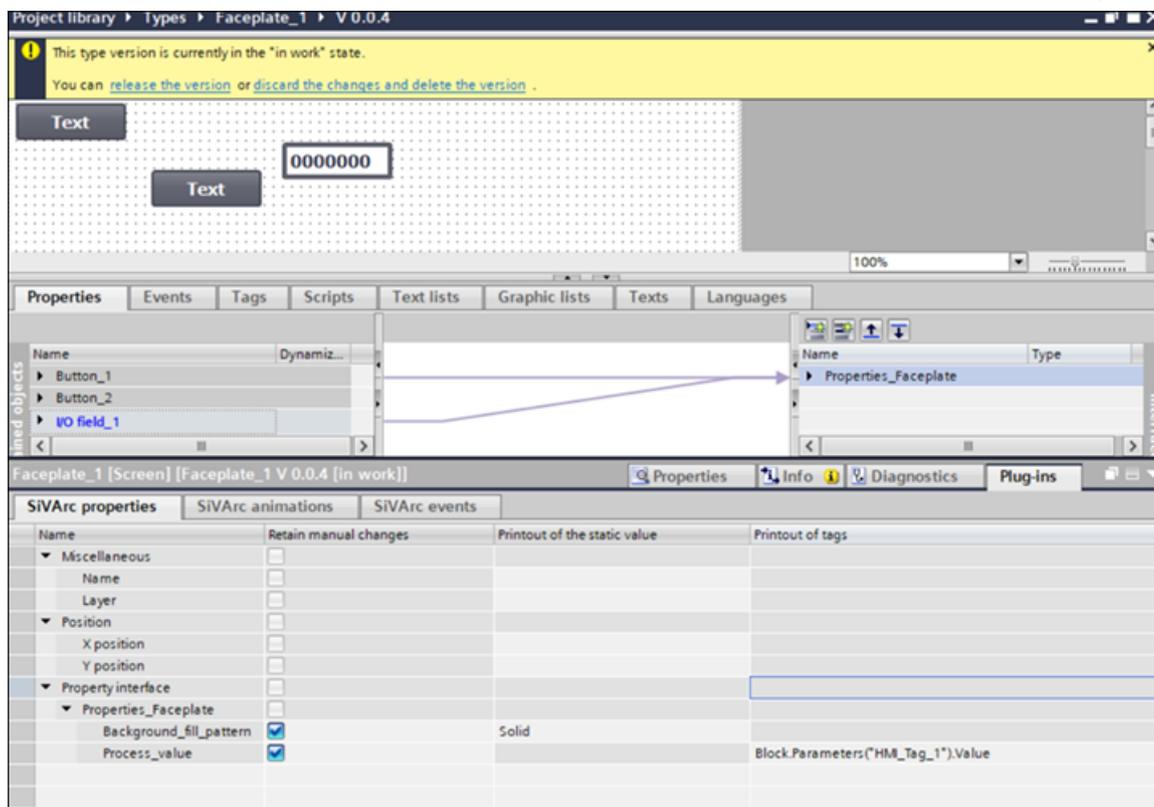
Wenn die Bildbausteininstanz nach der SiVArC-Generierung erstellt wird und das Kontrollkästchen **Manuelle Änderungen beibehalten** ausgewählt ist:

- Wenn ein Anwender den Eigenschaftswert manuell in der generierten Bildbausteininstanz bei der SiVArC-Generierung ändert, werden die Werte beibehalten.

Vorgehensweise

Zum Generieren von SiVArC mit Hilfe von Schnittstelleneigenschaften von Bildbausteinen in Bildern führen Sie die folgenden Schritte aus:

1. Konfigurieren Sie die Schnittstelleneigenschaften eines Bildbausteins mit Bildobjekten.
2. Wählen Sie das Kontrollkästchen "Manuelle Änderungen beibehalten" und konfigurieren Sie die Spalte mit statischen und die Spalte "Ausdruck der Variablen" mit dynamischen Werten.



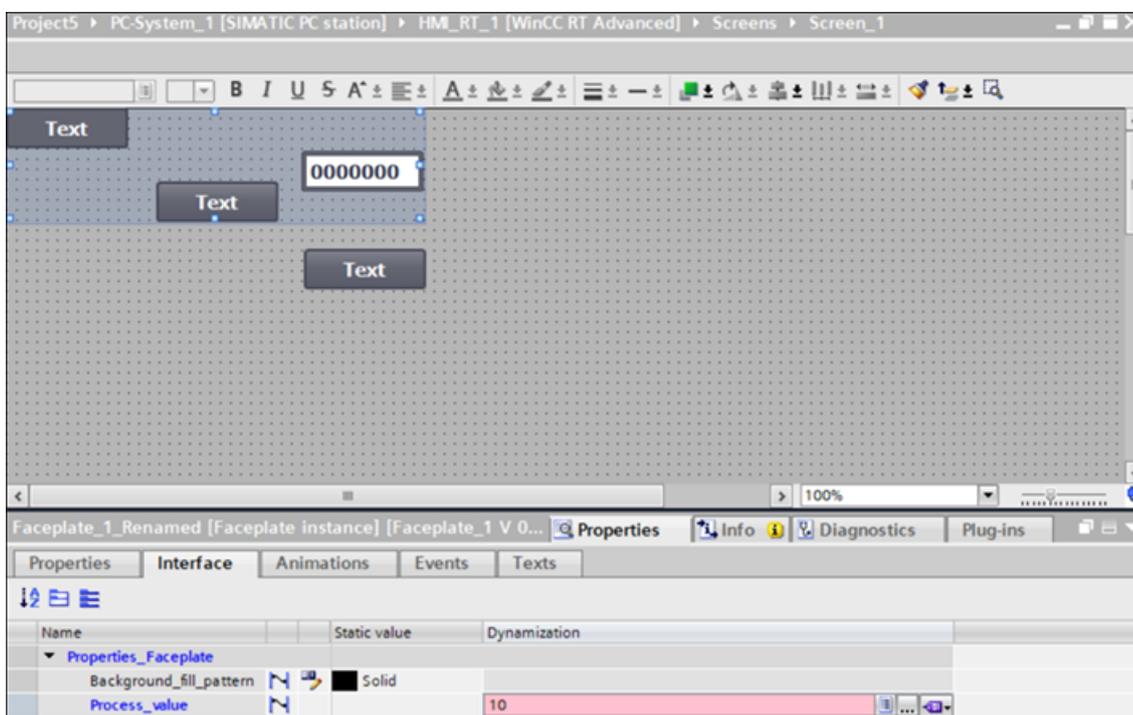
3. Erstellen Sie eine Bildregel mit einem Bildobjekt als Bildbaustein und einem Bild als "Kopiervorlage eines Bilds". Weitere Einzelheiten zum Konfigurieren von Bildregeln finden Sie unter Erstellen von Regeln (Seite 43).

4. Generieren Sie die SiVArc-Visualisierung.
5. Das generierte Bildobjekt enthält die Schnittstelleneigenschaften des Bildbausteins. Weitere Informationen über den Generierungsvorgang finden Sie unter Generierung (Seite 43).

Bearbeiten der Instanz des Bildbausteineigenschaften

Nach der SiVArc-Generierung können Sie die Instanz eines Bildbausteins bearbeiten, der auf dem Bild generiert wird. Bei ausgewähltem Kontrollkästchen "Manuelle Änderungen beibehalten" bleiben alle nicht gespeicherten Änderungen an der Schnittstelleneigenschaft des Bildbausteins, der generiert wird, für die nächste Generierung unverändert.

Berücksichtigen Sie das folgende Szenario, bei dem die Schnittstelleneigenschaft des Bildbausteins den "Prozesswert" 10 enthält, der sich aus der ersten Generierung ergibt. Nach der Generierung wird die Schnittstelleneigenschaft des Bildbausteins bei im Bibliotheksfenster des Bildbausteins markiertem Kontrollkästchen "Manuelle Änderungen beibehalten" geändert, und die SiVArc-Generierung wird ausgelöst. Bei allen nachfolgenden Generierungen wird der "Prozesswert" bei Änderung der Schnittstelleneigenschaft entsprechend der manuell vorgenommenen Änderungen geändert: Das gilt für statische Eigenschaften. Beispiel: Background_fill-pattern.



Ausnahmefälle:

- Manuelle Änderungen sind nur anwendbar, wenn ähnliche Versionen in der Bildbausteinbibliothek und dem generierten Bildbaustein verfügbar sind.
- Wird der Bildbaustein gelöscht, ist "Manuelle Änderungen beibehalten" nicht anwendbar und der Bildbaustein wird aus der letzten verfügbaren Version generiert.

- Statische Schnittstelleneigenschaften werden für erweiterte Bildbausteine nicht unterstützt, wenn dynamische Eigenschaften unterstützt werden.
- Bei erweiterten Bildbausteinen werden für einige der Eigenschaften, die SiVArC nicht beibehalten kann, Werte vom Engineering-System vorgegeben.
- Für dynamische Werte sind HMI-Variablen die unterstützten Eingabetypen.

Siehe auch

Editor "Bildregeln" (Seite 27)

Editor "Generiermatrix" (Seite 36)

Generierübersicht (Seite 43)

6.2.24 Beispiel: Animation "Position" für Bildbausteine generieren

Beispielsszenario

In einer Leiterplattenfabrik werden in der Teilanlage "Packaging" die produzierten Leiterplatten in Kisten abgepackt und an Transportwagen übergeben. Dieser Vorgang soll am Bediengerät animiert dargestellt werden.

Umsetzungskonzept

Die abgepackten Kisten sind als Bildbaustein-Generiervorlagen in der Bibliothek hinterlegt. Zur Darstellung der horizontalen Bewegung der fertig gepackten Kiste auf einen Transportwagen werden die Bildbausteine mit der Animation "Position" parametrisiert. Die Positionswerte für die horizontale Bewegung erhält der Bildbaustein aus der Steuerung.

Animation "Position" für Bildbausteine in SiVArC

Bildbausteine unterstützen für RT Professional die Animation "Position".

Um mit SiVArC-Animationen für Bildbausteine zu generieren, projektieren Sie im Bildbaustein-Typ, der als Generiervorlage dient, dynamische Eigenschaften für die Animation.

Voraussetzung

- Die Generiervorlage des Bildbaustein-Typs "Plate_Box_Ready" ist in der Bibliothek abgelegt.
- Der Funktionsbaustein "Packaging" enthält den Input-Parameter "XPosition" vom Datentyp INT.
- Die Werte des Parameters "XPosition" sind im zugehörigen Datenbaustein gespeichert.
- Die Ziel-Bediengeräte für die Generierung der Visualisierung der Verpackungsanlage haben die gleiche Bildschirmauflösung.

Vorgehen

Um eine Animation "Position" für einen Bildbaustein zu generieren, gehen Sie folgendermaßen vor:

1. Öffnen Sie den Bildbaustein-Typ "Plate_Box_Ready" aus der Bibliothek.
2. Erstellen Sie in der Liste "Schnittstelle" für den Bildbaustein-Typ die Eigenschaft "IFace_XPosition" vom Datentyp INT für eine horizontale Animation.
3. Projektieren Sie in den WinCC-Animationen aller im Bildbaustein-Typ enthaltenen Objekte eine neue Variablenverbindung. Verbinden Sie die Variable mit der Eigenschaft "X position".
4. Projektieren Sie die an die Eigenschaft "X position" gebundene Variable mit der Schnittstelleneigenschaft "IFace_XPosition".
5. Projektieren Sie in den SiVArc-Eigenschaften des Bildbaustein-Typs die Schnittstelleneigenschaft "IFace_XPosition" mit dem SiVArc-Ausdruck "Block.DB.SymbolicName&"_XPosition".
6. Erstellen Sie eine neue Bildbaustein-Typversion.
7. Verwenden Sie den Bildbaustein-Typ und den relevanten Programmbaustein in einer Bildregel.

Ergebnis

Nach dem Generieren sind alle erzeugten Instanzen des Bildbaustein-Typs "Plate_Box_Ready" mit einer Animation parametrisiert. In Runtime folgt die Position des Bildbausteins dem Positionswert der verschalteten Variablen, z. B. "Block_1_DB_XPosition".

6.2.25 Beispiel: Erstellung von Generiervorlagen für Kurvenanzeigen

Beispielszenario

Bei einer Simulation von Hydraulikkomponenten in einer Fertigungsanlage wird der Temperaturparameter in Bezug auf den Zeitparameter gemessen. Die Werte werden über ein Bediengerät als Kurven angezeigt.

Umsetzungskonzept

Kurvenanzeigen dienen dazu, Variablenwerte von bestimmten Parametern grafisch in Kurvenform darzustellen. SiVArc automatisiert die Generierung von Variablenwerten und zeigt die Variablenwerte im Bediengerät als Kurven an. Bei der Simulation der Hydraulikkomponenten wird ein Temperaturanstieg oder -rückgang über einen bestimmten Zeitraum gemessen und im Bediengerät in Kurvenform angezeigt. Mithilfe der im Bediengerät angezeigten Kurvendaten wird der Status der Hydraulikkomponenten überwacht.

Die Eigenschaften der Kurvenanzeige können in TIA und auch in SiVArc projiziert werden. Eine ausführliche Beschreibung, wie Sie die Eigenschaften der Kurvenanzeige projektieren, finden Sie in der Online-Hilfe des TIA Portals.

In SiVArC erfolgt die Projektierung der Eigenschaften der Kurvenanzeige unter "SiVArC-Eigenschaften > Plug-ins".

Je nach verwendetem Gerät können verschiedene Eigenschaften der Kurvenanzeige projiziert werden. Weitere Informationen zu den Eigenschaften der Kurvenanzeige finden Sie in der Online-Hilfe des TIA Portals.

Voraussetzungen

- Ein Bild mit einer Kurvenanzeige namens "TrendView1" wird für ein Bediengerät erstellt.

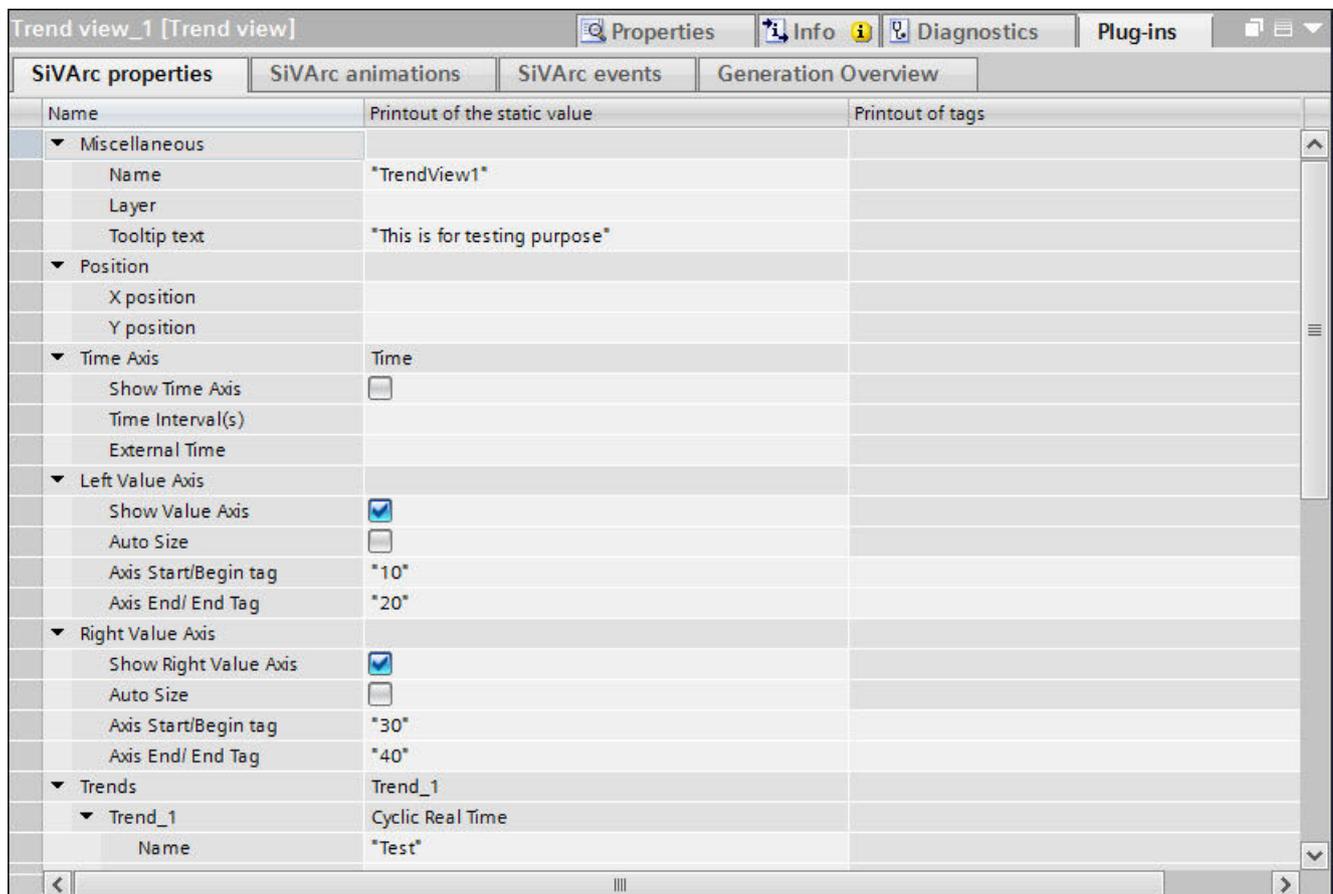
Beispiel: Erstellung von Generiervorlagen für Kurvenanzeigen

Das folgende Beispiel zeigt, wie eine Kurvenanzeige in SiVArC für RT Advanced projiziert wird.

1. Projektieren Sie unter "SiVArC-Eigenschaften > Plug-Ins" die Eigenschaften von TrendView1 wie "Ebene" ("Layer") und "Position".

Hinweis

In RT Advanced können Sie außerdem die Eigenschaft "Tooltip-Text" projektieren.



2. Wenn Sie die Kurvenanzeige mit Zeitachse projektieren wollen, zeigen Sie unter "Plug-Ins" die Eigenschaften von "Zeitachse" an und aktivieren das Kontrollkästchen "Zeitachse anzeigen". Dadurch wird die Kurvenanzeige mit Zeitachse angezeigt.

3. Zum Projektieren der Wertachse erweitern Sie die Eigenschaften "Wertachse links" und "Wertachse rechts".
 - Wenn die Wertachse in der Kurvenanzeige enthalten sein soll, aktivieren Sie das Kontrollkästchen "Wertachse anzeigen".
 - Wenn während der Generierung von Objekten die Achsenwerte automatisch eingefügt werden sollen, aktivieren Sie das Kontrollkästchen "Automatisch anpassen".
 - Geben Sie den Variablenwert für "Achse Anfang/Beginn Variable" und für "Achse Ende/Ende Variable" an.

Hinweis

In RT Advanced werden die folgenden "Achsenmodi" unterstützt.

- Punkte: Anzahl der Werte anzeigen
 - Variable/Konstante: Anzahl der Werte anzeigen
 - Zeit: Zeigt die externe Uhrzeit an.
-

4. Um Kurvenwerte zu projektieren, erweitern Sie die Eigenschaften von "Kurve".
 - Geben Sie unter "Name" einen Namen für den betreffenden Trend ein.
 - Geben Sie unter "Datenquelle Prozesswerte" einen gültigen Wert für Prozesswerte ein.
5. Projektieren Sie Bildregeln für ein bestimmtes Bild, das die Kurve enthält. Weitere Informationen zum Konfigurieren von Bildregeln finden Sie im Abschnitt "Bildregel für Generierung eines Bildobjekts definieren".
6. Generieren Sie die Visualisierung. Weitere Informationen über den Generierungsvorgang finden Sie im Abschnitt "Visualisierung generieren".

Ergebnis

Nach der Generierung werden die Variablenwerte für die Temperatur im Zeitverlauf als Kurven im Bediengerät grafisch dargestellt.

6.2.26 Beispiel: Hardwarekonfiguration für Bilder

Einleitung

Sie können am PLC angeschlossene IO-Devices konfigurieren und Bilder generieren (über Ausdrücke), die zu IO-Gerätenamen, Invariant-Typ und Artikelnummer führen. Am PLC angeschlossene IO-Devices können einen ähnlichen oder unterschiedlichen Invariant-Typ haben. Wenn ein IO-Device einem PLC zugeordnet wird und Sie versuchen, SiVArc-Regeln zu konfigurieren, wird in der Spalte "Regel-Trigger" der für den angeschlossenen PLC verfügbare Invariant-Typ angezeigt. Die SiVArc-Generierung führt zum Erstellen von Bildern für einzelne oder mehrere Instanzen von IO-Gerätenamen, Invariant-Typ und Artikelnummer.

Beispielszenario

Stellen Sie sich die Programmierung der Teilemontage in einem Automobilfertigungswerk vor, wobei der PLC mit ähnlichen oder unterschiedlichen Invariant-Typen am IO-Device angeschlossen ist.

Ziel

Im Automobilfertigungswerk dienen mehrere HMI-Bilder dazu, die Teilemontage zu visualisieren. Sie konfigurieren SiVArc-Eigenschaften mit Ausdrücken, die zur Generierung von Bildern mit Namen von IO-Devices führen, die dem PLC zugeordnet sind. Im Ergebnis profitieren Sie von erhöhter Produktivität und Skalierbarkeit über mehrere Bediengeräte hinweg.

Voraussetzung

- Einem einzelnen oder mehreren IO-Devices (nur ET200 Profinet) mit ähnlichen oder unterschiedlichen Invariant-Typen zugeordneter PLC
- An PLC angeschlossenes Bediengerät
- SiVArc-Ausdruck, der IO-Gerätename, Invariant-Typ, Artikelnummer unterstützt

Vorgehensweise

Um HMI-Bilder mit Hilfe von IO-Devices zu generieren, gehen Sie wie folgt vor:

1. Ordnen Sie einen PLC mit einem oder mehreren IO-Devices mit einem Invariant-Typ Ihrer Wahl zu.
2. Fügen Sie ein Bild hinzu. Konfigurieren Sie im Inspektorfenster unter "Plug-ins > SiVArc-Eigenschaften > Name" die Bildeigenschaft mit einem Ausdruck, was zu IO-Gerätename, Invariant-Typ und Artikelnummer führt. Beispiel: "Device.Name". Der aufgelöste Name des Ausdrucks wird im Zielbild angezeigt.
3. Fügen Sie das Bild in den Ordner der Kopiervorlagen mit dem Bildobjekt ein.
4. Klicken Sie unter "SiVArc > Bildregeln" auf "Neue Regel erstellen".
5. Suchen Sie in der Spalte "Regel-Trigger" nach einem Invariant-Ordner Ihrer Wahl und dem Bildobjekt aus dem Ordner "Kopiervorlagen".
6. Generieren Sie auf dem Bediengerät die SiVArc-Visualisierung.

Hinweis

- Für IO-Devices unter "Kopiervorlagen" > "Typen" werden keine Bilder generiert.
 - Bei Bildregeln mit Bausteinen und Invariant-Typen werden zunächst Bildregeln mit Bausteinen ausgeführt, gefolgt von den Invariant-Typen.
-

Ergebnis

Ein HMI-Bild "<<Gerätename>>" wird generiert. Beispiel: IO-Device_1.

6.2.27 Beispiel: Hardwarekonfiguration für Meldungen

Einleitung

Sie können an den PLC angeschlossene IO-Devices konfigurieren und Meldungen, Meldegruppen und Meldeklassen generieren (über Ausdrücke), die zu IO-Gerätenamen, Invariant-Typ und Artikelnummer führen. Am PLC angeschlossene IO-Devices können einen ähnlichen oder unterschiedlichen Invariant-Typ haben. Wenn ein IO-Device einem PLC zugeordnet wird und Sie versuchen, SiVArc-Regeln zu konfigurieren, wird in der Spalte "Regel-Trigger" der für den angeschlossenen PLC verfügbare Invariant-Typ angezeigt. Die SiVArc-Generierung führt zum Erstellen von Meldungen für einzelne oder mehrere Instanzen von IO-Gerätenamen, Invariant-Typ und Artikelnummer.

Beispielszenario

Stellen Sie sich die Überwachung des Status einer Fertigungsanlage vor. Eine Meldung mit einem spezifischen Gerätenamen zeigt an, dass das Gerät Aufmerksamkeit benötigt.

Ziel

Im Automobilfertigungswerk dienen mehrere HMI-Meldungen dazu, die Status der Teilemontage zu überwachen. Sie konfigurieren SiVArc-Eigenschaften mit Ausdrücken, die zur Generierung von Meldungen mit Namen von IO-Devices führen, die dem PLC zugeordnet sind.

Voraussetzung

- Einem einzelnen oder mehreren IO-Devices (nur ET200 Profinet) mit ähnlichen oder unterschiedlichen Invariant-Typen zugeordneter PLC
- An PLC angeschlossenes Bediengerät
- SiVArc-Ausdruck, der IO-Gerätenamen, Invariant-Typ, Artikelnummer unterstützt

Vorgehensweise

Um HMI-Meldungen mit Hilfe von IO-Devices zu generieren, gehen Sie wie folgt vor:

1. Ordnen Sie einen PLC mit einem oder mehreren IO-Devices mit einem Invariant-Typ Ihrer Wahl zu.
2. Wählen Sie "HMI-Meldungen" und klicken Sie auf "Neu hinzufügen", um eine neue Meldung hinzuzufügen.
3. Konfigurieren Sie im Inspektorfenster unter "Plug-ins > SiVArc Eigenschaften > Name" die Meldungsnameeigenschaft mit einem Ausdruck, was zu IO-Gerätenamen, Invariant-Typ und Artikelnummer führt. Beispiel: "Device.Name". Der aufgelöste Name des Ausdrucks wird in der Zielmeldung angezeigt.
4. Fügen Sie die Meldung in den Ordner "Kopiervorlagen" ein.
5. Klicken Sie unter "SiVArc > Melderegeln" auf "Neue Regel erstellen".

- Suchen Sie in der Spalte "Regel-Trigger" nach einem Invariant-Ordner Ihrer Wahl und dem Meldeobjekt aus dem Ordner "Kopiervorlagen".
- Generieren Sie auf dem Bediengerät die SiVArc-Visualisierung.

Hinweis

- Für IO-Devices unter "Kopiervorlagen" > "Typen" werden keine Meldungen generiert.
 - Bei Melderegeln mit Bausteinen und Invariant-Typen werden zunächst Melderegeln mit Bausteinen ausgeführt, gefolgt von den Invariant-Typen.
-

Ergebnis

Eine HMI-Meldung "<<Gerätename>>" wird generiert. Beispiel: IO-Device_1.

6.2.28 Generiervorlage für ein Bild erstellen

Voraussetzung

- WinCC-Projekt ist geöffnet.

Vorgehen

Um eine Generiervorlage für ein Bild zu erstellen, gehen Sie folgendermaßen vor:

- Legen Sie ein neues Bild an.

Hinweis

Vergeben Sie einen aussagekräftigen Namen. Ein eindeutiger Name erleichtert die spätere Weiterarbeit, da der Bildname als Name für die Generiervorlage verwendet wird.

- Konfigurieren Sie die Eigenschaften des Bildes und fügen Sie bei Bedarf die benötigten Bildobjekte ein.
- Parametrieren Sie im Inspektorfenster unter "Plug-Ins > SiVArc-Eigenschaften > Allgemein" die gewünschten Eigenschaften:
 - Um einen eindeutigen Bildnamen zu generieren, geben Sie unter "Name" einen SiVArc-Ausdruck oder eine Zeichenkette ein.
 - Wenn Sie das generierte Bild in einer Gruppe oder in der Anlagenstruktur ablegen wollen, geben Sie unter "Bildgruppe" einen SiVArc-Ausdruck ein.
 - Projektieren Sie bei Bedarf Überlaufbilder.

4. Um eine Kopiervorlage zu erzeugen, legen Sie das Bild in einer Bibliothek unter "Kopiervorlagen" ab.
 5. Um einen Bildtyp zu erzeugen, legen Sie das Bild in einer Bibliothek unter "Typen" ab und vergeben Sie den Typnamen.
-

Hinweis**SiVArc-Eigenschaften eines Bildtyps**

Im Bildtyp stehen Ihnen weniger SiVArc-Eigenschaften zur Verfügung als in der Kopiervorlage eines Bildes.

Ergebnis

Die Generiervorlage für ein Bild ist erstellt.

Siehe auch

Bildregel anlegen (Seite 188)

Visualisierung generieren (Seite 215)

SiVArc-Objekteigenschaften (Seite 259)

Ablagestrategien für generierte Objekte (Seite 126)

6.3 Regeln erstellen

6.3.1 SiVArc-Regeln

Definition

SiVArc-Regeln legen fest, wie HMI-Objekte während der Generierung verarbeitet werden.

Die verschiedenen SiVArc-Regeln definieren unterschiedliche Generieraufgaben:

- Bild- und Textlistenregeln verknüpfen Generiervorlagen und Steuerungsanweisungen
- Variablenregeln steuern die Ablagestruktur für die von SiVArc generierten HMI-Variablen.
- Kopierregeln lösen auf der Basis von Kopiervorlagen oder Typen die Generierung folgender HMI-Objekte aus:
 - Bilder
 - C- und VB-Skripte
 - Textlisten
 - Meldungen
 - Variablentabellen
 - Grafiklisten
 - Grafiktabelle
 - Grafiken
 - Geplante Aufgaben

SiVArc-Regeln sind eine zentrale Funktionalität von SiVArc und haben einen direkten Bezug zum Anwenderprogramm. Daher können Sie SiVArc-Regeln wie Anweisungen mit einem Know-How-Schutz belegen. Beim Projektieren von Bildregeln können Sie unter "Kopiervorlage/Typ eines Bildes" den gewünschten "Bildtyp" suchen. Bildtypen können zusammen mit Eigenschaften und Ereignissen projektiert werden.

Sie können in jedem Editor eine neue Regel/Regelgruppe zu einem beliebigen Index hinzufügen. Dazu führen Sie einen Rechtsklick auf den Index aus > und fügen anschließend eine neue Regel/Regelgruppe ein.

- Beim Hinzufügen einer neuen Regel an einem Indexpunkt wird die neue Regel unmittelbar unterhalb des ausgewählten Indexes eingefügt und wie folgt benannt: "*Name des ausgewählten Indexes_<<laufende Nummer>>*".
- Wenn eine neue Regelgruppe hinzugefügt wird, wird die neue Regelgruppe am ersten Indexpunkt innerhalb der ausgewählten Regelgruppe hinzugefügt und wie folgt benannt: "*Name der Indexgruppe_<<laufende Nummer>>*".

Unterschiede zur Projektierung ohne SiVArc-Regeln

Im Gegensatz zur herkömmlichen WinCC-Projektierung bleibt in einem SiVArc-Projekt der Bezug zwischen einer SiVArc-Regel und einem generierten HMI-Objekt erhalten.

Wenn Sie eine SiVArc-Regel verändern, werden die auf der Basis dieser Regel generierten Objekte bei der nächsten Generierung überschrieben. Wenn Sie eine Regel löschen, werden die zur Regel zugehörigen generierten Objekte bei der nächsten Generierung automatisch entfernt.

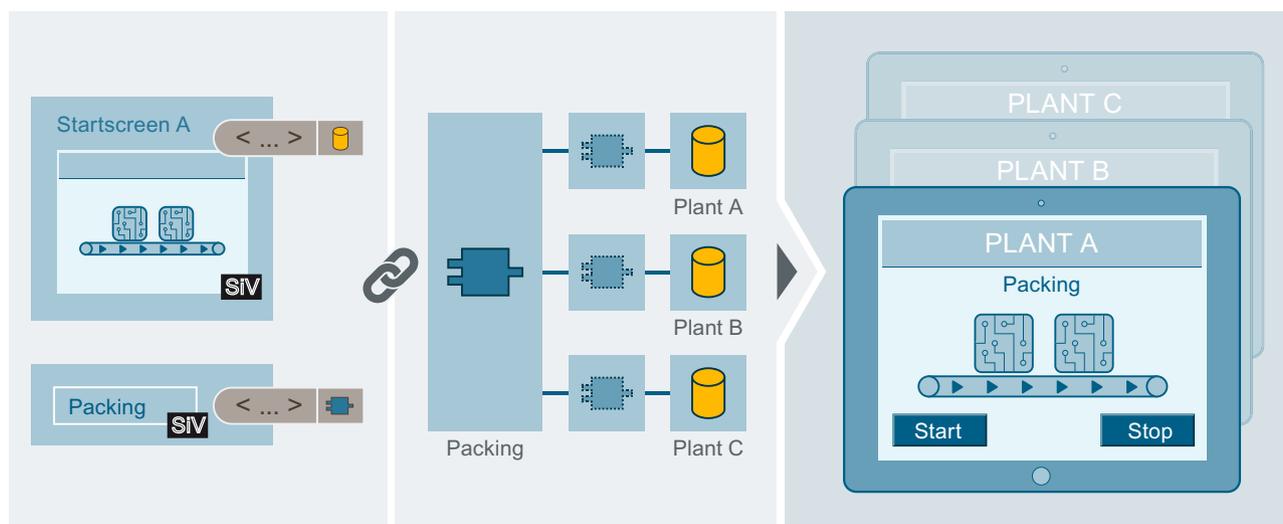
Außerdem können Sie über die SiVArc-Regeln auch für einzelne Geräte die Visualisierung getrennt erzeugen.

Zweck und Nutzen von SiVArc-Regeln

Über SiVArc-Regeln steuern Sie die HMI-Objekte zentral, mit direktem Bezug zum Steuerungsprogramm und individuell für jedes HMI-Gerät. Änderungen sind daher zentral und projektweit umsetzbar. Konzeption und Entwicklung von SiVArc-Regeln bieten einen hohen Mehrwert in Bezug auf die Steuerbarkeit und Effizienz eines WinCC-Projekts.

Beispiel: Bildregeln

Das folgende Beispiel zeigt abstrahiert, wie Sie Texte aus Datenbausteinen über Generiervorlagen in ein HMI-Bild integrieren:



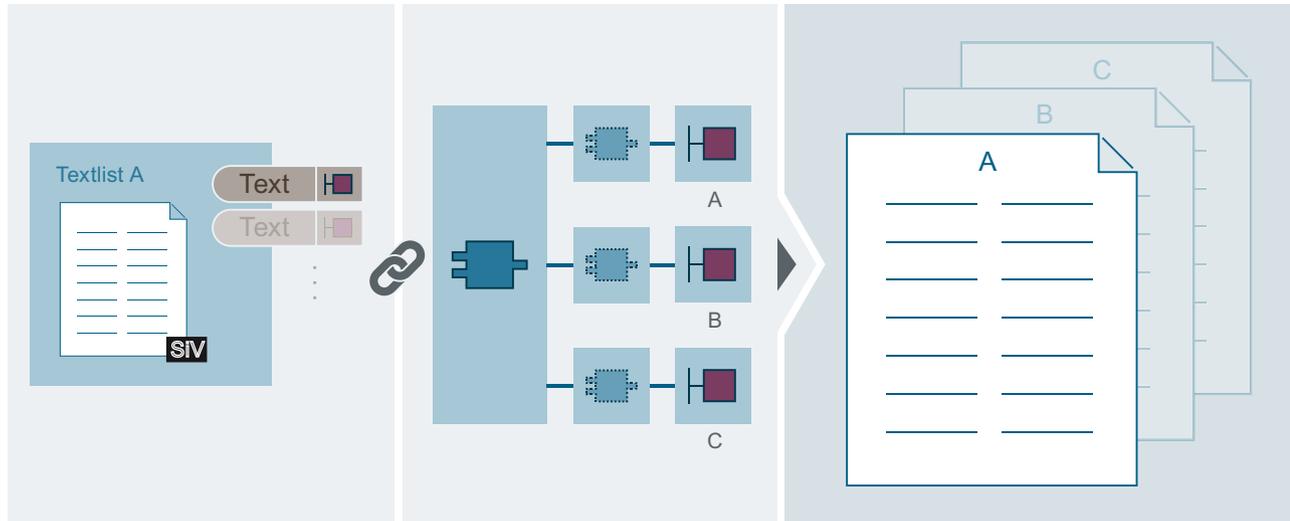
-  SiVArc-Generiervorlage
-  SiVArc-Eigenschaft mit referenzierter Textquelle
-  Prozessanweisung
-  Instanz einer Anweisung im OB Main
-  Datenbaustein

Ergebnis

Für jede Instanz des referenzierten Bausteins wird ein Anzeige- und Bedienobjekt der referenzierten SiVArc-Kopiervorlage erstellt. Die Eigenschaften des Anzeige- und Bedienobjekts werden entsprechend der SiVArc-Regel und den SiVArc-Eigenschaften erstellt. SiVArc legt die generierten Bilder entsprechend Ihrer Projektierung ab.

Beispiel: Textlistenregeln

Das folgende Beispiel zeigt abstrahiert, wie Sie Textlisten mit Texten aus einem Netzwerk generieren:



	SiVArc-Generiervorlage
	SiVArc-Eigenschaft mit referenzierter Textquelle
	Anweisung
	Instanz einer Anweisung im OB Main
	Netzwerk

Ergebnis

Für jede Instanz des referenzierten Bausteins wird eine Textliste der referenzierten SiVArc-Kopiervorlage erstellt. Die Eigenschaften der Textliste werden entsprechend der SiVArc-Regel und den SiVArc-Eigenschaften erstellt.

SiVArc generiert anschließend für jeden aufgerufenen Programmbaustein die Werte für die im Anwenderprogramm konfigurierten Textlisteneinträge.

Siehe auch

Know-how-Schutz für ein SiVArc-Projekt einrichten (Seite 204)

Ändern von SiVArc-Regeln (Seite 212)

Erstellen von SiVArc-Regeln (Seite 165)

Bildregel anlegen (Seite 188)

Textlistenregel anlegen (Seite 189)

6.3.2 Erstellen von SiVArc-Regeln

Definition

Die SiVArc-Regeln erstellen und bearbeiten Sie in den SiVArc-Editoren.

Mit den SiVArc-Regeleditoren steuern Sie zentral die Generierfunktionen von SiVArc. Auf diese Weise können Sie alle Regeln gesammelt bearbeiten und organisieren. Zum Beispiel können Sie Regeln gesammelt exportieren oder importieren.

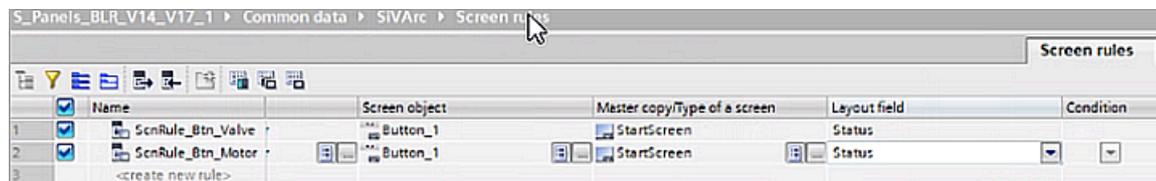
Die SiVArc-Editoren sind Tabelleneditoren. Für jede Art von SiVArc-Regel gibt es einen eigenen Editor. Die Editoren haben unterschiedliche Zielsetzungen.

Zugriff auf die SiVArc-Regeleditoren

Um einen SiVArc-Editor zu öffnen, doppelklicken Sie in der Projektnavigation "Gemeinsame Daten > SiVArc" auf den jeweiligen Eintrag.

Über die Symbole der Symbolleiste blenden Sie einzelne Spalten ein oder aus, z. B. die Spalten "PLC", "HMI-Gerät" oder "HMI-Gerätetyp".

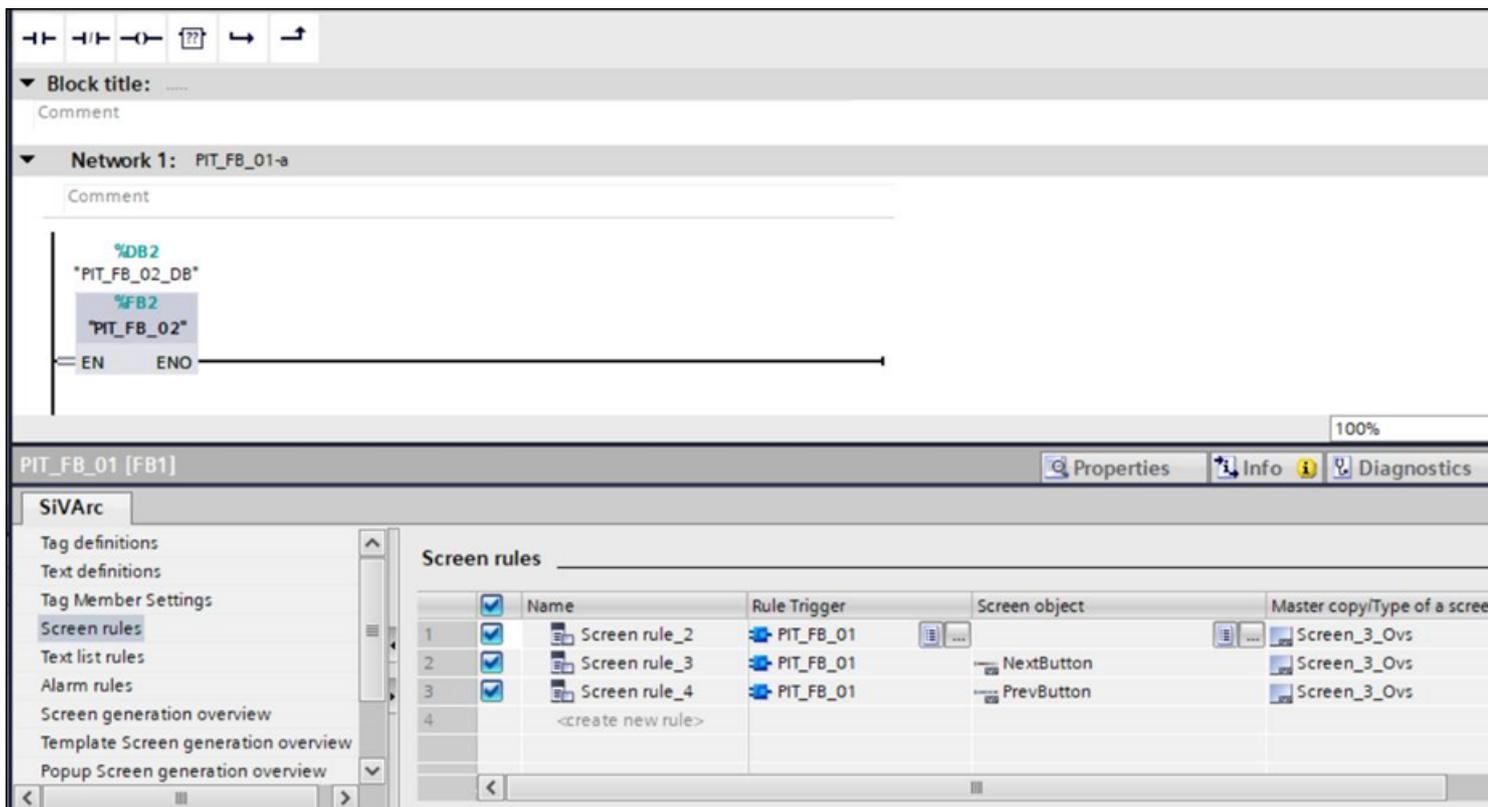
Sie haben die Möglichkeit, Bilder durch das Projektieren von Ausdrücken in der Spalte "Bedingung" zu generieren, die zur Breite und Höhe des Zielgeräts aufgelöst wird.



	Name	Screen object	Master copy/Type of a screen	Layout field	Condition
1	ScnRule_Btn_Valve	Button_1	StartScreen	Status	
2	ScnRule_Btn_Motor	Button_1	StartScreen	Status	
3	<create new rule>				

Bild- und Textlistenregeln sind auch in STEP 7 erreichbar:

Direkt am Programmbaustein sind alle Bild- und Textlistenregeln erreichbar, die für den ausgewählten Programmbaustein angelegt sind. Der Umfang der angezeigten Regeln ist abhängig von der Steuerung.



Abgesehen vom Import/Export erstellen und bearbeiten Sie in STEP 7 die SiVArc-Regeln wie im eigentlichen SiVArc-Editor. Im Inspektorfenster gibt es keine Symbolleiste.

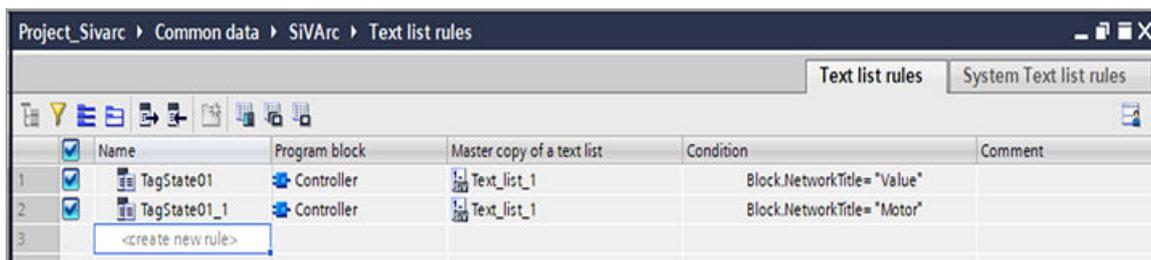
Den Know-How-Schutz von SiVArc-Regeln in STEP 7 heben Sie ausschließlich mit den Befehlen des Kontextmenüs in der Projektnavigation unter "Gemeinsame Daten > SiVArc" auf.

Nach der ersten Generierung stehen Ihnen im Inspektorfenster unter "Plug-Ins > SiVArc" zusätzlich die Registerkarten "Bild-Generier-Übersicht" und "Textlisten-Generier-Übersicht" zur Verfügung.

Verschaltung von HMI-Objekten mit Programmbausteinen

Im Editor "Bildregeln" definieren Sie die Bildregeln, nach denen SiVArc die HMI-Objekte in Bildern für verschiedene Geräte generiert.

Im Editor "Textlistenregeln" definieren Sie SiVArc-Regeln, nach denen Textlisten für verschiedene Geräte generiert werden.



Dabei legen Sie die Regeln an und legen für jede Regel Folgendes fest:

- Verknüpfte Komponenten
 - Bildregel: Programmbaustein, Bildobjekt, Bild, Positionierungsfeld
 - Textlistenregel: Programmbaustein, Textliste
- Bedingungen für die Ausführung der Regel
- Kommentar zur Regel

Im Editor "Bildregeln" können Sie zusätzlich die Spalten "PLC", "HMI-Gerät" oder "HMI-Gerätetyp" einblenden. In den Spalten können Sie die Geräte aktivieren oder deaktivieren, auf die SiVArc diese Regeln bei der Generierung anwendet. Auf diese Weise generieren Sie für ausgewählte Geräte HMI-Objekte.

Steuerung der Ablagestruktur für Variablen

Im Editor "Variablenregeln" definieren Sie Variablenregeln, nach denen die von SiVArc generierten externen Variablen strukturiert in der Projektnavigation abgelegt werden.

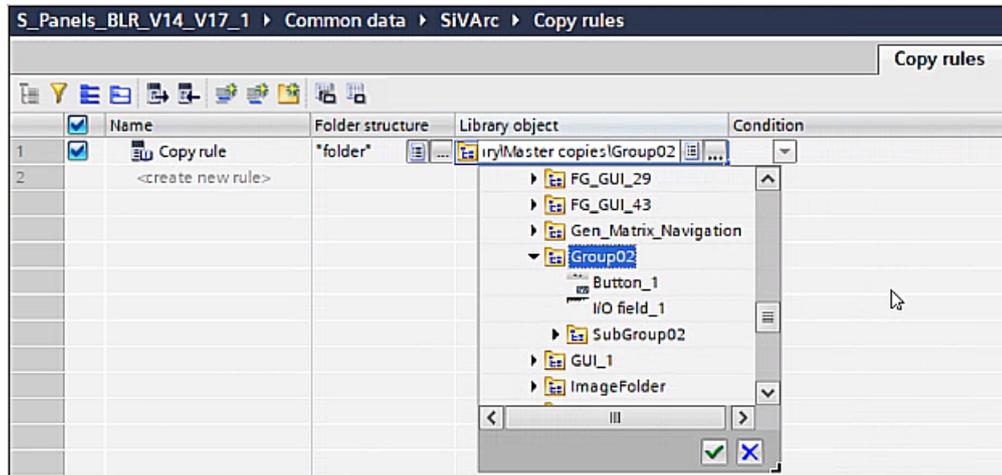
	Name	Index	Tag group hierarchy	Tag table	Condition
1	<input checked="" type="checkbox"/> Tag rule	0	HmiTag.DB.FolderPath	HmiTag.DB.SymbolicName	
2	<input checked="" type="checkbox"/> <create new rule>				

Dabei legen Sie die Regeln an und definieren für die generierten Variablen Folgendes:

- Name der Variablengruppe
- Name der Variablen-tabelle
- Reihenfolge der Abarbeitung der Regel
- Bedingungen für die Ausführung der Regel
- Kommentar zur Regel

Systematisches Einfügen von HMI-Objekten in ein Projekt

Im Editor "Bibliotheksregeln" definieren Sie Regeln, nach denen ausgewählte Objekte aus der Bibliothek für verschiedene HMI-Geräte erzeugt werden.



Dabei legen Sie die Regeln an und legen damit Folgendes fest:

- zu erzeugendes Bibliotheksobjekt
oder
zu erzeugende HMI-Objekte einer Gruppe von Bibliotheksobjekten
- Kommentar zur Regel

Sie können zusätzlich die Spalten "HMI-Gerät" und "HMI-Gerätetyp" einblenden. In den Spalten können Sie die Geräte aktivieren oder deaktivieren, auf die SiVArC diese Regeln bei der Generierung anwendet. Auf diese Weise generieren Sie für ausgewählte Geräte HMI-Objekte.

Siehe auch

SiVArC-Regeln exportieren und importieren (Seite 201)

Know-how-Schutz für ein SiVArC-Projekt einrichten (Seite 204)

SiVArC-Regeln bearbeiten und verwalten (Seite 199)

Bildregel anlegen (Seite 188)

Textlistenregel anlegen (Seite 189)

Anzeige in den SiVArC-Editoren bearbeiten (Seite 277)

6.3.3 Einsatz von SiVArC-Scripting in SiVArC-Regeln

Definition

In SiVArC-Regeln verwenden Sie SiVArC-Scripting in den Bedingungen zur Ausführung einer Regel. In Variablenregeln verwenden Sie SiVArC-Ausdrücke auch zur Strukturierung von externen Variablen.

Grundsätzlich können Sie immer einen SiVArC-Ausdruck verwenden, wenn an einem Eingabefeld der Editor "SiVArC-Ausdrücke" hinterlegt ist.

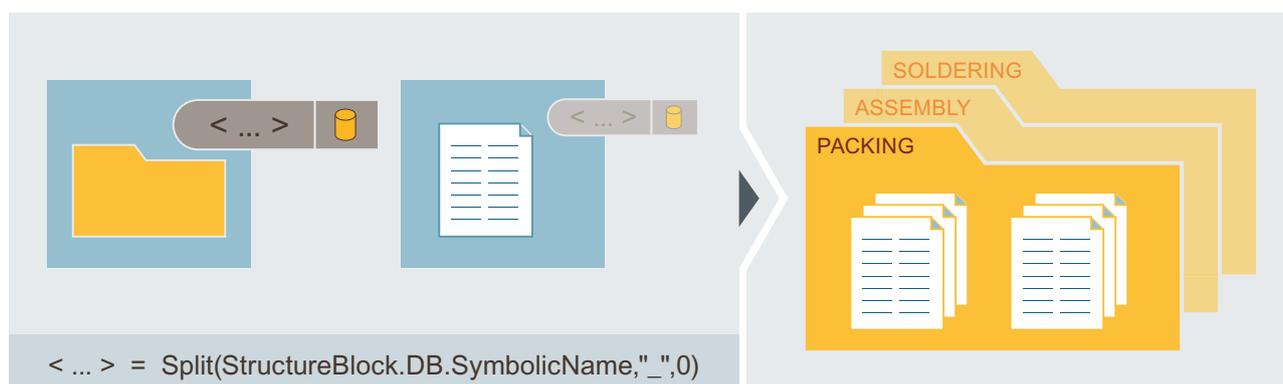
Bedingungen in SiVArC-Regeln

Für Bedingungen verwenden Sie eine SiVArC-Scripting-Funktion, die den Booleschen Wert TRUE oder FALSE zurückgibt.

Wenn keine Bedingung formuliert ist, wird die Bildregel immer ausgeführt. Sie können einer ganzen Regelgruppe eine Bedingung geben. Die Bedingung gilt dann für alle in der Gruppe enthaltenen Regeln. Über Operanden können Sie für einzelne Regeln der Regelgruppe die Bedingung verfeinern.

Ablagestruktur von Variablen

Über SiVArC-Ausdrücke legen Sie die Namen der Variablengruppe und Variablen-tabelle in der Projektnavigation fest. Dadurch verknüpfen Sie die Ablagestruktur mit dem Anwenderprogramm. Die Variablen werden auf diese Weise z. B. entsprechend der Bausteininstanz eingeordnet.



Durch die SiVArC-Ausdrücke `HmiTag.DB.SymbolicName` und `HmiTag.DB.FolderPath` für den Editor "Variablenregeln" strukturieren Sie mit nur einer Variablenregel die Variablen-tabellen entsprechend dem Steuerungsprogramm. Nur der Steuerungsprogrammierer strukturiert das Projekt. Für die Visualisierung übernimmt SiVArC dann die Ablagestruktur aus STEP 7.

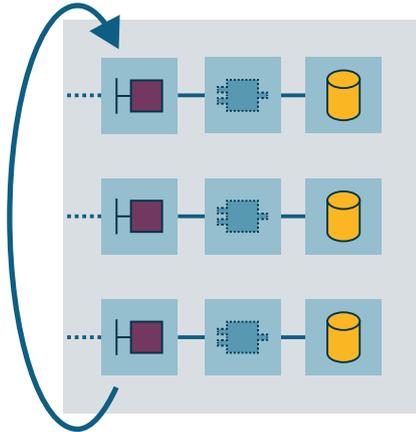
Nur im Editor "Variablenregeln verwenden Sie SiVArC-Ausdrücke, die das SiVArC-Objekt `HMI Tag` ansprechen.

	<input checked="" type="checkbox"/>	Name	Index	Variablengruppe	Variablen-tabelle	Bedingung
1	<input checked="" type="checkbox"/>	Plantsection_1	0	HmiTag.DB.FolderPath	HmiTag.DB.SymbolicName	
2		<neue Regel ers...				

6.3.4 Verarbeitung von Regeln

Funktionsweise

SiVArC durchläuft beim Generieren das Anwenderprogramm. Falls für einen Funktionsbaustein eine Regel gilt, wird die Regel ausgeführt. Parallel durchläuft SiVArC die Datenbausteine.



Sobald eine externe Variable zu generieren ist, durchläuft SiVArC die Variablenregeln von oben nach unten und legt die Variable nach den Vorgaben ab. Daher wird pro Variable nur eine Regel angewandt.

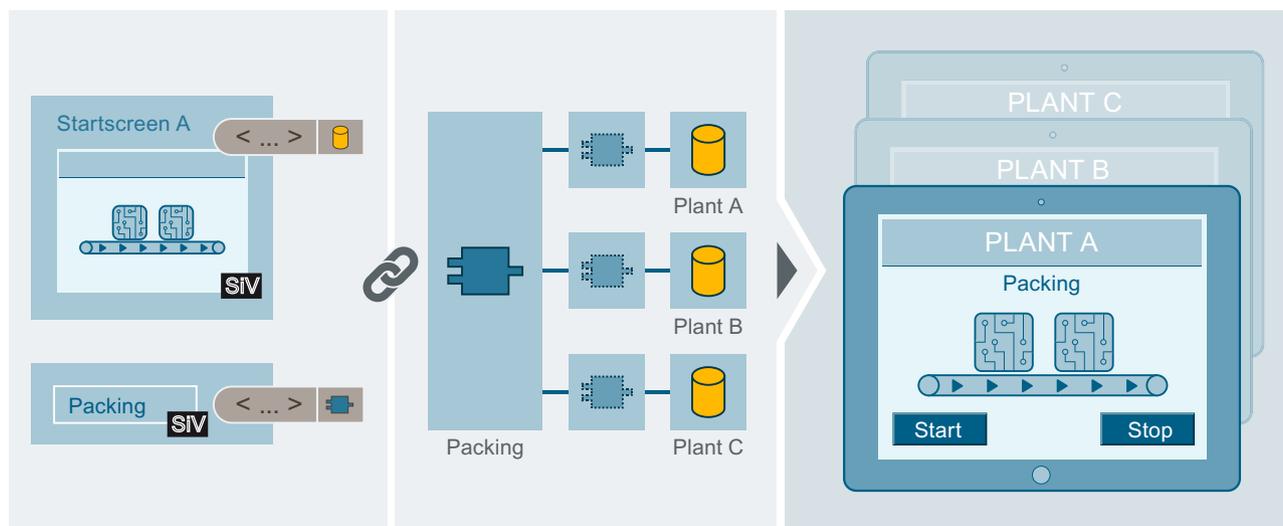
Alle Bild- und Textlistenregeln eines Projekt, die aktiviert sind und eine projektierte Bedingung erfüllen, werden ausgeführt.

Auswertung der Bildregeln

Für eine Bildregel gelten folgende Grundsätze:

- Für jedes zu generierende Bildobjekt müssen Sie eine Bildregel definieren.
- Wenn Sie aus einem Programmbaustein unterschiedliche Bildobjekte generieren wollen, müssen Sie für jedes Bildobjekt eine Bildregel mit einer Bedingung definieren. In der Bedingung legen Sie fest, welches Bildobjekt generiert wird.
- Wenn für ein zu generierendes Bildobjekt das Bild noch nicht existiert, wird das Bild beim Generieren angelegt.
- Wenn im Editor "Bildregeln" ein Baustein in mehreren Bildregeln enthalten ist, dann werden die Objekte in der Reihenfolge der Bildregeln angelegt.
- Wenn Sie zu einem Programmbaustein ein Bild ohne Bildobjekt generieren wollen, dann lassen Sie das Feld "Bildobjekt" leer.

SiVArC durchläuft das Anwenderprogramm entlang der Aufrufhierarchie der gewählten PLCs. Für jede PLC wird der Baustein "Main" durchlaufen. SiVArC wertet für jeden aufgerufenen Programmbaustein die Bildregeln aus.



Für jede zutreffende Bildregel wird auf Basis der Generiervorlage das entsprechende Anzeige- und Bedienobjekt im angegebenen Bild generiert. Beim Generieren werden die SiVArc-Ausdrücke in den SiVArc-Eigenschaften, -Ereignissen und -Animationen der Generiervorlagen ausgewertet.

Auswertung der Variablenregeln

Die Reihenfolge der Variablenregeln ist für die Ablage der externen Variablen relevant. Sortieren Sie bei Bedarf die Reihenfolge per Drag&Drop.

Ordnen Sie z. B. anlagenspezifische Regeln an erster Stelle an und Regeln, die Variablen für anlagenübergreifende Funktionen strukturiert ablegen, an letzter Stelle an. So stellen Sie sicher, dass alle anlagenspezifischen Variablen gesammelt abgelegt werden.

SiVArc durchläuft die Datenbausteine aller PLCs, die im Dialog zur Stationenauswahl aktiviert wurden. Wenn im Datenbaustein die Option "Erreichbar in HMI" gesetzt ist, generiert SiVArc für die Variablen des Datenbausteins jeweils eine externe Variable.

Für jede zu generierende externe Variable durchläuft SiVArc die Variablenregeln von oben nach unten und wertet die zugehörige Bedingung aus. Sobald eine Bedingung zutrifft, wird die Regel angewendet und die externe Variable der Regeln entsprechend in der Projektnavigation abgelegt. Die nachfolgenden Variablenregeln werden nicht mehr verarbeitet. Stattdessen fährt SiVArc mit der nächsten zu generierenden externen Variablen fort.

Wenn keine der Variablenregeln für eine zu generierende externe Variable zutrifft, wird diese externe Variable in der Standard-Variablentabelle abgelegt.

Abhängig von der Einstellung unter "Extras > Einstellungen > SiVArc" generiert SiVArc nur externe Variablen, die auch im generierten SiVArc-Projekt verschaltet werden.

Beim Generieren verarbeitet SiVArc die Einstellungen für Variablen in den Runtime-Einstellungen des Bediengeräts. Die generierten Namen der externen Variablen bilden die symbolische Adresse der Variablen im Datenbaustein ab, und zwar entsprechend der Variablen-Synchronisierung von WinCC.

Auswertung der Kopierregeln

SiVArC verarbeitet die Kopierregeln. Pro Kopierregel wird für jedes angegebene Bediengerät das entsprechende HMI-Objekt in der Projektnavigation angelegt.

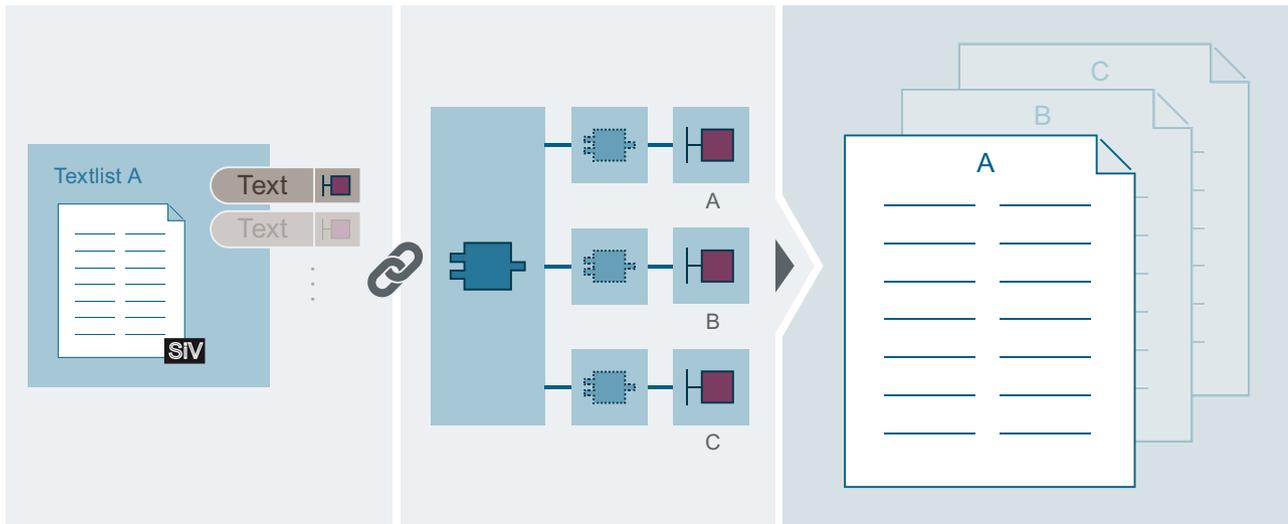
Auswertung der Textlistenregeln

Die Reihenfolge der Textlistenregeln hat keine Auswirkung, weil die Anwendung der Textlistenregeln durch die Aufrufhierarchie der Programmbausteine im Anwenderprogramm definiert wird.

Grundsätzlich verarbeitet SiVArC alle Textlistenregeln, die den gerade von SiVArC ausgewerteten Programmbaustein enthalten.

Die SiVArC-Eigenschaften der Textliste werden dabei ausgewertet. Wenn die Textliste bereits generiert wurde, wird die Textliste um die neuen Einträge erweitert und bestehende identische Einträge werden überschrieben.

Die Textliste wird in dem Bediengerät abgelegt, für das die Generierung ausgelöst wurde.



SiVArC generiert anschließend für jeden aufgerufenen Programmbaustein die Werte für die im Anwenderprogramm konfigurierten Textlisteneinträge. Dabei durchläuft SiVArC das Anwenderprogramm entlang der Aufrufhierarchie aller OBs der gewählten PLCs.

Priorität der generierten Objekte bei Namenskonflikten

Bei Namenskonflikten priorisiert SiVArc während einer Generierung wie folgt:

1. Generierte Objekte aus Bild-, Variablen- und Textlistenregeln
2. Generierte Objekte aus Kopierregeln
Aus den Kopierregeln generierte Objekte werden wie manuell erstellte Objekte behandelt. Sie werden bei der Generierung zuerst erzeugt. Bei Namenskonflikten mit nachfolgend generierten Objekten werden Objekte aus Kopierregeln umbenannt mit dem Anhang "_renamed".
3. Manuell erstellte Objekte
Bei Namensgleichheit von manuell erzeugten Objekten und generierten Objekten werden die manuell erzeugten Objekte umbenannt.

Siehe auch

Bildregel anlegen (Seite 188)

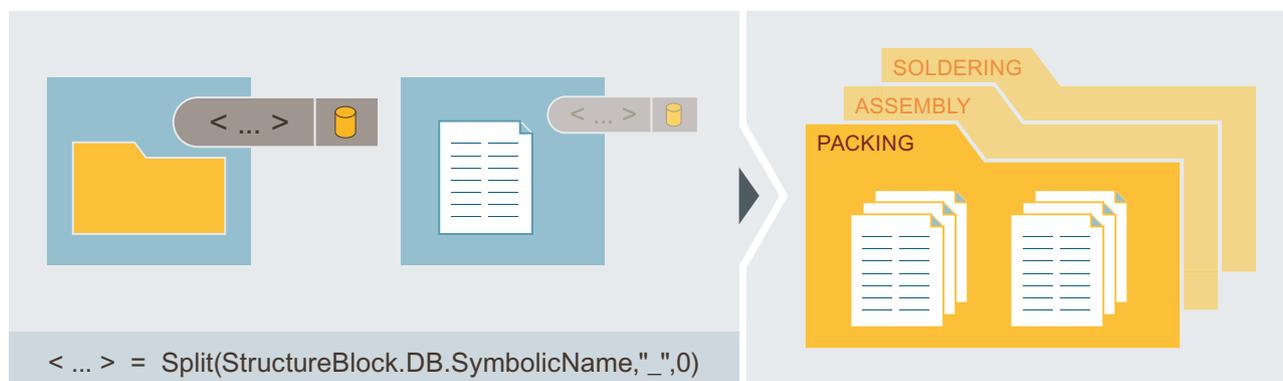
Textlistenregel anlegen (Seite 189)

Bildlegende (Seite 279)

6.3.5 Generieren von Variablen

Definition

SiVArc generiert eine externe HMI-Variable für jede PLC-Variable mit der Option "Zugriff von HMI". Gemäß dem unter "Optionen > Einstellungen > SiVArc" projektweit festgelegten Modus generiert SiVArc entweder alle diese Variablen oder nur die im Projekt verwendeten Variablen. Bereits bei der Generierung werden externe Variablen in Variablen tabellen gespeichert und der Projektbaum gemäß Ihren Strukturierungsvorgaben gruppiert.



Interne HMI-Variablen und SiVArc-Variablen

Wenn Sie interne HMI-Variablen mit SiVArc erstellen möchten, speichern Sie die entsprechende Masterkopie in der Bibliothek. Sie verwenden diese Masterkopie dann in einer Bibliotheksregel.

Externe Variablen werden immer vor internen HMI-Variablen generiert. Bei Benennungskonflikten werden generierte interne HMI-Variablen und manuell erstellte Variablen umbenannt.

Im Gegensatz zu manuell erstellten HMI-Variablen und SiVArC-Variablen besitzen generierte externe Variablen eine direkte Verknüpfung zu SiVArC und werden während jeder Generierung erneut erhoben.

Vorteile der Variablengenerierung

Der Generierungsmechanismus für die Variablen ermöglicht die automatische Verschaltung der PLC-Variablen in den Anzeige- und Bedienobjekten. Mit dem für die Variablengenerierung ausgewählten Modus entfernen Sie überflüssige Variablen aus dem Projekt. Das Ergebnis ist eine effiziente und fehlerfreie Konfiguration, ohne dabei unnötigen Speicherplatz zu belegen.

Verschaltung von externen Variablen

Sie erstellen automatische Verschaltungen durch Konfigurieren der Prozessvariablen als SiVArC-Ausdrücke in den Generierungsvorlagen. Wenn der SiVArC-Ausdruck als ein vorhandener Variablenname während der Generierung ausgewertet wird, wird das generierte Anzeigeobjekt mit dieser Variable verschaltet.

Festlegen des Aktualisierungszyklus und des Erfassungstyps

Sie können bei Bedarf den Aktualisierungszyklus und den Erfassungstyp von generierten externen HMI-Variablen in mehreren Schritten festlegen:

- Für einzelne Programmbausteine
Sie definieren den Aktualisierungszyklus und den Erfassungstyp von Variablen für einen Programmbaustein mit der Option "Konfiguration auf alle Variablen anwenden" im Inspektorfenster eines Datenbausteins unter "Plug-Ins > SiVArC > HMI-Variablen-Einstellungen". Diese Einstellung deaktiviert die Einstellungen für einzelne Variablen.
- Für einzelne Variablen
Wenn die Option "Konfiguration auf alle Variablen anwenden" deaktiviert ist, konfigurieren Sie jede Variable im Datenbaustein einzeln.
- Projektweit
In den SiVArC-Einstellungen unter "Allgemeine Daten > SiVArC > SiVArC Einstellungen > Variablengenerierung Einstellungen" konfigurieren Sie alle externen Variablen des Projekts, die generiert werden. Diese Einstellung wird nur dann ausgewertet, wenn keine andere Einstellung für die Variablengenerierung definiert ist.

Benutzerdatentypen unterstützen nur zyklische Erfassungstypen. Wenn Sie den Erfassungstyp "Auf Anforderung" für das gesamte Projekt oder für einen Programmbaustein festlegen, wird der Aktualisierungszyklus auf 1 s und der Erfassungstyp auf "Zyklisch im Betrieb" für Benutzerdatentypen gesetzt.

Der Aktualisierungszyklus 500 ms wird automatisch für HMI-Geräte festgelegt, die das Festlegen des Erfassungstyps und des Aktualisierungszyklus nicht unterstützen.

Für Unified-Geräte ist ein Erfassungszyklus von 250 ms verfügbar.

Hinweis**Kopieren des Programmbausteins mit Variablenkonfiguration**

Sie nehmen die Einstellungen am Aktualisierungszyklus und Erfassungstyp für jeden Programmbaustein erneut vor. Auch wenn Sie einen vollständig konfigurierten Programmbaustein kopieren, konfigurieren Sie seine Einstellungen für die Variablengenerierung erneut.

Standardeinstellungen für Variablennamen

Die folgenden Standardeinstellungen werden für generierte Variablennamen im TIA Portal festgelegt:

- Das Trennzeichen ist immer "_"
- Eckige Klammern "[" und "]" werden durch "{" und "}" ersetzt

Verwenden Sie, falls nötig, das SiVArc-Objekt `TagNaming`, das diese Einstellungen verarbeitet, in den SiVArc-Ausdrücken. Weitere Informationen finden Sie im Abschnitt "Prinzip der Variablengenerierung (Seite 181)".

Hinweis**Trennzeichen in strukturierten Variablen**

Die Hierarchieebenen werden in den strukturierten Variablen immer durch "." getrennt.

Leerzeichen in Variablennamen

SiVArc berücksichtigt keine Leerzeichen bei der Generierung von Variablen. Das gilt beispielsweise auch dann, wenn ein Funktionsbaustein, der mit Variablen über SiVArc versorgt wird, ein Leerzeichen in seinem Namen besitzt.

SiVArc ignoriert dieses Leerzeichen. In diesem Fall können Fehler bei der Verschaltung auftreten.

Beispiel

Ein DB-Instanzname eines Funktionsbausteins enthält ein Leerzeichen: "TT5684 Temperatur", da dieser als Meldetext verwendet wird. Wenn Sie das Leerzeichen nicht aus dem Namen des Funktionsbausteins entfernen, wird der Baustein erstellt und die Schnittstelleneigenschaft mit der nicht vorhandenen Variable "TT5684Temperatur" (ohne Leerzeichen) wird rot hervorgehoben.

Löschen Sie das Leerzeichen in dem Variablennamen des Funktionsbausteins. Auf diese Weise passen Sie den SiVArc-Ausdruck an den Variablennamen an, wie er von der Variablenregel erstellt wurde.

Szenarien der Variablengenerierung

PLC-Variablen werden auf HMI-Geräten über das Kontrollkästchen "Zugriff von HMI" im PLC-Variablenfenster verwendet. Beim Erstellen von Bildern werden die Eingabe- und Ausgabefelder eines Bildobjekts mit den PLC-Variablen konfiguriert und dieselben werden bei der HMI-Generierung verwendet. Weitere Informationen zu Variablen finden Sie unter Prinzip der Variablengenerierung.

Fall 1: Bei Auswahl aller HMI-Variablen

Im Fenster "PLC-Variablen" generiert das System, nach der Konfigurierung der PLC-Variablen, die HMI-Variablen unter dem Ordner HMI-Variablen, sofern das Kontrollkästchen "Zugriff von HMI" markiert ist und die Generierung mit dem auf "Alle HMI Variablen" gesetzten Modus der Variablengenerierung ausgelöst wird. Die generierten HMI-Variablen werden unter der Standard-Variablen-tabelle mit dem Variablennamen "Bausteinname"_"Eingangs-/Ausgangs-Variablenname" generiert.

Fall 2: Bei Auswahl von "Benutzte HMI Variablen"

Fügen Sie ein Ein-/Ausgabegerät hinzu und konfigurieren Sie das Ein-/Ausgabefeld mit einem dynamischen Variablennamen im SiVArC-Bildeditor. PLC-Variablen werden für einen PLC und dynamische Variablen werden für Bilder oder Bildobjekte konfiguriert. Der dynamische Variablenname bezieht sich auf einen beliebigen Ausdruck, der sich in einen PLC-Namen auflöst und dem der PLC-Variablenwert als Suffix angehängt ist. Wenn das Kontrollkästchen "Zugriff von HMI" im PLC-Variablenfenster markiert ist und die SiVArC-Generierung mit dem Modus "Variablengenerierung", der auf "Benutzte HMI Variablen" gesetzt ist, ausgelöst wird, werden nur solche PLC-Variablen, die im HMI-Gerät verwendet werden, unter "HMI Variablen" > "Standard-Variablen-tabelle" generiert.

Fall 3: Bei mehreren PLCs und Variablenregel

Bei mehreren angeschlossenen PLCs konfigurieren Sie die Variablenregeln mit den Ausdrücken "S7.Control.Name" und "HmiTag.SymbolicName" und markieren das Kontrollkästchen "Zugriff von HMI". Konfigurieren Sie die SiVArC-Bilder und lösen Sie die SiVArC-Generierung mit dem "Modus der Variablengenerierung" aus, den Sie auf "Alle HMI Variablen" oder "Benutzte HMI Variablen" setzen. Die HMI-Variablen werden mit der Benennung "PLC-Name"_"PLC-Variablenname" generiert. Sie finden die generierten PLC-Variablen unter der "Generier-Übersicht". Die HMI-Variablen werden unter dem entsprechenden Variablenordner und der Variablen-tabelle generiert, die auf den in den Variablenregeln konfigurierten Ausdrücken basieren. Die "Generier-Übersicht" unterstützt die Anzeige von PLC-Variablen zusammen mit Bausteinvariablen. Weitere Details über den Generierungsvorgang finden Sie unter Visualisierung generieren (Seite 205).

Siehe auch

Beispiel: Variablennamen anpassen (Seite 197)

Variablenregel anlegen (Seite 186)

6.3.6 Einsatz von Kopierregeln

Definition

Eine Kopierregel kopiert beim Generieren der Visualisierung HMI-Objekte auf der Grundlage von Kopiervorlagen oder Typen.

Kopierregeln erstellen Sie ausschließlich für folgende HMI-Objekte:

- Interne Variablen
- Textlisten
- Meldungen
- Vorlagen
- Bilder
- Skripte
- Grafiklisten
- Grafiken
- Grafiktabelle
- Geplante Aufgaben

Eine Kopierregel unterstützt SiVArc-Ausdrücke und Bedingungen. Die aus Kopierregeln erzeugten HMI-Objekte sind zwar unabhängig vom Anwenderprogramm, haben jedoch wie andere generierte Objekte einen Bezug zu SiVArc. Bei der nächsten Generierung werden die kopierten Objekte daher neu erfasst und bei Bedarf aktualisiert. Wenn die verwendeten Kopiervorlagen aus der Bibliothek gelöscht wurden, werden die vorher daraus generierten Objekte aus der Projektnavigation entfernt.

Mit Hilfe von geplanten Aufgaben als Bibliotheksobjekten können Sie Regeln auf der Basis von Kopierregeln erstellen.

So verwenden Sie Kopierregeln mit Bibliotheksobjekten:

- Sie können die SiVArc-Generierung mit dem Aufgabenplaner durchführen. Hierdurch werden Objekte als geplante Aufgaben erstellt und auf WinCC Unified-Geräte erweitert. Mehrere Kopierregeln mit demselben Bibliotheksobjekt werden als eine einzige Regel behandelt.
- Wenn von SiVArc generierte geplante Aufgaben und vom Anwender erstellte geplante Aufgaben denselben Namen aufweisen, benennt SiVArc die vom Anwender erstellten Aufgaben automatisch in "Aufgabe <<laufende Nummer>>_umbenannt" um.
 - Geräte mit WinCC Professional unterstützen sowohl geplante Aufgaben, die nur den Typ "Funktionsliste" haben, als auch geplante Aufgaben, die nur den Typ "Druckauftrag" aufweisen.
- Wenn Sie eine geplante Aufgabe vom Typ "Druckauftrag" auf einem Gerät mit WinCC Runtime erstellen und bei der Generierung angeben, dass ein Objekt auf einem Gerät mit WinCC Runtime Advanced generiert werden soll, zeigt SiVArc einen Fehler an.

- Wenn Sie in einem Ordner geplante Aufgaben, ein Bild und eine Variablen-tabelle zusammenfassen, führt dies bei der SiVArc-Generierung dazu, dass Objekte im angegebenen HMI-Gerät des TIA Portals als geplante Aufgaben, ein Bild und eine Variablen-tabelle erzeugt werden.
- Im Hinblick auf die oben genannten Punkte können Sie unter "Station und Steuerungen für SiVArc-Generierung auswählen" auch angeben, dass Energy Suite-Regeln verwendet werden sollen.

Zweck von Kopierregeln

Bei standardisierten Lösungen für das Bedienen und Beobachten werden HMI-Objekte oftmals zentral erstellt und dann als globale Bibliotheken an die Projektoren verteilt. Mithilfe von Kopierregeln erzeugen Sie diese HMI-Objekte automatisiert für jedes HMI-Gerät in Ihrem Projekt.

Mit den Bibliotheksregeln kopieren Sie systematisch auch eine große Anzahl von Objekten, die schon nach bestimmten Projektkriterien vorsortiert sein können. Auf diese Weise stellen Sie die Standardisierung in Ihren Projekten auch für Skripte, Texte, interne Variablen und Grafiken sicher.

Wenn Sie Anzeige- und Bedienobjekte generieren, die mit Skripten verschaltet sind, stellen Sie über die Kopierregeln sicher, dass diese Skripte auch am HMI-Gerät vorhanden sind.

Außerdem legen Sie mit Kopierregeln zu Beginn z. B. interne Variablen und Variablen-tabellen nur für ein HMI-Gerät an. Für alle weiteren HMI-Geräte legen Sie die Variablen in der Projektbibliothek ab, um sie beim nächsten Generieren durch SiVArc automatisch auf alle Geräte kopieren zu lassen.

Sie können Grafiken aus der Projektbibliothek und der globalen Bibliothek mit Hilfe von Kopierregeln bzw. Systemkopierregeln in den Ordner "Languages & Resources > Project graphics" kopieren. Beim Konfigurieren einer Kopierregel bzw. Systemkopierregel können Sie

die Grafiken als Bibliotheksobjekte durchsuchen. Sie können folgende Schritte mit Hilfe von Grafiken in Kopierregeln bzw. Systemkopierregeln ausführen:

- In einem TIA Portal-Projekt:
 - Wird die Kopierregel bzw. die Systemkopierregel mit Grafiken konfiguriert, legt das System bei der SiVArc-Generierung automatisch den Ordner Project graphics in Languages & resources an.
 - Mit Grafiken konfigurierte Kopierregeln bzw. Systemkopierregeln können Grafikobjekte generieren, auch wenn Sie im Stationsauswahldialog kein HMI/PLC-Gerät auswählen. Die generierten Grafikobjekte stehen im Ordnerpfad "Languages & resources > Project graphics zur Verfügung.
 - Bei der zweiten Generierung wird das benutzerdefinierte Grafikobjekt von der Kopierregel bzw. Systemkopierregel umbenannt.
 - Wird die Kopierregel bzw. Systemkopierregel mit einem Ordnerpfad mit SiVArc-Objekten konfiguriert, werden bei der SiVArc-Generierung die nicht unterstützten Objekte vom System ignoriert und die Grafiken im Ordner Project graphics generiert.
 - Beim Konfigurieren von Kopierregeln bzw. Systemkopierregeln können Sie nach bereits generierten Grafikobjekten suchen. Wenn das generierte Grafikobjekt umbenannt wurde und Sie eine zweite Generierung durchführen, generiert das System ein neues Grafikobjekt.

Funktionsweise

Die zu kopierenden HMI-Objekte legen Sie in der Bibliothek innerhalb einer Gruppe oder als Einzelobjekte an. Je nach Bedarf verwenden Sie diese Gruppe in einer Bibliotheksregel oder Sie verwenden einzelne Bibliothekselemente. Die Regeln können Sie in Gruppen sortieren und nach Bedarf auch gruppenweise aktivieren oder deaktivieren. In der Kopierregel steuern Sie außerdem, für welche HMI-Geräte die Regel ausgeführt wird.

Pro Kopierregel wird für jedes angegebene HMI-Gerät das entsprechende HMI-Objekt in der Projektnavigation angelegt.

Namenskonflikte

Aus den Kopierregeln generierte Objekte werden bei Namenskonflikten wie manuell erstellte Objekte behandelt. Sie werden bei der Generierung zuerst erzeugt. Bei Namenskonflikten mit nachfolgend generierten Objekten werden Objekte aus Kopierregeln umbenannt und mit dem Namenszusatz "_renamed" versehen.

Generieren von internen Variablen

Um interne Variablen zu generieren, gehen Sie folgendermaßen vor:

1. Erstellen Sie eine Variablen-tabelle.
2. Projektieren Sie in dieser Variablen-tabelle die internen Variablen.
3. Legen Sie die Variablen-tabelle als Kopiervorlage in der Projektbibliothek ab.
4. Erstellen Sie eine Kopierregel, mit der die Kopiervorlage der Variablen-tabelle in das angegebene HMI-Gerät kopiert wird.

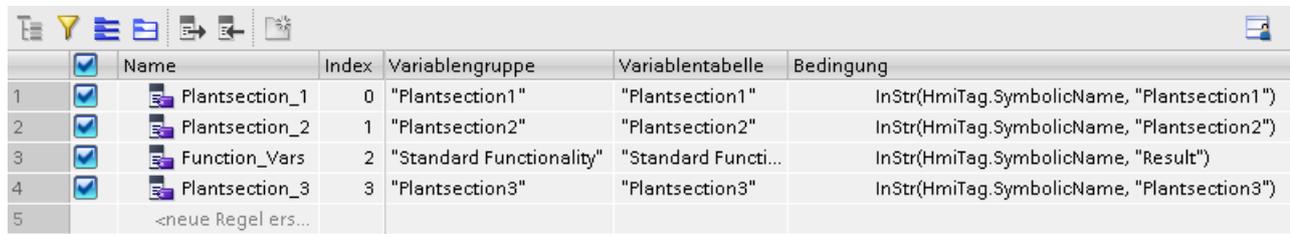
Siehe auch

SiVArC-Regeln (Seite 161)

Kopierregel anlegen (Seite 192)

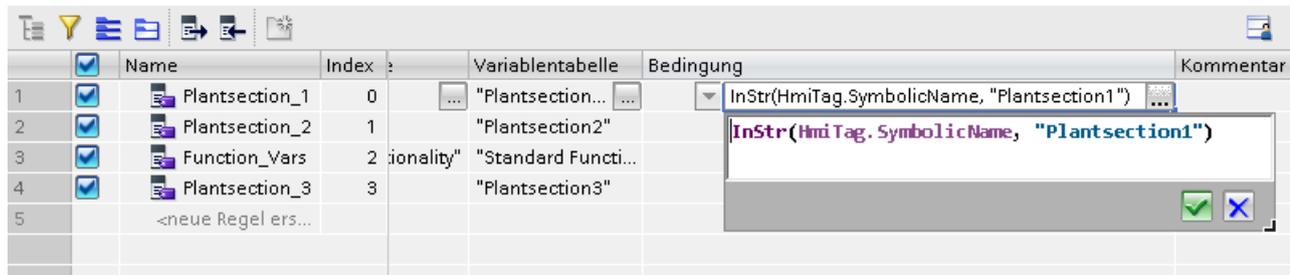
6.3.7 Zusammenhang zwischen SiVArC-Ausdrücken und Bedingungen**Anwendungsbeispiel**

SiVArC-Ausdrücke und Bedingungen verwenden Sie z. B. in Variablenregeln. Über die SiVArC-Ausdrücke legen Sie Gruppennamen und Namen von Variablen fest. Ob die Variablenregel für eine Variable angewandt wird, legen Sie über eine Bedingung fest.



	<input checked="" type="checkbox"/>	Name	Index	Variablengruppe	Variablen-tabelle	Bedingung
1	<input checked="" type="checkbox"/>	Plantsection_1	0	"Plantsection1"	"Plantsection1"	InStr(HmiTag.SymbolicName, "Plantsection1")
2	<input checked="" type="checkbox"/>	Plantsection_2	1	"Plantsection2"	"Plantsection2"	InStr(HmiTag.SymbolicName, "Plantsection2")
3	<input checked="" type="checkbox"/>	Function_Vars	2	"Standard Functionality"	"Standard Functi...	InStr(HmiTag.SymbolicName, "Result")
4	<input checked="" type="checkbox"/>	Plantsection_3	3	"Plantsection3"	"Plantsection3"	InStr(HmiTag.SymbolicName, "Plantsection3")
5		<neue Regel ers...				

Die folgende Bedingung legt z. B. fest, dass die Variablenregel nur für Variablen angewandt wird, in deren Namen die Textkette "Plantsection1" enthalten ist:



	<input checked="" type="checkbox"/>	Name	Index	Variablengruppe	Variablen-tabelle	Bedingung	Kommentar
1	<input checked="" type="checkbox"/>	Plantsection_1	0	...	"Plantsection..."	InStr(HmiTag.SymbolicName, "Plantsection1")	
2	<input checked="" type="checkbox"/>	Plantsection_2	1		"Plantsection2"		
3	<input checked="" type="checkbox"/>	Function_Vars	2	ionality"	"Standard Functi..."		
4	<input checked="" type="checkbox"/>	Plantsection_3	3		"Plantsection3"		
5		<neue Regel ers...					

Beschreibung

SiVArC-Ausdrücke geben einen Text zurück, der auf Textquellen zugreift. Innerhalb von SiVArC-Ausdrücken können Sie auch Bedingungen formulieren. Der Rückgabewert ist immer ein Text.

Bedingungen geben einen Booleschen Wert zurück. Sie nutzen Bedingungen in SiVArC-Regeln.

SiVArC-Ausdrücke verwenden Sie hauptsächlich in Generiertvorlagen. Beim Generieren werden die SiVArC-Ausdrücke ausgewertet. Die SiVArC-Eigenschaften der generierten HMI-Objekte werden mit dem generierten Text projiziert. Die generierten Anzeige- und Bedienobjekte werden auf diese Weise über SiVArC-Ausdrücke z. B. benannt, beschriftet, in Ordnern abgelegt und mit den richtigen Variablen verschaltet.

Bedingungen verwenden Sie für Bild-, Variablen- und Textlistenregeln. Beim Generieren wird die Bedingung ausgewertet und die Regeln werden entsprechend ausgeführt oder übergangen. Auf diese Weise schränken Sie Regeln ein und ermöglichen Ausnahmen.

Siehe auch

SiVArc-Scripting (Seite 107)

If-Bedingungen (Seite 275)

SiVArc-Ausdruck (Seite 109)

6.3.8 Prinzip der Variablengenerierung**Einleitung**

Die Variablengenerierung mit SiVArc greift auf die Variablensynchronisierung von WinCC zurück und verarbeitet die dazugehörigen Runtime-Einstellungen beim Generieren.

Zusätzlich verfügt SiVArc über Funktionen, welche den Umfang der Variablengenerierung und die Ablage der generierten Variablen steuern. Sie definieren vor dem Generieren mit SiVArc den Aktualisierungszyklus und die Erfassungsart von Variablen.

Aufgabe	Umsetzung	TIA Portal
Umfang der Generierung	Kennzeichnung der zu generierenden Variablen in STEP 7	"Erreichbar in HMI" im Datenbaustein
	Modus zur Variablengenerierung mit SiVArc	"Extras > Einstellungen > SiVArc"
Variablenamen	Runtime-Einstellungen für Variablen in WinCC	Runtime-Einstellungen eines Bediengeräts
Ablagestruktur	Variablenregeln von SiVArc	Editor "Variablenregeln"
Aktualisierungszyklus und Erfassungsart	Am Datenbaustein in STEP 7	"Plug-Ins > SiVArc > HMI-Variablen"
	Projektweit	"Gemeinsame Daten > SiVArc > SiVArc-Einstellungen > Variablengenerierung Einstellungen"

Runtime-Einstellungen für Variablen

SiVArc berücksichtigt während der Generierung die Einstellungen für Variablen in den Runtime-Einstellungen des Bediengeräts. So benennt SiVArc die generierten externen Variablen nach den gesetzten Namenskonventionen.

Wenn Sie die Einstellungen für Variablen nach der ersten SiVArc-Generierung ändern, generiert SiVArc alle Variablen entsprechend den neuen Einstellungen. Bestehende SiVArc-Variablen werden umbenannt.

Projektieren Sie die Einstellungen für Variablen einmalig vor der ersten SiVArc-Generierung. Wenn unterschiedliche Einstellungen für Variablen für die Bediengeräte des Projekts erforderlich sind, haben Sie die Möglichkeit, über SiVArc-Ausdrücke mit dem SiVArc-Objekt `TagNaming` auf die Runtime-Einstellungen zuzugreifen.

Umfang der Variablengenerierung mit SiVArc festlegen

Den Modus zur Variablengenerierung wählen Sie unter "Extras > Einstellungen > SiVArc" die gewünschte Option für Ihr Projekt aus.

Wenn Sie die Generierung bereits gestartet haben, prüfen Sie die Einstellung für diese Optionen im Dialog zum Generieren der Visualisierung.

Wenn Sie diese Einstellung erst nach der ersten Generierung wählen, werden die bestehenden externen Variablen entsprechend den Regeln für generierte Objekte verarbeitet:

- Im SiVArc-Projekt ungenutzte externe Variablen werden gelöscht. Auf diese Weise bereinigen Sie die Speicherplatzbelegung.
- Manuell bearbeitete Variablen bleiben erhalten und werden gegebenenfalls umbenannt.

Ablagestrukturen mit SiVArc steuern

Wenn Sie in den Variablenregeln SiVArc-Ausdrücke mit dem SiVArc-Objekt `HMITag` verwenden, werden Änderungen im Anwenderprogramm bei der nächsten Generierung in der Ablagestruktur umgesetzt. Wenn z. B. Funktionsbereiche der Anlage neu aufgeteilt werden, werden auch die verschalteten Variablen nach der neuen Aufteilung in der Projektnavigation sortiert.

SiVArc legt die Variable nach der erste Variablenregel ab, die für die Variable zutrifft. Sortieren Sie bei Bedarf die Reihenfolge per Drag&Drop. Wenn danach eine andere zutreffende Variablenregel nun weiter oben platziert ist, wird bei der nächsten Generierung die Variable nach dieser Regel abgelegt.

Aktualisierungszyklus und Erfassungsart mit SiVArc festlegen

Aktualisierungszyklus und Erfassungsart stellen Sie für die Generierung dreistufig ein:

- Projektweit
- Für einzelne Programmbausteine
- Für einzelne Variablen

Bei jeder Generierung werden diese Einstellungen neu zugewiesen. Trotz zentraler Steuerung können Sie flexibel projektieren und optimieren so die Leistungsfähigkeit des Projekts in Runtime.

Zweck und Nutzen

Die Variablengenerierung ermöglicht eine effiziente und fehlerfreie Projektierung von Variablen. Die Generierung ermöglicht die einheitliche Benennung von Variablen und die automatisierte Verschaltung der generierten Anzeige- und Bedienobjekte.

Die Variablengenerierung mit SiVArc erleichtert den Umgang mit Änderungen und optimiert ein bestehendes Projekt.

Siehe auch

Beispiel: Variablennamen anpassen (Seite 197)

Generieren von Variablen (Seite 173)

Variablenregel anlegen (Seite 186)

Beispiel: Einsatz von SiVArc zum Generieren von Variablen (Seite 18)

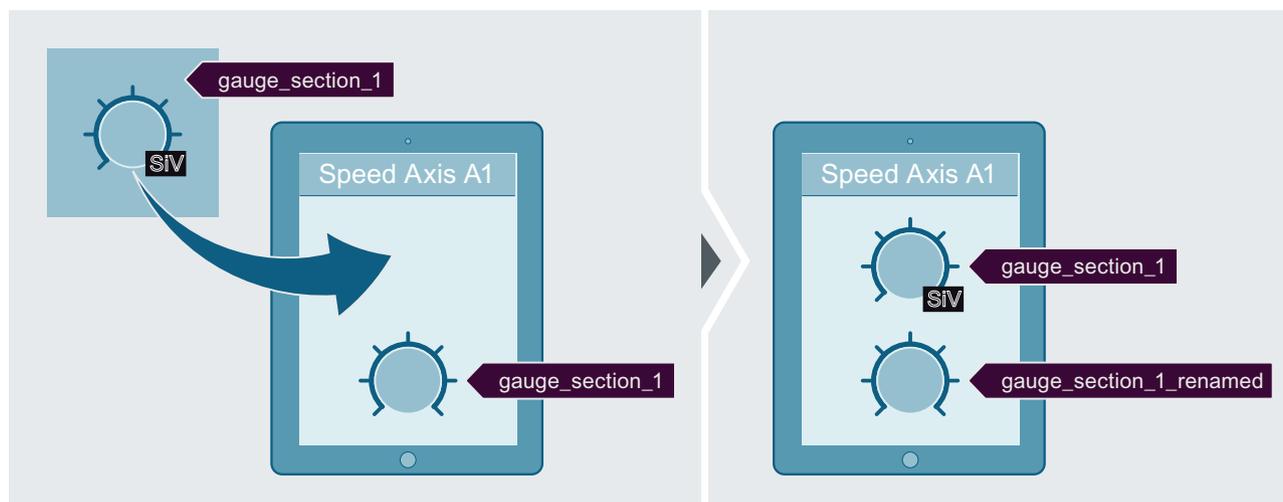
6.3.9 Vermeiden von Konflikten beim Generieren

Namenskonflikte zwischen bestehenden und generierten HMI-Objekten

Wenn in einem SiVArc-Projekt generierte HMI-Objekte und manuell erstellte HMI-Objekte nebeneinander verwendet werden, können Namenskonflikte entstehen. Ein ausgewerteter SiVArc-Ausdruck kann den Namen eines bereits bestehenden Objekts ergeben.

Außer bei Bildern und Textlisten verhält sich SiVArc bei Namenskonflikten wie folgt:

Wenn ein manuell angelegtes HMI-Objekt mit einem von SiVArc zu generierenden Namen bereits existiert, erhält das bestehende Objekt das Suffix "_renamed". Wenn auch dieser Name bereits vergeben ist, wird der Name automatisch hochgezählt.



Beispiel: Im Projekt wurde manuell ein Symbolisches E/A-Feld mit einer Textliste verschaltet. Über die Textlistenregeln wird dann mit SiVArc eine namensgleiche Textliste erstellt. Die bestehende, manuell verschaltete Textliste wird umbenannt. Die Verschaltung bleibt bestehen.

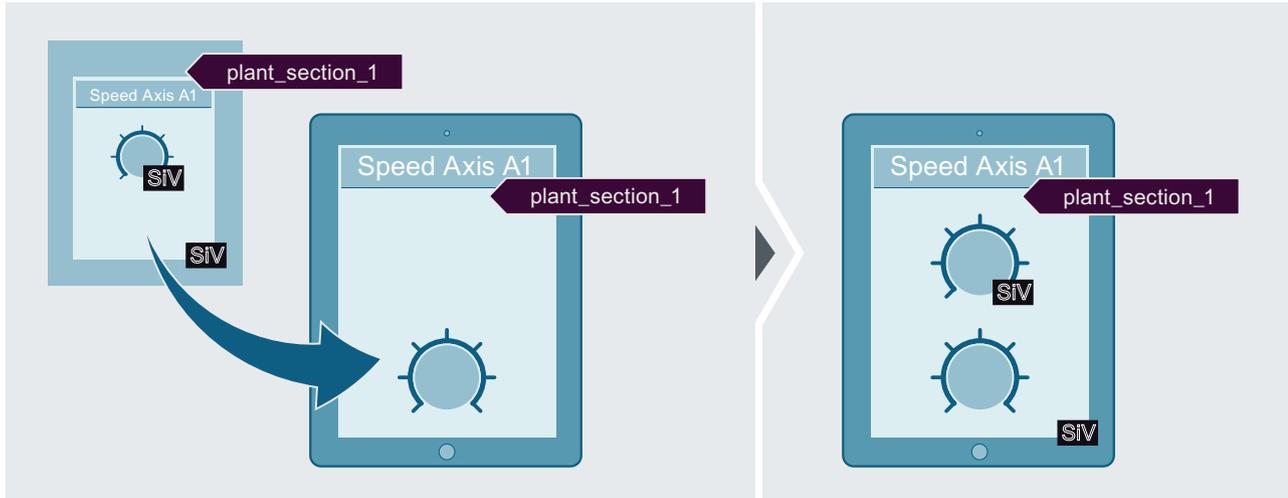
Stellen Sie von Anfang an sicher, dass sich die Namenskonzepte für die generierten HMI-Objekte und für manuell erstellte HMI-Objekte unterscheiden. Auf diese Weise vermeiden Sie nachträgliche Korrekturen.

Namensgleichheit bei Bildern und Textlisten

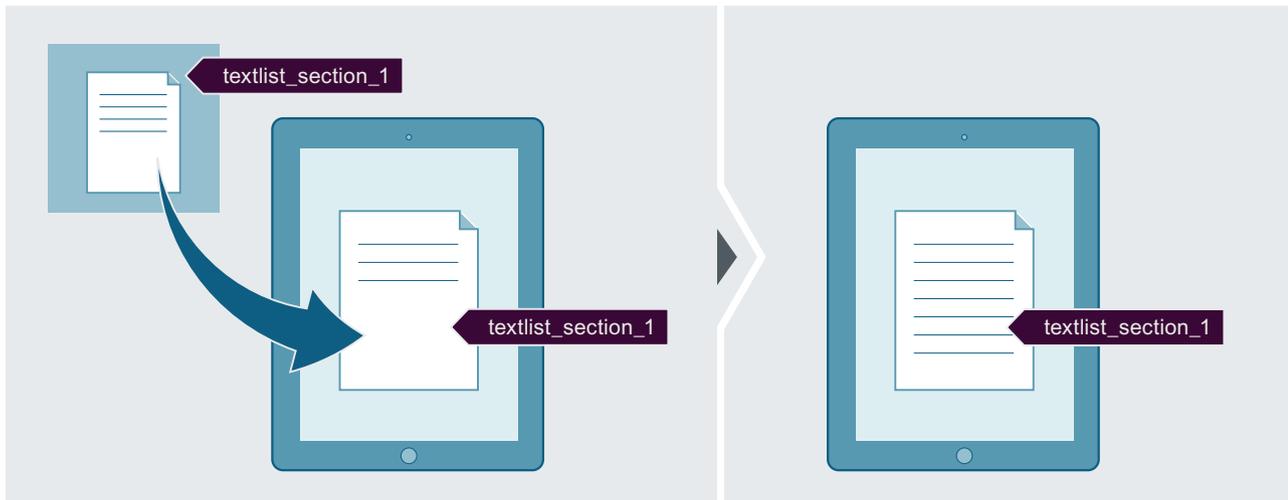
Bei Bildern und Textlisten verarbeitet SiVArc bei Namensgleichheit anders:

Wenn bereits generierte Bilder oder Textlisten mit den gleichen Namen vorhanden sind, erfasst SiVArc diese Bilder oder Textlisten trotz des Namenskonflikts bei der Generierung. Beachten Sie, dass dabei für Textlisten eine Fehlermeldung ausgegeben wird, für Bilder jedoch nicht.

Die folgende Darstellung zeigt die Generierung von Bildern bei Namensgleichheit mit einem manuell angelegten Bild. Das bestehende Bild wird von der Generierung verarbeitet:



Die folgende Darstellung zeigt die Generierung von Textlisten bei Namensgleichheit mit einer manuell angelegten Textliste. Die Textlisteneinträge der beiden Textlisten werden in der bestehenden Textliste zusammengeführt:



Priorität der generierten Objekte bei Namenskonflikten

Ein generiertes Objekt steht in einem SiVArc-Projekt über einem manuell angelegten HMI-Objekt. Das generierte Objekt hat einen festen Bezug zu SiVArc und wird bei jeder Generierung aktualisiert. Manuelle Änderungen daran werden zurückgesetzt.

Bei Namenskonflikten priorisiert SiVArc während einer Generierung wie folgt:

1. Generierte Objekte aus Bild-, Variablen- und Textlistenregeln
2. Generierte Objekte aus Kopierregeln
Aus den Kopierregeln generierte Objekte werden wie manuell erstellte Objekte behandelt. Sie werden bei der Generierung zuerst erzeugt. Bei Namenskonflikten mit nachfolgend generierten Objekten werden Objekte aus Kopierregeln mit dem Anhang "_renamed" umbenannt.
3. Manuell erstellte Objekte
Bei Namensgleichheit von manuell erzeugten Objekten und generierten Objekten werden die manuell erzeugten Objekte umbenannt.

Namenskonflikte durch mehrere PLCs im Projekt

Wenn beim Generieren auf Basis mehrerer verbundener PLCs Namenskonflikte entstehen, generiert SiVArc nur das erste von der Generierung erfasste HMI-Objekt und gibt eine Fehlermeldung aus.

Um in einem Projekt mit mehreren PLCs eine eindeutige Zuweisung der Variablen zu ermöglichen, verwenden Sie in den Runtime-Einstellungen für Variablen die Option "PLC Name als Präfix im HMI-Variablennamen verwenden".

Bei der Generierung von Textlisteneinträgen für mehrere PLCs können Namenskonflikte entstehen, weil ein Programmbaustein in mehreren PLCs vorhanden sein kann. Abhängig davon, welche Textquelle aus STEP 7 Sie verwenden, verhält sich die Generierung von Textlisteneinträgen in diesen Konfliktfällen unterschiedlich:

- Bausteinparameter
Zusätzliche Textlisteneinträge werden mit einem Namenszusatz generiert.
- Netzwerk
Textlisteneinträge werden nur für die zuerst ausgewertete PLC erstellt. Zu generierende Textlisteneinträge für alle nachfolgenden PLCs werden ignoriert. Der Fehler wird in einer Meldung und im Protokoll angezeigt.

Namenskonflikte beim Import von Regeln

Um SiVArc-Regeln zu importieren, stehen Ihnen folgende Optionen zur Verfügung.

- Bestehende Regeln durch Importieren überschreiben
Gleichnamige Regeln und Regelgruppen werden aktualisiert. Sonstige Regeln bleiben erhalten.
- Zu importierende Regeln umbenennen, falls Regelname schon vergeben
Bei Namenskonflikten werden die Namen der importierten Regeln und Regelgruppen um eine fortlaufende Nummer erweitert.
- Alle bestehenden Regeln vor dem Import löschen
Nach dem Import enthält der Regeleditor nur noch die Regeln aus der Importdatei.

Zweck und Nutzen

Wenn Sie Namenskonflikte bei der Projektierung mit SiVArC im Vorfeld vermeiden, erhalten Sie ein konsistentes Projekt mit hoher Fehlersicherheit. Auf Basis eines gelungenen Konzepts zur eindeutigen Abgrenzung der Namenskonzepte können Sie mit wenig Aufwand weitere Automatisierungsprojekte ableiten.

Siehe auch

SiVArC-Regeln exportieren und importieren (Seite 201)

6.3.10 Variablenregel anlegen

Einleitung

Abhängig von Ihren Einstellungen generiert SiVArC alle externen Variablen oder nur die für das SiVArC-Projekt relevanten Variablen. SiVArC generiert externe Variablen für Instanz-Datenbausteine und globale Datenbausteine.

In einer Variablenregel legen Sie Folgendes fest:

- Name des Ordners, in dem eine generierte Variable abgelegt wird
- Name der Variablen-tabelle, in der eine generierte Variable angelegt wird

Voraussetzung

- Ein Funktionsbaustein mit Datenbaustein ist angelegt.
- Für die relevanten PLC-Variablen ist an der Bausteinschnittstelle die Option "Erreichbar aus HMI" gesetzt.
- Der Editor "Variablenregeln" ist geöffnet.

Vorgehen

Um eine Variablenregel anzulegen, gehen Sie folgendermaßen vor:

1. Legen Sie eine Variablenregel an.
2. Benennen Sie die Regel eindeutig.
3. Öffnen Sie unter "Variablen-gruppe" den Editor "SiVArC-Ausdrücke".
 - Um den Namen der Variablen-gruppe als Text zu definieren, tragen Sie eine Zeichenkette in Anführungszeichen ein.
 - Um den Namen aus dem Anwenderprogramm abzuleiten, geben Sie einen SiVArC-Ausdruck ein.
 - Um die Struktur des Anwenderprogramms in der Projektnavigation abzubilden, geben Sie den SiVArC-Ausdruck `HmiTag.DB.FolderPath` ein.

4. Öffnen Sie unter "Variablentabelle" den Editor "SiVArc-Ausdrücke".
 - Legen Sie einen Text oder einen SiVArc-Ausdruck als Namen der Variablentabelle fest.
 - Um alle Variablen eines Datenbausteins in eine Variablentabelle zu schreiben, geben Sie den SiVArc-Ausdruck `HmiTag.DB.SymbolicName` an.
5. Um bei Bedarf eine Bedingung einzugeben, verwenden Sie SiVArc-Scripting.

Ergebnis

Eine Variablenregel ist angelegt.

Weitere Einstellungen

Um die Variablengenerierung zu steuern, optimieren Sie folgende Einstellungen:

- Modus der Variablengenerierung unter "Extras > Einstellungen > SiVArc"
- Runtime-Einstellungen für Variablen
- Aktualisierungszyklus und die Erfassungsart

Anordnung von Variablenregeln ändern

Variablenregeln ordnen Sie per Drag&Drop oder über die Befehle des Kontextmenüs an. Diese Funktionalität steht nur zur Verfügung, wenn die Spalten des Editors "Variablenregeln" weder sortiert noch gefiltert sind. Über das Kontextmenü sortieren Sie auch im gefilterten Editor "Variablenregeln" Regeln um.

Um die Anordnung von Variablenregeln über Drag&Drop zu ändern, gehen Sie folgendermaßen vor:

1. Markieren Sie die erste Zelle der Regel.
2. Ziehen Sie die Regel per Drag&Drop an die gewünschte Position im Editor.

Siehe auch

Generieren von Variablen (Seite 173)

Prinzip der Variablengenerierung (Seite 181)

Beispiel: Variablennamen anpassen (Seite 197)

Ändern von SiVArc-Regeln (Seite 212)

Ablagestrategien für generierte Objekte (Seite 126)

Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)

6.3.11 Bildregel anlegen

Voraussetzung

- Das Anwenderprogramm ist erstellt.
- Die Generiervorlage eines Anzeige- und Bedienobjekts ist erstellt.
- Die Generiervorlage eines Bildes ist erstellt.
- Der Editor "Bildregeln" ist geöffnet.
- Die Spalten "PLC" und "HMI-Gerät" sind eingeblendet.

Vorgehen

Um eine Bildregel für das Generieren eines Anzeige- und Bedienobjekts zu definieren, gehen Sie folgendermaßen vor:

1. Legen Sie eine Bildregel an.
2. Benennen Sie die Regel eindeutig.
3. Wählen Sie unter "PLC" die Steuerungen, für welche die Bildregel gelten soll.
Wenn Sie keine Steuerung auswählen, gilt die Bildregel für alle Steuerungen im Projekt.
4. Wählen Sie den Programmbaustein aus, für den das HMI-Objekt generiert wird.
5. Wählen Sie unter "Bildobjekt" die Generiervorlage des Anzeige- und Bedienobjekts aus.
6. Wählen Sie unter "Bild" die Generiervorlage des Bildes aus, in welches das Objekt generiert wird.
Wenn der Generiervorlage ein Positionierungsschema hinterlegt ist, wählen Sie unter "Positionierungsfeld" den Platzierungsbereich. Wenn Sie keinen Platzierungsbereich festlegen, wird das generierte HMI-Objekt nach dem SiVArc-Positionierungsschema im Bild platziert.
7. Wählen Sie unter "HMI-Gerät" die Bediengeräte, für welche die Bildregel gelten soll. Über die Symbole der Werkzeugleiste können Sie die Gerätetypen der HMI-Geräte einblenden.
Wenn Sie kein Bediengerät auswählen, gilt die Bildregel für alle Bediengeräte, die mit der gewählten Steuerung verbunden sind.
Wenn die Regel nur für Objekte oder Programmbausteine ausgeführt werden soll, die eine bestimmte Bedingung erfüllen, programmieren Sie unter "Bedingung" mit SiVArc-Scripting den entsprechenden Ausdruck.

Sie können die Programmbausteine und Vorlagen auch per Drag&Drop aus der Bibliothek einfügen.

Ergebnis

Wenn Sie die Visualisierung generieren, wird im angegebenen Bild das Objekt generiert.

Wenn Sie einen Platzierungsbereich in der Bildregel gewählt haben, wird das HMI-Objekt anstelle eines Positionierungsfeldes innerhalb dieses Bereichs platziert. Welches Positionierungsfeld verwendet wird, ist hängt ab von der Generierreihenfolge der Bildregeln und dem Index des Positionierungsfeldes.

Siehe auch

Visualisierung generieren (Seite 215)
Unterstützte Objekte im Anwenderprogramm (Seite 106)
Beispiel: Bildregel mit Bedingung anlegen (Seite 194)
Beispiel: Bild- und Textlistenregel organisieren (Seite 195)
SiVArc-Regeln (Seite 161)
Erstellen von SiVArc-Regeln (Seite 165)
Verarbeitung von Regeln (Seite 170)
Ändern von SiVArc-Regeln (Seite 212)
Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)

6.3.12 Textlistenregel anlegen

Einleitung

Mit einer Textlistenregel legen Sie fest, welche Textliste für einen Programmbaustein generiert wird.

Voraussetzung

- Das Anwenderprogramm ist erstellt.
- Eine Generiervorlage der Textliste ist in einer Bibliothek im entsprechenden Ordner abgelegt.
- Der Editor "Textlistenregeln" ist geöffnet.

Vorgehen

Um eine Textlistenregel zu definieren, gehen Sie folgendermaßen vor:

1. Legen Sie eine Textlistenregel an.
2. Benennen Sie die Regel eindeutig.
3. Wählen Sie den gewünschten Programmbaustein aus.
4. Wählen Sie die gewünschte Generiervorlage einer Textliste aus.
5. Um bei Bedarf eine Bedingung einzugeben, verwenden Sie SiVArc-Scripting.

Sie können die Programmbausteine oder Kopiervorlagen auch per Drag&Drop einfügen.

Ergebnis

Wenn Sie die Visualisierung generieren, wird die Textliste im Editor "Text- und Grafiklisten" angelegt.

Siehe auch

- Beispiel: Bild- und Textlistenregel organisieren (Seite 195)
- Beispiel: Generiervorlagen für Textlisten erstellen (Seite 142)
- Beispiel: Generiervorlage für eine Textliste für Bausteinparameter erstellen (Seite 145)
- SiVArc-Regeln (Seite 161)
- Erstellen von SiVArc-Regeln (Seite 165)
- Verarbeitung von Regeln (Seite 170)
- Ändern von SiVArc-Regeln (Seite 212)
- Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)
- Melderegeln erstellen (Seite 190)

6.3.13 Melderegeln erstellen

Einleitung

Melderegeln sind integraler Bestandteil der SiVArc-Generierung. Beim Erstellen von Melderegeln programmieren Sie die PLC-Bausteine mit STEP 7 so, dass durch PLC-Bausteine Objekte in Bediengeräten erstellt werden. Auf diese Weise fungieren Melderegeln als abstrakter Typ beim Verbinden von STEP 7 mit Bediengeräten.

Voraussetzung

- Globale Bibliothek und Projektbibliothek mit Kopiervorlagen und Typen.
- Die generierten HMI-Objekte werden in der Bibliothek abgelegt.

Vorgehensweise

Um in SiVArc Melderegeln zu erstellen, gehen Sie wie folgt vor:

1. Klicken Sie im Ordner SiVArc auf "Melderegeln".
2. Um eine neue Regel hinzuzufügen, klicken Sie im Editor "Melderegeln" auf "Neue Regel erstellen". Standardmäßig wird als erster Regelname eine Melderegel angezeigt. Anschließende Regelnamen werden mit einer nachgestellten laufenden Nummer wie <Melderegel_1> angezeigt.
3. Suchen Sie in der Spalte "Regel-Trigger" einen entsprechenden Programmbaustein oder Programmtyp.
4. Suchen Sie in der Spalte "Kopiervorlage von Meldungen/Klassen/Gruppen" nach einem einzelnen oder mehreren Objekten für die Kopiervorlage.
5. Legen Sie in der Spalte "Bedingung" eine Bedingung für die auszuführende Regel fest. Die Bedingung ist in SiVArc optional.

Hinweis

- Ähnliche, ebenfalls projektierte Regeln werden nicht verarbeitet.
 - Wählen Sie im Programmbaustein den Kontextmenübefehl "Gehe zu referenziertem Objekt". Der ausgewählte Programmbaustein erscheint hervorgehoben im Bereich "Regel-Trigger". Wählen Sie im Editor "Plug-Ins" die Option "Melderegeln" aus. Daraufhin wird eine gefilterte Liste mit Melderegeln für einen Baustein angezeigt.
 - "Gehe zu referenziertem Objekt" ist verfügbar, wenn Sie in der Spalte "Kopiervorlage von Meldungen/Klassen/Gruppen" eine Einfachselektion vorgenommen haben.
-

Ergebnis

Bei der SiVArc-Generierung werden die Meldeobjekte nur dann erstellt, wenn die Bedingung erfüllt ist und die Regel erfolgreich verarbeitet wurde. Die generierten Objekte können Sie in der "Generierübersicht" ansehen. Näheres zur Generierübersicht finden Sie unter dem Thema Visualisierung generieren (Seite 205)

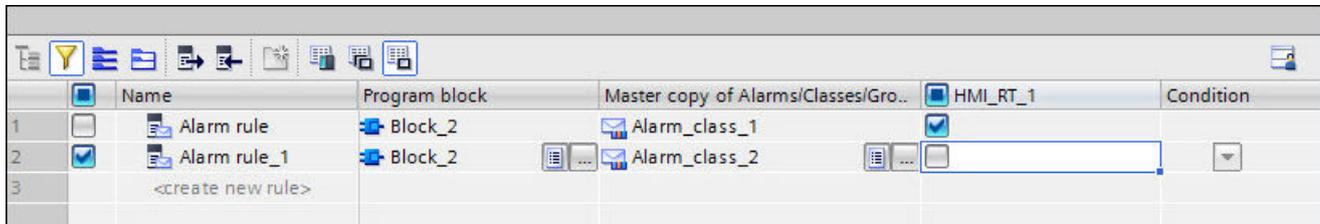
Allgemeine Hinweise

- Die im Editor "Melderegeln" erstellten Regeln bleiben auch nach dem Schließen und erneuten Öffnen des Editors gespeichert.
- Im Editor "Melderegeln" können Sie die Aktionen "Kopieren", "Einfügen", "Ziehen" und "Ablegen" ausführen und rückgängig machen.
- Wenn Sie beim Bearbeiten der Optionen im Editor "Melderegeln" Objekte ändern oder löschen, zeigt das referenzierte Objekt im Editor "Melderegeln" durch eine rosa Einfärbung des Bereichs einen ungültigen Bezug an.
- Sie können im Editor "Melderegeln" auch Regeln zu einer Gruppe zusammenfassen. Dazu wählen Sie mehrere Regeln aus und wählen nach Rechtsklick auf die Meldungen "Neue Regelgruppe hinzufügen".
- Neue Melderegeln können Sie innerhalb der "Regelgruppe" hinzufügen.
- Bei der Gruppierung von Melderegeln wird automatisch die Bedingung AND angezeigt. Aus der Auswahlliste für den Bediener können Sie auch einen anderen Bedingungsoperator wählen. Oder Sie können eine Bedingung für eine Gruppe manuell festlegen.
- Bei der Auswahl der Programmbausteine und Kopiervorlagen unterstützt SiVArc IntelliSense.

Gerätespezifische Regeln

Gerätespezifische Melderegeln sind Melderegeln, die spezifisch für ein Bediengerät gelten. Der Editor "Melderegeln" bietet Ihnen in der Symbolleiste die Option  zum Ein- oder Ausblenden des Bediengeräts. Gehen Sie zum Projektieren von gerätespezifischen Regeln wie folgt vor:

1. Klicken Sie auf die Option zum Ein- oder Ausblenden von Bediengeräten . Die verfügbaren Bediengeräte werden jeweils zusammen mit einem Kontrollkästchen angezeigt.
2. Hiermit können Sie ein Bediengerät in Bezug auf eine bestimmte Meldung auswählen oder abwählen. Dadurch gelten die Melderegeln bei der SiVArC-Generierung nur für die ausgewählten Bediengeräte.



	Name	Program block	Master copy of Alarms/Classes/Gro..	HMI_RT_1	Condition
1	Alarm rule	Block_2	Alarm_class_1	<input checked="" type="checkbox"/>	
2	Alarm rule_1	Block_2	Alarm_class_2	<input type="checkbox"/>	
3	<create new rule>				

Hinweis

- Standardmäßig ist das erste verfügbare Bediengerät ausgewählt.
- Sie können auch die PLCs auswählen oder abwählen.

Auswahl des Gerätetyps über die Option "Eigenschaften"

Um die Auswahl von PLC-/Bediengeräten aus einer Liste mit solchen Geräten zu vereinfachen, verwenden Sie die Registerkarte "Eigenschaften" im Editor "Melderegeln". Die Registerkarte "Eigenschaften" im Editor "Melderegeln" bietet Ihnen Optionen, um PLC-/Bediengeräte für eine bestimmte Regel aus- oder abzuwählen. Die betroffenen Melderegeln gelten dann nur für die ausgewählten PLC-/Bediengeräte.

Siehe auch

Visualisierung generieren (Seite 205)

6.3.14 Kopierregel anlegen

Einleitung

Bei standardisierten Lösungen für das Bedienen und Beobachten werden HMI-Objekte oftmals zentral erstellt und dann als globale Bibliotheken an die Projektueure verteilt.

Voraussetzung

- Globale Bibliothek mit Typen und Kopiervorlagen.
- Die zu generierenden HMI-Objekte werden in der Bibliothek abgelegt.

Vorgehensweise

Gehen Sie wie folgt vor, um in SiVArc mithilfe von Kopierregeln HMI-Objekte anzulegen:

1. Öffnen Sie die globale Bibliothek mit den Kopiervorlagen und Typen.
2. Synchronisieren Sie den Inhalt der geöffneten globalen Bibliothek mit der Projektbibliothek.
3. Erstellen Sie für jedes zu generierende HMI-Objekt eine Kopierregel.
Oder:
Erstellen Sie für einen Bibliotheksordner eine Kopierregel.
4. Vergeben Sie für die Regeln einen eindeutigen Namen.

Ergebnis

Die HMI-Objekte werden bei der Generierung im entsprechenden Ordner der Projektnavigation erstellt. Die HMI-Objekte werden für jedes in einer Regel angegebene Bediengerät erstellt.

Allgemeine Hinweise

- Wenn bei der SiVArc-Generierung der Name des standardmäßigen Meldeobjekts im Bediengerät und das von den Kopierregeln generierte Objekt identisch sind, zeigt die Anwendung folgende Fehlermeldung an:
"Objekt <<Name_der_Meldung>> wurde in einem anderen SiVArc-Editor geändert. Der Generator für Copy_rule kann dieses Objekt nicht ändern."
- Im Editor für Kopierregeln können Benutzer die Regel, die für Bediengeräte generiert werden muss, auswählen oder abwählen.
- Sie können die entsprechenden Objekte per Drag & Drop von der Kopiervorlage in die Spalte "Bibliotheksobjekt" ziehen.
- Der Editor für Kopierregeln unterstützt IntelliSense.

Hinweis

Kopierregeln für Meldungen

Die mithilfe des Editors für Kopierregeln projizierten Regeln werden verarbeitet und generieren bei der SiVArc-Generierung HMI-Objekte.

Siehe auch

Einsatz von Kopierregeln (Seite 177)

Ändern von SiVArc-Regeln (Seite 212)

Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)

6.3.15 Beispiel: Bildregel mit Bedingung anlegen

Beispielsszenario

In einem standardisierten Anwenderprogramm wird zur Steuerung eines Ventils oder eines Motors der gleiche Programmbaustein eingesetzt.

Anforderung

Abhängig vom Einsatz des Programmbausteins soll in der Bedienoberfläche eine Schaltfläche mit der Beschriftung "Open Valve" oder "Start Motor" generiert werden. Die Schaltfläche für das Ventil soll immer am oberen Bildrand platziert werden, die Schaltfläche für den Motor am unteren Bildrand.

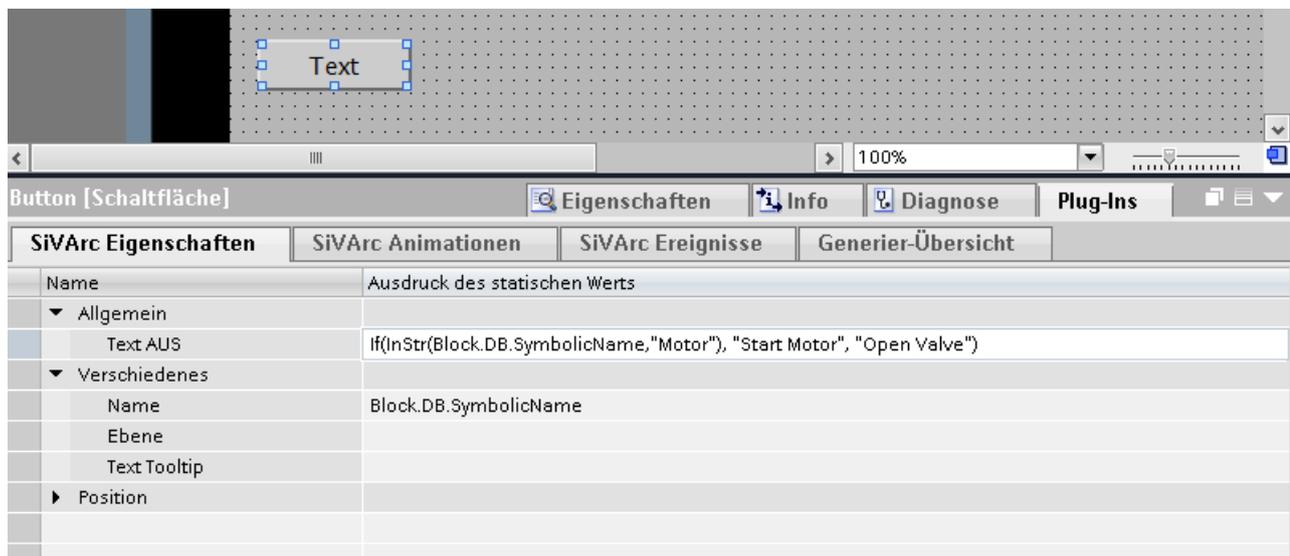
Die Bildregeln sollen so gesteuert werden, dass für jeden Anwendungsfall des Programmbausteins eine andere Schaltfläche generiert wird.

Umsetzungskonzept

Der Projektteur steuert über Bedingungen die Beschriftung der Schaltflächen und die Bildregeln. Für jeden Anwendungsfall des Programmbausteins generiert SiVArc eine andere Schaltfläche. Dazu greift der Projektteur mit den Bedingungen auf die symbolischen Namen der Datenbausteine der Bausteininstanzen zurück:

- Wenn der Programmbaustein zur Ventilsteuerung genutzt wird, heißt der Datenbaustein "Valve_DB_Instance_<n>".
- Wenn der Programmbaustein zur Motorsteuerung genutzt wird, heißt der Datenbaustein "Motor_DB_Instance_<n>".

Um die Beschriftung der Schaltfläche zu steuern, programmiert der Projektteur in der SiVArc-Eigenschaft "Text AUS" eine Bedingung:

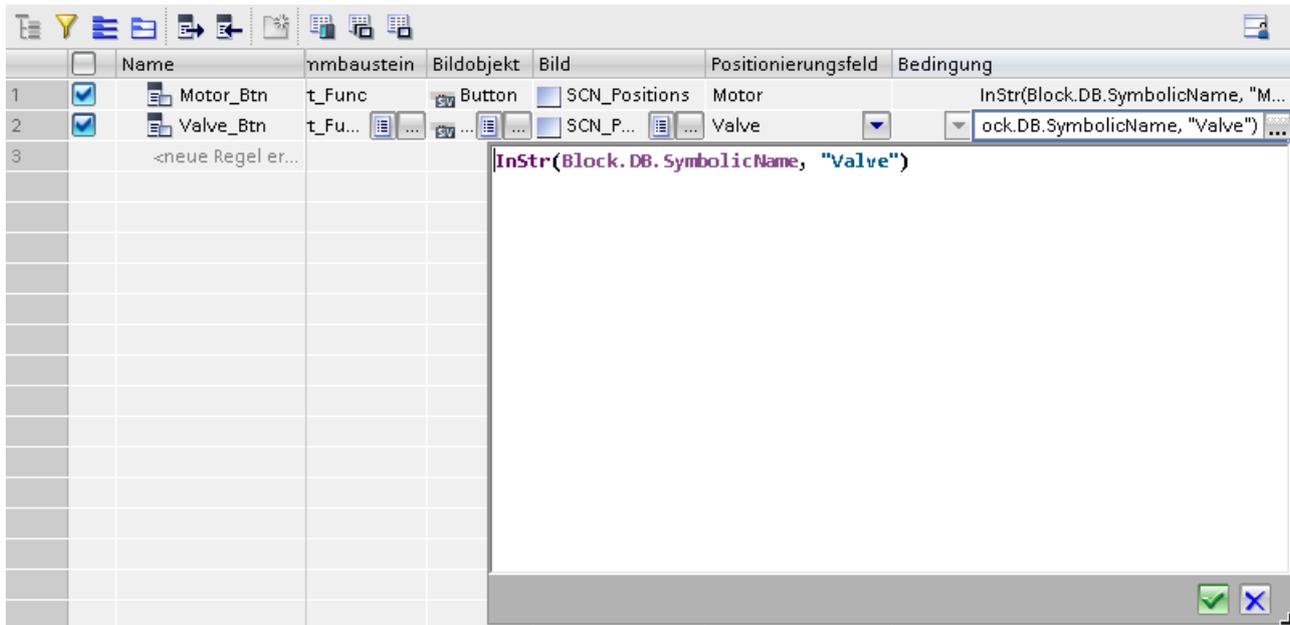


The screenshot shows the SiVArc software interface. At the top, a 'Text' button is visible on a grid. Below it, the 'Eigenschaften' (Properties) panel is open, showing the 'SiVArc Eigenschaften' (SiVArc Properties) section. The 'Text AUS' property is highlighted, showing the following condition:

Name	Ausdruck des statischen Werts
▼ Allgemein	
Text AUS	if(InStr(Block.DB.SymbolicName,"Motor"), "Start Motor", "Open Valve")
▼ Verschiedenes	
Name	Block.DB.SymbolicName
Ebene	
Text Tooltip	
▶ Position	

Um die Positionierung der Schaltflächen zu steuern, erstellt der Projektteur ein Positionierungsschema mit den Positionierungsfeldern "Valve 1/1" und "Motor 1/1". Das Schema hinterlegt er in der Generiervorlage für das Anlagenbild und legt die beiden nachfolgenden Bildregeln an.

Um die Ausführung der Bildregeln zu steuern, programmiert er eine Bedingung. Die Bedingung legt fest, dass die Regel nur ausgeführt wird, wenn im symbolischen Namen des Instanz-Datenbausteins die Zeichenkette "Valve" bzw. "Motor" enthalten ist.



Siehe auch

Bildregel anlegen (Seite 188)

6.3.16 Beispiel: Bild- und Textlistenregel organisieren

Beispielsszenario

In einem Ingenieurbüro arbeiten mehrere Projektueure an einem Großprojekt. Jeder Ingenieur hat seinen eigenen Aufgabenbereich und sein eigenes Spezialgebiet.

Anforderung

Die Bearbeitung der Bildregeln soll getrennt nach Funktionalität erfolgen. Die Bedienung und Darstellung von Rezepturen wird einem Projektueur zugeordnet. Ein anderer Kollege richtet alle Diagnosefunktionen ein.

Diese Funktionen sollen beim Generieren je nach Bedarf gesammelt aktiviert oder deaktiviert werden.

Das Layout der Bediengeräte wird von einer einzelnen Kollegin gesteuert. Sie soll sich schnell einen Überblick über die verwendeten Bediengeräte machen können.

Umsetzungskonzept

In den Regeleditoren wird eine funktionsspezifische Ordnerstruktur angelegt, z. B.

- Startbilder
 - ComfortPanel 19"
 - ComfortPanel 19" Porträt
 - MobilePanel 277 8"
 - PC Station
- Diagnosebilder
 - ...
 - ...
- Rezepturbilder
 - ...
 - ...

Um die Bild- und Textlistenregeln zu prüfen, blendet der Steuerungsprogrammierer über die Werkzeugleiste nur die relevanten Spalten im Regeleditor ein. Innerhalb der Ordner sortiert er die Regeln nach der Spalte "Programmbaustein" und filtert nach dem relevanten Funktionsbaustein.

Um das Projekt anlagenweise zu generieren, werden die Hauptgruppen gesammelt aktiviert oder deaktiviert.

Um das Projekt geräteweise zu generieren, werden die jeweiligen Gruppen für Bediengerättypen aktiviert oder deaktiviert.

Den Projekteuren werden über die Kommentarspalte Gruppen zugewiesen, welche die Regeln für ihr Fachgebiet enthalten. Der Bearbeiter kopiert sich seine Gruppe in ein Testprojekt. Er bearbeitet die bestehenden Regeln und erstellt neue Regeln. Nach Freigabe durch die technische Leitung werden die Regeln gruppenweise wieder importiert und in anderen Projekten wiederverwendet.

Die für das Layout zuständige Projektleiterin vergibt die Vorlagen für die entsprechenden Bediengeräte pro Bildart.

Siehe auch

Bildregel anlegen (Seite 188)

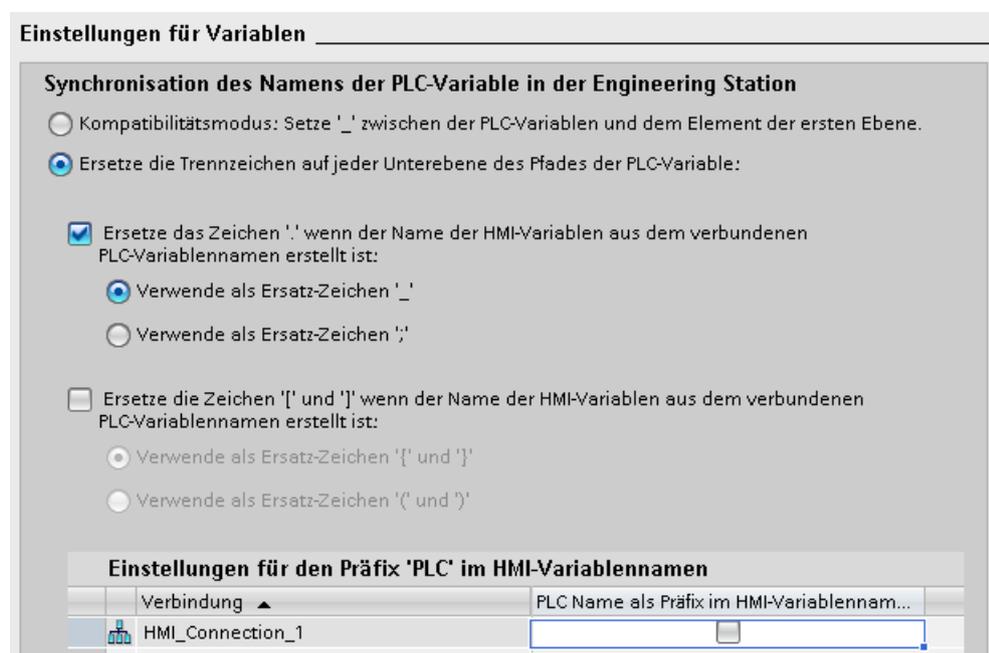
Textlistenregel anlegen (Seite 189)

6.3.17 Beispiel: Variablennamen anpassen

Beispielsszenario

Im Steuerungsprogramm kommt ein neuer, strukturierter PLC-Datentyp zum Einsatz. Um die Synchronisation der PLC-Variablen mit den externen HMI-Variablen sicherzustellen, ändert der Projektteur die Einstellungen für Variablen an einem Bediengerät. Die Runtime-Einstellungen für Variablen unterscheiden sich dadurch für die Bediengeräte innerhalb eines Projekts:

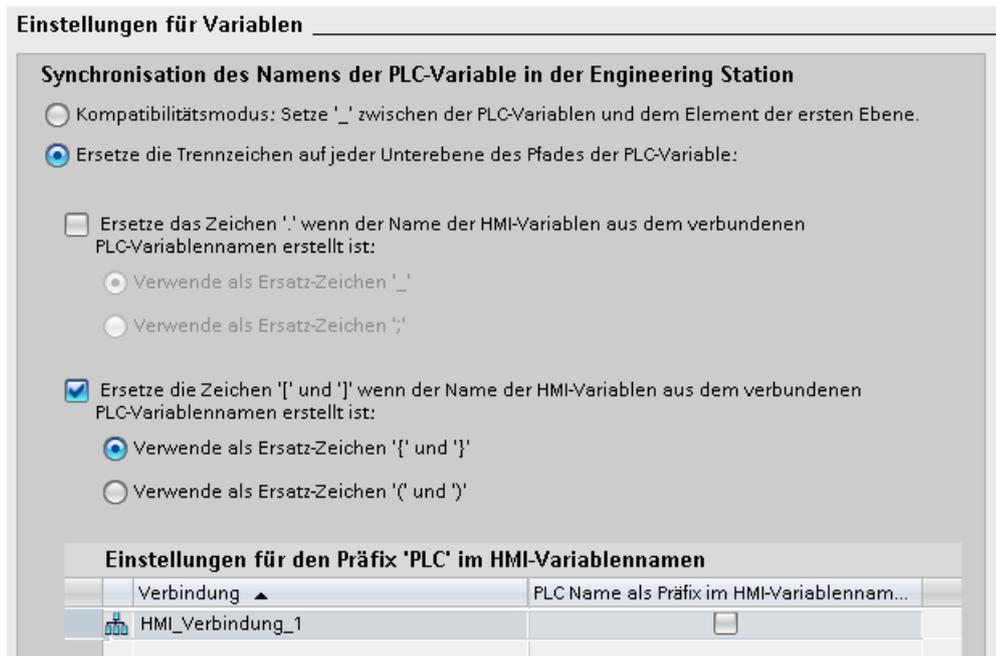
Runtime-Einstellungen für Bediengerät 1:



Synchronisierte HMI-Variablen:

Plantsection1	
	Name
	Plantsection1_DB_Activate
	Plantsection1_DB_Productionline_Instance_1_Container_Position
	Plantsection1_DB_Productionline_Instance_1_Container_Position(1)
	Plantsection1_DB_Productionline_Instance_1_Dispatchunit_Instance_1_Conveyor_Instance_Speed

Runtime-Einstellungen für Bediengerät 2:



Synchronisierte HMI-Variablen

Plantsection1	
Name	
<	Plantsection1_DB.Activate
<	Plantsection1_DB.Activate_Renamed
<	Plantsection1_DB.Productionline_Instance_1.Container_Position
<	Plantsection1_DB.Productionline_Instance_1.Container_Position_Renamed
<	Plantsection1_DB.Productionline_Instance_1.Dispatchunit_Instance_1.Conveyor_Instance.Speed

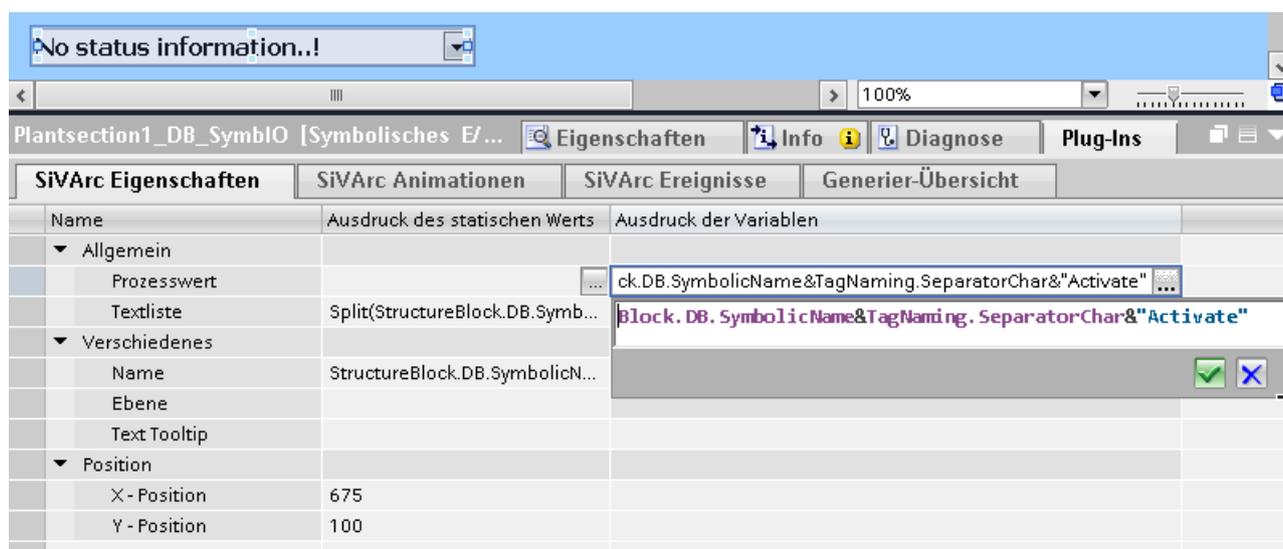
Anforderung

Die Verschaltungen der generierten Anzeige- und Bedienobjekte sollen auch funktionieren, wenn innerhalb eines Projekts unterschiedliche Synchronisationsmodi verwendet wurden.

Umsetzungskonzept

Im SiVArC-Ausdruck, der die Triggervariable definiert, wird die gewählte Variablensynchronisation über das SiVArC-Objekt `TagNaming` adressiert.

Der folgende SiVArC-Ausdruck ergibt nach der Generierung der Visualisierung den Variablennamen: "Plantsection1_DB.Activate" für Bediengerät 1 und "Plantsection1_DB_Activate" für Bediengerät 2.



Siehe auch

- TagNaming (Seite 249)
- Prinzip der Variablengenerierung (Seite 181)
- SiVArc-Ausdruck (Seite 109)

6.3.18 SiVArc-Regeln bearbeiten und verwalten

Einleitung

In komplexen SiVArc-Projekten gibt es eine große Zahl von SiVArc-Regeln. Sie müssen die SiVArc-Regeln deshalb übersichtlich sortieren und strukturieren und sie über die Bibliothek zur Verfügung stellen.

Für die übersichtliche Anzeige der Regeln werden mehrere Funktionen bereitgestellt:

- Filterfunktion
- Gruppier- und Sortierfunktion
- Kontextmenüs
- Drag & Drop

Um die Regeln auszuwerten, navigieren Sie mithilfe der "Gehe zu"-Befehle des Kontextmenüs zwischen den SiVArc-Editoren, dem Anwenderprogramm und den Generiervorlagen.

SiVArc-Regel erstellen

1. Klicken Sie auf "Regel hinzufügen".
Im Tabelleneditor wird eine neue Zeile erstellt.
2. Vergeben Sie einen eindeutigen Namen für die Regel.
3. Fügen Sie die Programmbausteine und Generiervorlagen aus der Bibliothek mit Drag & Drop ein.

Alternativ können Sie auch die ersten Buchstaben des Objekts eingeben, das Sie referenzieren wollen. SiVArc zeigt daraufhin eine Liste mit referenzierbaren Objekten an, die diese Buchstabenfolge im referenzierten Pfad enthalten.

Wenn Sie per Drag & Drop unter "Name" einen neuen Programmbaustein einfügen, wird mit dem ausgewählten Programmbaustein eine neue Regel erstellt.

Gruppieren von SiVArc-Regeln

Wenn Sie SiVArc-Regeln nach Ihren eigenen Kriterien gruppieren, erhalten Sie einen besseren Überblick über Ihr SiVArc-Projekt:

- Die in Regelgruppen zusammengefassten Regeln werden zusammen aktiviert oder deaktiviert.
- Die für eine Regelgruppe geltenden Bedingungen gelten für alle Regeln innerhalb der Gruppe. Die Sonderfälle richten Sie mithilfe von Operanden ein.
- Sie können auch einzelne Regeln nach Bedarf innerhalb und außerhalb von Gruppen verschieben und anordnen.
- Beim Verschieben von Regeln zwischen Gruppen werden jeweils die für die aktuelle Gruppe geltenden Optionseinstellungen angewandt.

Gehen Sie zum Erstellen einer Regelgruppe wie folgt vor:

1. Wählen Sie die Regeln aus, für die eine Gruppe erstellt werden soll.
2. Wählen Sie im Kontextmenü "Neue Regelgruppe hinzufügen".
Die ausgewählten Regeln werden in eine neue Gruppe verschoben.
3. Benennen Sie die Regelgruppe.

Um eine Untergruppe zu erstellen, bearbeiten Sie die gewünschten Regeln auf dieselbe Weise innerhalb einer Gruppe.

Um alle Regelgruppen gleichzeitig zu öffnen oder zu schließen, klicken Sie auf die Schaltfläche "Alles einblenden" oder "Alles ausblenden".

Hinweis

Filtern über Regelgruppen

Eine Gruppe wird bei einem Filtervorgang erst erstellt, wenn alle Regeln der Gruppe die Filterbedingung erfüllen.

Verschachtelte Gruppierung von SiVArc-Regeln

Für eine Regelgruppe können Bedingungen festgelegt werden. Mithilfe von Regelgruppen sortieren Sie die SiVArc-Regeln nach Bedarf, beispielsweise entsprechend der Anlagenstruktur, Bildstruktur oder nach Themenbereichen von WinCC.

Anwendungsbeispiel für Regelgruppen

In einem SiVArc-Projekt sind alle Bildregeln nach folgenden Bildtypen in Gruppen geordnet:

- Startbilder
- Diagnosebilder
- Rezeptbilder

Dadurch können Sie z. B. die SiVArc-Projektierung themenspezifisch an Projekteure einer Abteilung vergeben.

Über Gruppenbedingungen legen Sie z. B. fest, welche Variablen in einem Programmbaustein enthalten sein müssen, damit die betreffende Regelgruppe von der Generierung erfasst wird.

Über die Bedingungsoperanden einer Regel innerhalb der Gruppe generieren Sie je nach den enthaltenen Parametern verschiedenartigste Bilder. Auf diese Weise visualisieren Sie mit einem SiVArc-Projekt und wenigen Regeln viele Anlagenbereiche.

SiVArc-Regeln in einer Bibliothek verwenden

Um SiVArc-Regeln zentral und projektübergreifend einheitlich aktualisieren zu können, legen Sie die SiVArc-Regeln oder -Regelgruppen als Kopiervorlage in einer Bibliothek ab. Wenn im Projekt bereits eine SiVArc-Regel mit demselben Namen vorhanden ist, können Sie die Regel überschreiben oder eine neue Regel erstellen.

Wenn Sie eine Regel mit einer Regel aus der Bibliothek überschreiben, gilt Folgendes:

- SiVArc erkennt die in einem früherem Generiervorgang generierten HMI-Objekte und erfasst diese Objekte bei der Generierung.
- Manuelle Änderungen an den generierten Objekten werden überschrieben.

Siehe auch

Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)

6.3.19 SiVArc-Regeln exportieren und importieren

Einleitung

SiVArc-Regeln und -Regelgruppen können nach MS-Excel exportiert und von dort aus importiert werden.

Export und Import sind pro SiVArc-Editor oder für das gesamte Projekt möglich.

Einzelne Regeln außerhalb von Gruppen können Sie auch direkt aus dem MS-Excel-Tabellenblatt in einen SiVArc-Editor kopieren und umgekehrt.

Hinweis

Exportieren und Kopieren von Regeln

Wenn Sie Regeln kopieren und einfügen, werden nur die sichtbaren Spalten eingefügt.

SiVArc-Regeln aus einem SiVArc-Editor exportieren

1. Öffnen Sie den gewünschten SiVArc-Editor.
2. Klicken Sie in der Symbolleiste auf die Schaltfläche "Export".
Ein Dialogfeld wird geöffnet.
3. Wählen Sie den gewünschten Ablageort und den Namen der Exportdatei.
4. Klicken Sie auf "OK".

Die Exportdatei wird angelegt.

SiVArc-Regeln eines Projekts exportieren

1. Wählen Sie in der Projektnavigation "Gemeinsame Daten > SiVArc".
2. Wählen Sie im Kontextmenü "Export aller Regeln".
Ein Dialogfeld wird geöffnet.
3. Wählen Sie den gewünschten Ablageort und den Namen der Exportdatei.
4. Klicken Sie auf "OK".

Die Exportdatei wird angelegt.

Aufbau der Exportdatei

In der Arbeitsmappe wird für jeden SiVArc-Editor ein Tabellenblatt mit den exportierten SiVArc-Regeln angelegt. Die Tabellenblätter haben folgende Bezeichnungen:

- ScreenRules
- AlarmRules
- TagRules
- TextlistRules
- CopyRules

Regeln zum Import

Beachten Sie beim Importieren der SiVArc-Regeln in einen einzelnen SiVArc-Editor Folgendes:

- Die Importdatei muss das Format "*.xlsx" haben.
- Wenn eine Importdatei nur ein Tabellenblatt hat, wird dieses unabhängig von seiner Bezeichnung importiert.
- Nur wenn Tabellenblätter einer Importdatei umbenannt oder gelöscht wurden, wählen Sie die gewünschten Tabellenblätter über einen Dialog aus.
 - Um ein umbenanntes Tabellenblatt zu importieren, bestätigen Sie den Import gesondert in einem Dialog.
 - Um ein Tabellenblatt vom Import auszuschließen, überspringen Sie das Tabellenblatt im Dialog. Wenn Sie es vor dem Import gelöscht haben, müssen Sie im Dialog trotzdem eine leere Anzeige überspringen.

Hinweis

Achten Sie beim Import darauf, dass die eingestellte Projektierungssprache Ihres Projekts und die in der Importdatei verwendete Sprache übereinstimmen.

Importoptionen

Folgende Optionen sind beim Import von SiVArc-Regeln verfügbar:

- Bestehende Regeln durch Importieren überschreiben.
Die Regel/Regelgruppe im Regeleditor wird an die Hierarchie der Regeln/Regelgruppen der importierten Datei angepasst.
- Zu importierende Regeln umbenennen, falls Regelname schon vergeben.
Bei Namenskonflikten werden die importierten Regeln und Regelgruppen mit einem Unterstrich angehängt. Beispiel: "Rule" wird nach dem Umbenennen zu "Rule_1".
- Alle bestehenden Regeln vor dem Import löschen.
Nach dem Import werden die alten Regeln vom Regeleditor gelöscht und die Regeln/Regelgruppen der importierten Datei werden angezeigt.

Regelgruppen importieren

Wenn eine Regelgruppe nicht eindeutig zugeordnet werden kann, wird sie in der ersten Hierarchieebene des Editors eingefügt, z. B. wenn in der Importdatei ein Zirkelbezug vorliegt oder darin die übergeordnete Gruppe fehlt.

Wenn bereits enthaltene Regeln beim Import nicht umbenannt werden, wird eine Regelgruppe, die mehrfach in der Importdatei enthalten ist, jeweils von der in der Importdatei am weitesten unten aufgeführten Regelgruppe überschrieben.

SiVArc-Regeln in einen SiVArc-Editor importieren

1. Öffnen Sie den gewünschten SiVArc-Editor.
2. Klicken Sie in der Symbolleiste des Editors auf die Schaltfläche "Import".
Ein Dialogfeld wird geöffnet.
3. Wählen Sie die gewünschte Importdatei und -option.
Wenn die Importdatei mehrere Tabellenblätter enthält, wird ein Dialog geöffnet.
4. Wählen Sie das gewünschte Tabellenblatt.
5. Klicken Sie auf "OK".

SiVArc-Regeln in ein Projekt importieren

1. Wählen Sie in der Projektnavigation "Gemeinsame Daten > SiVArc".
2. Wählen Sie im Kontextmenü "Import aller Regeln".
Ein Dialogfeld wird geöffnet.
3. Wählen Sie die gewünschte Importdatei und -option.
4. Klicken Sie auf "OK".

Ergebnis

Die SiVArc-Regeln werden in den SiVArc-Editoren angelegt. Die Fertigstellungsmeldung enthält einen Link zur Protokolldatei. Alternativ finden Sie das Importprotokoll unter "Gemeinsame Daten > Protokolle".

6.3.20 Know-how-Schutz für ein SiVArc-Projekt einrichten

Einleitung

Ihr SiVArc-Projekt enthält individuell mit der SiVArc-Scripting-Funktionalität erstellte SiVArc-Generiervorgaben. Um SiVArc-Ausdrücke im gesamten Projekt zu schützen, aktivieren Sie für Ihr Projekt den Know-how-Schutz.

Der Know-how-Schutz erstreckt sich nur auf die SiVArc-Editoren, aber nicht auf die Einstellungen von SiVArc. Die Bibliothek und die SiVArc-Registerkarten im Inspektorfenstern bleiben unberührt, ebenso generierte Objekte.

Passwort

Für den Know-how-Schutz vergeben Sie ein Passwort. Das Passwort muss mindestens 8 Zeichen sein und folgenden Zeichentypen enthalten:

- Groß- und Kleinbuchstaben
- Sonderzeichen
- Zahlen

Know-how-Schutz einrichten

1. Selektieren Sie in der Projektnavigation "Gemeinsame Daten > SiVArc".
2. Wählen Sie im Kontextmenü "Know-how-Schutz > Aktivieren".
Ein Dialog wird geöffnet.
3. Legen Sie das Passwort fest.
4. Speichern Sie Ihr Projekt.

Über das Kontextmenü bearbeiten Sie auch Ihr Passwort und heben den Know-how-Schutz wieder auf.

Ergebnis

Der Know-how-Schutz ist für alle SiVArc-Editoren aktiviert. Wenn Sie einen SiVArc-Editor in der Projektnavigation, in STEP 7 oder über einen Einsprung aus den anderen Editoren öffnen wollen, wird erst ein Passwort abgefragt. Zugleich ist der Know-how-Schutz für den Import und Export von SiVArc-Regeln aktiviert.

6.4 Visualisierung generieren

6.4.1 Grundlagen zur Generierung

Definition

Beim Generieren der Visualisierung generieren Sie HMI-Objekte entsprechend den Texten und Strukturen des Anwenderprogramms. Zusätzlich kopieren Sie beim Generieren auch HMI-Objekte ohne Bezug zum Anwenderprogramm.

Im Unterschied zum Kopieren von HMI-Objekten, die auf Bibliothekselementen basieren, werden die Ergebnisse einer Generierung mit SiVArc durch nachfolgende Generierungen erneut erfasst und angepasst.

Phasen der Generierung

Mit SiVArc wird die Visualisierung in mehreren Phasen generiert. Auf diese Weise wird das Projekt von einer Generierung zur nächsten verbessert und ergänzt. Die vorhandenen Generaten werden dabei ständig bereinigt und aktualisiert.

Bei SiVArc ist zwischen der ersten und nachfolgenden Generierungen zu unterscheiden. Die nachfolgenden Generierungen basieren auf der ersten Generierung. Manuelle Änderungen, beispielsweise Neuplatzierungen der generierten Anzeige- und Bedienobjekte, werden für nachfolgende Generierungen beibehalten.

Sie können eine vollständig neue Generierung auslösen oder die Auswahl der für die nächsten Generierungen zu erzeugenden Geräte ändern.

Geräteübergreifendes Generieren der Visualisierung

Wenn in Projekten mit vielen PLCs und mehreren Bedienstationen einzelne Änderungen erforderlich sind, ist es vorteilhaft, die Visualisierung nur für einzelne Geräte vorzunehmen, beispielsweise wenn Geräte auszutauschen oder Fehler zu beheben sind. Die Generier- und Ladezeiten verringern sich entsprechend. HMI-Objekte können mithilfe einer Bildregel oder einer Bildregelgruppe für mehrere Bediengeräte generiert werden.

Auf diese Weise aktualisieren und optimieren Sie in einem großen SiVArC-Projekt die Prozessbilder der Anlage sowohl für alle Geräte oder Gerätetypen zusammen als auch jeweils einzeln. Folgende Funktionen helfen Ihnen dabei:

- Aus- und Einblenden gerätespezifischer Spalten in den SiVArC-Editoren über die Symbolleiste
- Verteilen von einzelnen Regeln auf verbundene Geräte und Steuerungen
- Anzeige der Gerätetypen in den Bildregeln zur einfacheren Zuordnung der passenden Positionierungsschemata
- PLC-abhängige Anzeige der Gerätetypen im Inspektorfenster der Bildregeln

Wenn Bildregeln, die Kopiervorlagen für Unified-Geräte verwenden, aus Unified- und HMI-Geräten bestehen, werden die Bilder bei der SiVArC-Generierung nur auf Unified-Geräten generiert. Für Unified konfigurierte Bilder werden nur auf Unified-Geräten generiert.

Fehlende Zuordnung des Bediengeräts wiederherstellen

Wenn Sie innerhalb eines Projekts ein Gerät kopieren, einfügen oder umbenennen, behält das Gerät den Runtime-Gerätenamen bei. Der textliche Querverweis auf diesen Runtime-Namen kann nach der Umbenennung nicht entfernt werden.

1. Den Eintrag für das betreffende Bediengerät können Sie unter "Geräte und Netze > Netzsicht > Netzübersicht > Gerät" erweitern.
2. Passen Sie den Runtime-Gerätenamen entsprechend an.

Ihre Zuordnung wird nach der nachfolgenden Generierung wirksam.

Umfang der Generierung

SiVArC bietet Ihnen mehrere Optionen, um den Umfang der Generierung festzulegen.

- Variablen-Generierungsmodus unter "Optionen > Einstellungen > SiVArC"
Falls erforderlich, beschränken Sie die Generierung von Variablen auf die im Gebrauch befindlichen Variablen.
- Regeln in den Regeleditoren von der Generierung ausnehmen
- Regeln im Generierungsdialog von der Generierung ausnehmen

Wenn Sie keine SiVArC-Regeln erstellen, generiert SiVArC nur externe Variablen.

Erste geräte-abhängige Generierung

Wenn Ihr Projekt mehrere Bediengeräte oder verbundene PLCs umfasst, generiert SiVArC die Visualisierung für die von Ihnen ausgewählten Bediengeräte und PLCs.

Beim Start der ersten Generierung in einem Projekt wird ein Dialogfeld zur Auswahl der Bedienstation angezeigt. In diesem Dialogfeld wählen Sie die Geräte aus, für die SiVArC die Visualisierung generieren soll.

SiVArc generiert die Visualisierung Gerät für Gerät.

- Ist die Generierung für ein Gerät nicht möglich, fährt SiVArc mit dem nächsten Gerät fort.
- Wenn Sie die Generierung abbrechen, bleibt die letzte zuvor vollständig generierte Visualisierung für ein Gerät erhalten.

Die verfügbare Auswahl der Bedienstationen wird nach der ersten Generierung eingefroren. Alle nachfolgenden Generierungen basieren auf dieser Auswahl.

Namensgleichheit

Wenn die Namen manuell erstellter HMI-Objekte identisch sind, werden die manuell erstellten HMI-Objekte umbenannt. Die von SiVArc neu generierten Objekte werden immer mit dem zu generierenden Namen erstellt.

Wenn ein manuell erstelltes HMI-Objekt mit einem von SiVArc zu generierenden Namen bereits vorhanden ist, erhält das vorhandene Objekt das Suffix "_renamed". Sollte dieser Name ebenfalls bereits belegt sein, wird er automatisch inkrementiert.

Kommt es bei der Generierung mit mehreren verbundenen PLCs zu Namenskonflikten, generiert SiVArc nur das von der Generierung erfasste HMI-Objekt und gibt eine Fehlermeldung aus.

Hinweis

Laufzeiteinstellungen von Bediengeräten

Bei der Generierung von Variablen mit mehreren PLCs wird die Option "PLC-Präfix" aus den Laufzeiteinstellungen des Bediengeräts ausgewertet.

Achten Sie darauf, dass die Option "PLC-Präfix" in den Laufzeiteinstellungen der einzelnen PLCs aktiviert ist. Andernfalls wird die SiVArc-Generierung abgebrochen.

Hinweis

Ausnahme

In Bezug auf Bilder und Textlisten weicht das Verhalten von SiVArc bei Namensgleichheit wie folgt ab:

Wenn generierte Bilder oder Textlisten mit gleichem Namen bereits vorhanden sind, generiert SiVArc diese Bilder oder Textlisten trotz des Namenskonflikts erneut. Beachten Sie, dass eine Fehlermeldung für Textlisten, aber nicht für Bilder ausgegeben wird.

Ergebnis der ersten Generierung

SiVArc generiert die HMI (Unified)-Objekte anhand der SiVArc-Regeln und speichert sie gemäß der Projektierung. Für die generierten Objekte werden die Plug-in-Eigenschaften mit den konfigurierten Daten aktualisiert.

Erneute geräte-abhängige Generierung

Wenn Sie ein Gerät im Dialogfeld für die Stationsauswahl nicht erneut für die nächste Generierung aktivieren, bleiben die generierten Objekte und die manuellen Änderungen im Projekt erhalten.

Hinweis

Um generierte Objekte eines PLC zu entfernen, der bei der nächsten Generierung nicht mehr aktiviert ist, löschen Sie die Verbindung zwischen dem PLC und dem Bediengerät.

Wenn eine vorhandene Verbindung zwischen dem Bediengerät und der Steuerung gelöscht wurde, wird im Dialogfeld für die Stationsauswahl eine Warnmeldung ausgegeben. Beim Löschen einer Verbindung zwischen PLC und Bediengerät werden alle zugeordneten generierten Objekte bei der nächsten Generierung ebenfalls entfernt.

Generierung mit neuer Stationsauswahl

Der Dialog für die Stationsauswahl wird bei der ersten Generierung eines Projekts immer angezeigt.

Der Dialog für die Stationsauswahl wird erst wieder angezeigt, wenn die nächste Generierung startet. SiVArc generiert dann dieselben Bediengeräte und PLCs wie bei der vorherigen Generierung. Zum Ändern von Einstellungen gehen Sie wie folgt vor:

1. Wählen Sie das Projekt oder Gerät in der Projektnavigation.
2. Klicken Sie auf "Visualisierung (SiVArc)". Sie haben die Auswahl zwischen den folgenden Möglichkeiten:
 - Generieren der Visualisierung (SiVArc)
 - Generieren mit Stationsauswahl
 - Löschen der Visualisierung (SiVArc)

Alternativ drücken Sie die Tastenkombination <ALT+Umschalt+G>.

Wenn Sie Bildbausteine mit Multiinstanz-DBs projektieren, werden bei der Generierung die Warnmeldungen nicht angezeigt, wenn "Gemeinsame Daten > SiVArc Einstellungen > Einstellungen Warnungen > Warnungen für bereits bestehende Bildobjekte verbergen" aktiviert ist. Das Kontrollkästchen "Warnungen für bereits bestehende Bildobjekte verbergen" ist standardmäßig deaktiviert. Warnmeldungen stehen standardmäßig in Ordner "Gemeinsame Daten > Protokolle" zur Verfügung.

Änderungen an der Steuerung

Wenn Sie einen PLC löschen, für den Sie bereits eine Visualisierung generiert haben, werden alle mit diesem PLC generierten Objekte bei der nächsten Generierung gelöscht.

Wenn Sie einen Bausteinanruf im Anwenderprogramm löschen und erneut generieren, werden die für diesen Bausteinanruf generierten Objekte gelöscht.

Prüfung der PLC-Übersetzung unterdrücken

Sie können die Prüfung der PLC-Übersetzung mithilfe der Datei "SivarcDisableCompileClean" unterdrücken. In diesem Fall wird die Generierung auch dann durchgeführt, wenn bei der PLC-Übersetzung Fehler aufgetreten sind.

Hierzu gehen Sie wie folgt vor: Erstellen Sie im SiVArc-Installationsverzeichnis, das die Datei "Siemens.Simatic.Sivarc.dll" enthält, eine leere Datei mit dem Namen "SivarcDisableCompileClean".

Hinweis

Wenn sich im SiVArc-Installationsverzeichnis keine Datei mit dem Namen "SivarcDisableCompileClean" befindet und die PLC-Übersetzung Fehler aufweist, wird die SiVArc-Generierung abgebrochen.

Wenn sich im SiVArc-Installationsverzeichnis tatsächlich eine Datei mit dem Namen "SivarcDisableCompileClean" befindet und die PLC-Übersetzung fehlerfrei ist, wird die SiVArc-Generierung ausgeführt.

Siehe auch

Visualisierung generieren (Seite 215)

Vermeiden von Konflikten beim Generieren (Seite 183)

Nachträgliche Änderungen (Seite 209)

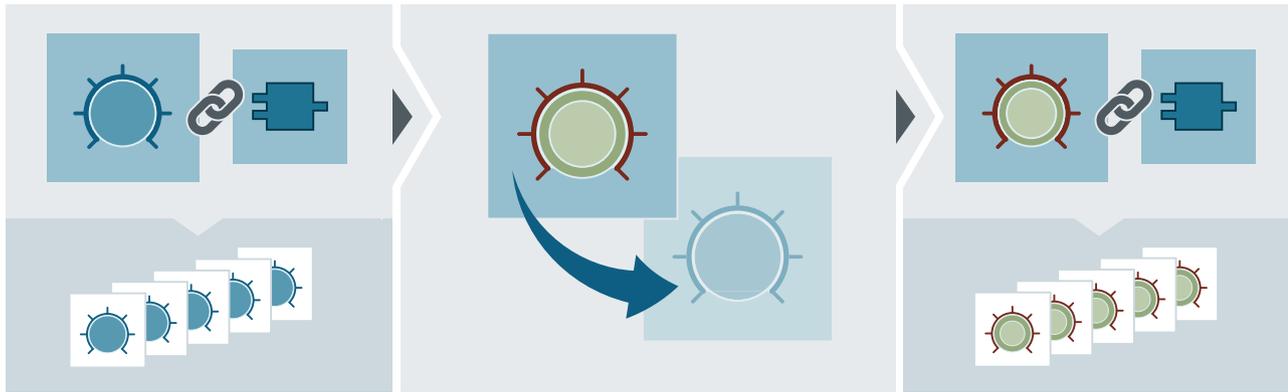
6.4.2 Nachträgliche Änderungen

6.4.2.1 Ändern von generierten Objekten

Anwendungsbezug

Generierte Anzeige- und Bedienobjekte ändern Sie mit SiVArc zentral über die Generiervorlagen oder im Anwenderprogramm. Manuelle Änderungen an den generierten Objekten gehen bei der nächsten Generierung verloren.

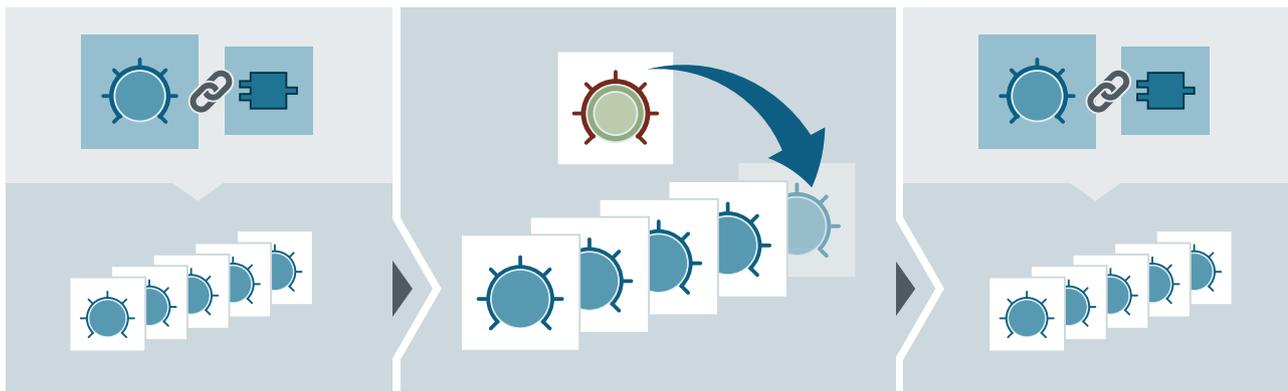
Wenn in einem Projekt nachträglich die Anzeige- und Bedienobjekte grafisch angepasst werden sollen, ändert der Visualisierungsingenieur lediglich zentral die dazugehörige Generiervorlage. Die Vorlage legt der Projektteur unter gleichem Namen wie die vorherige in der Bibliothek ab. Bei der nächsten Generierung werden die Anzeige- und Bedienobjekte ohne jede weitere Projektierung grafisch angepasst.



SiVArC-Bezug generierter Objekte

Generierte HMI-Objekte haben einen bleibenden Bezug zu den SiVArC-Regeln, aus denen sie abgeleitet wurden. Dieser Bezug führt bei jeder Neugenerierung zu folgenden Ergebnissen:

- Objekte, die keinen Bezug mehr zur SiVArC-Projektierung haben, werden entfernt. Der Bezug geht verloren, wenn z. B. eine Regel gelöscht wird.
- Objekte, deren Vorgaben zum Generieren verändert wurden, werden aktualisiert.
- Außer SiVArC-konforme manuellen Änderungen werden alle manuellen Veränderungen an generierten Objekten rückgängig gemacht.



SiVArC-konforme manuelle Änderungen

Folgende Änderungen an generierten Anzeige- und Bedienobjekten bleiben für alle weiteren Generierungen bestehen

- Die erste Neupositionierung von generierten Objekten
Die Neupositionierung bleibt auch erhalten, wenn die Position über ein eigenes Positionierungsschema definiert wurde. Auch wenn Sie das eigene Positionierungsschema ändern, bleibt die manuell eingerichtete Position nach einer weiteren Generierung erhalten. Nur bei Bildobjekten mit fester Positionierung wird eine manuelle Änderung der Position bei der nächsten Generierung wieder auf die hinterlegte feste Positionierung zurückgesetzt.
- Änderungen an Größe und Rotationswinkel bleiben außer bei Bildbausteinen und Bildfenstern bestehen.

- In einem generierten Bildfenster können Sie das angezeigte Bild ändern.
- Manuell geänderte Textlisteneinträge bleiben nach einer nachfolgenden Generierung erhalten.

Generiermatrix

Die Änderungen über die Generiermatrix bleiben bei jeder weiteren Generierung erhalten:

- Über die Generiermatrix generieren Sie Objekte auf andere Bilder.
- Über die Generiermatrix generieren Sie Bilder auf andere Geräte.

Manuell erzeugte HMI-Objekte

Manuell erzeugte HMI-Objekte bleiben, außer bei Namenskonflikten, von der SiVArc-Generierung unberührt.

Wenn Sie generierte Anzeige- und Bedienobjekte per Kopieren und Einfügen in Ihrem Projekt wiederverwenden, gelten diese als manuell angelegte Objekte und verlieren den SiVArc-Bezug. Wie alle anderen manuell angelegten Bildobjekte werden die kopierten Objekte von SiVArc niemals gelöscht.

Hinweis

Generierte Anzeige- und Bedienobjekte aus der Generiervorlage eines Bildes

Wenn Sie Bildobjekte per Copy&Paste wiederverwenden wollen, verwenden Sie nur Bildobjekte außerhalb eines Generats aus einer Kopiervorlage eines Bildes.

Namensänderungen generierter Anzeige- und Bedienobjekte

Wenn der Name eines generierten HMI-Objekts geändert wurde, wird das Objekt bei der nächsten SiVArc-Generierung erneut erzeugt und verschaltet. Zusätzlich ist im Projekt das Objekt mit dem geänderten Namen enthalten.

Ändern Sie Namen von generierten Anzeige- und Bedienobjekten ausschließlich über den SiVArc-Ausdruck in der Generiervorlage oder in den Textquellen im Anwenderprogramm. Bei der nächsten Generierung wird der Name des generierten Objekts entsprechend aktualisiert.

Manuell überschriebene Textlisteneinträge

Wenn Sie generierte Textlisteneinträge überschreiben, bleibt bei der nächsten Generierung der geänderte Textlisteneintrag nur für den Standardtext der Kopiervorlage erhalten.

Wenn der Text für die Textliste aus der Netzwerktextdefinition in STEP 7 oder den Symboltabellen generiert wird und Sie diesen Text ändern, werden Ihre Änderungen in der nachfolgenden Generierung überschrieben.

Das folgende Beispiel verdeutlicht, wie SiVArc geänderte Textlisteneinträge verarbeitet:

Die Textliste enthält zwei Einträge: "Eintrag_1" und "Eintrag_2". "Eintrag_1" enthält einen Text, der von SiVArc generiert wurde. "Eintrag_2" enthält einen Text, der aus der Kopiervorlage der Textliste kopiert wurde.

- Sie ändern "Eintrag_2" und starten die SiVArc-Generierung. Nach der Generierung sind Ihre Änderungen am "Eintrag_2" vorhanden.
- Sie ändern "Eintrag_1" und starten die SiVArc-Generierung. Nach der Generierung sind Ihre Änderungen am "Eintrag_1" durch einen von SiVArc generierten Text überschrieben.
- Sie ändern "Eintrag_1" und "Eintrag_2" und starten die SiVArc-Generierung. Nach der Generierung sind Ihre Änderungen am "Eintrag_1" durch einen von SiVArc generierten Text überschrieben. Ihre Änderungen am "Eintrag_2" sind durch den Text aus der Kopiervorlage der Textliste überschrieben.

Vorteile der SiVArc-Änderungsmechanismen

Die SiVArc-Funktionalität zum Ändern von generierten Anzeige- und Bedienobjekten ermöglicht dem Projektteur eine effiziente und konsistente Anpassung mit wenig Aufwand.

Dem Unternehmen bietet SiVArc damit die Möglichkeit, standardisierte Anzeige- und Bedienobjekte firmenweit zu verteilen und auch in laufenden Projekten anzuwenden.

Siehe auch

Beispiel: Generiermatrix einsetzen (Seite 224)

Kennzeichnungen im SiVArc-Projekt (Seite 214)

Bildlegende (Seite 279)

6.4.2.2 Ändern von SiVArc-Regeln

Wirkungen

Mit jeder Änderung von SiVArc-Regeln wird zentral in ein bestehendes Projekt eingegriffen.

SiVArc-Regeln müssen beispielsweise bearbeitet werden, wenn eine Funktion immer in einem anderen Bild visualisiert werden soll, oder wenn ein aktives Objekt entfernt werden soll, weil sich die Anlage geändert hat.

SiVArc-Regeln später bearbeiten

Sie können bereits erstellte Regeln ändern – wählen Sie dazu die Regel aus und verwenden Sie die Befehle im Kontextmenü. Wenn Sie den Namen und die Speicherpfade von Objekten im Projekt ändern, werden die betroffenen Regeln entsprechend aktualisiert.

Ändern Sie den Namen und die Speicherpfade von Objekten nur im Projekt oder in der Projektbibliothek. Änderungen in globalen Bibliotheken oder in den Pfadinformationen für referenzierte Objekte werden von SiVArc nicht unterstützt.

Wenn eine SiVArc-Bildregel und -Kopierregel das gleiche Bibliotheksobjekt enthalten, geht bei der SiVArc -Generierung die Kopierregel vor und erzeugt das Bibliotheksobjekt zusammen mit der in der Kopierregel festgelegten Ordnerstruktur.

Wenn eine SiVArc-Variablenregel und -Kopierregel das gleiche Bibliotheksobjekt (Variablen-tabelle) enthalten, geht bei der SiVArc-Generierung die Kopierregel vor und erzeugt das Bibliotheksobjekt zusammen mit der in der Kopierregel festgelegten Ordnerstruktur. Die Variablen-tabelle mit PLC-Variablen wird gemäß Festlegung in der Kopierregel generiert. Wenn eine Variablenregel für die Generierung verwendet wird, erscheint eine Warnmeldung.

Steht keine Kopierregel zur Verfügung und Sie wählen eine SiVArc-Bildregel, die zuerst generiert werden soll, wird ein Bildobjekt generiert. Beim Hinzufügen einer Kopierregel für die zweite Generierung wird eine Warnmeldung angezeigt.

SiVArc-Regeln in einer Bibliothek verwenden

Um SiVArc-Regeln zentral und projektübergreifend einheitlich aktualisieren zu können, legen Sie die SiVArc-Regeln oder -Regelgruppen als Kopiervorlage in einer Bibliothek ab. Wenn im Projekt bereits eine SiVArc-Regel mit demselben Namen vorhanden ist, können Sie die Regel überschreiben oder eine neue Regel erstellen.

Wenn eine Regel mit einer Regel aus der Bibliothek überschrieben wird, reagiert SiVArc so, als wäre die Regel manuell geändert worden:

- SiVArc erkennt die relevanten HMI-Objekte aus einem früheren Generiervorgang und schließt diese Objekte in die Generierung ein.
- Manuelle Änderungen an den relevanten HMI-Objekten werden überschrieben.

Namen von Kopiervorlagen für SiVArc-Regeln ändern

Um eine Verknüpfung zwischen einer umbenannten Bildregel in der Bibliothek und der darauf basierenden Bildregel im Projekt herzustellen, gehen Sie wie folgt vor:

1. Ändern Sie die Bildregeln im Projekt manuell anhand der neuen Namen der Kopiervorlagen in der Bibliothek.
2. Kopieren Sie dann die umbenannten Kopiervorlagen in Ihr Projekt. Überschreiben Sie die vorhandenen und neu benannten Bildregeln im Projekt.

Referenzen einer SiVArc-Regel bearbeiten

Wenn Sie referenzierte HMI-Objekte oder Programmteile im Projekt oder in der Projektbibliothek bearbeiten, wird die SiVArc-Regel automatisch angepasst.

Wenn Sie referenzierte Objekte in der globalen Bibliothek ändern, werden die entsprechenden SiVArc-Regeln ungültig.

Vorteile der Bearbeitung von SiVArc-Regeln

Da SiVArc-Regeln aus dynamischen Verknüpfungen im Projekt bestehen, können sie einfach angepasst werden, ohne dadurch Inkonsistenzen im Projekt zu verursachen. Sie können zum Beispiel mit festen Regelsätzen arbeiten, die Sie individuell anpassen oder im Unternehmen weiterentwickeln.

Siehe auch

- SiVArc-Regeln (Seite 161)
- Variablenregel anlegen (Seite 186)
- Bildregel anlegen (Seite 188)
- Textlistenregel anlegen (Seite 189)
- Kopierregel anlegen (Seite 192)

6.4.2.3 Kennzeichnungen im SiVArc-Projekt**Generierte Objekte und SiVArc-Projektierungen im Projekt**

Folgende Objekte sind in einem SiVArc-Projekt gekennzeichnet:

- Generierte Objekte, die von der nächsten Generierung erfasst werden
- Objekte, die SiVArc-Projektierungen enthalten

Die folgende Tabelle zeigt, in welcher Form Objekte in SiVArc gekennzeichnet sind:

Ort	Icon/ Markierung	Objekt
Projektnavigation		Relevantes Objekt (HMI-Bild)
Projektbibliothek oder Globale Bibliothek		Kopiervorlage mit projektierten SiVArc-Eigenschaften, -Ereignissen oder -Animationen
		Typ mit projektierten SiVArc-Eigenschaften, -Ereignissen oder -Animationen
		Typ-Version mit projektierten SiVArc-Eigenschaften, -Ereignissen oder -Animationen

Die Kennzeichnung für generierte Bildobjekte im Editor "Bilder" legen Sie unter "Extras > Einstellungen > SiVArc" fest.

Generiertes HMI-Objekt

Ein generiertes HMI-Objekt wird bei der nächsten Generierung neu generiert und überschrieben. Von der nächsten Generierung werden die Objekte erfasst, die den gleichen Namen wie in der letzten Generierung haben.

Wenn Sie den Namen eines generierten Objekts ändern, wird es von der Generierung nicht mehr erfasst.

Hinweis**Generierte Objekte in andere Projekte kopieren**

Wenn Sie ein generiertes Objekt in andere Projekte mit oder ohne SiVArc-Installation kopieren, bleibt die Kennzeichnung erhalten.

Kennzeichnung im Editor "Bilder"

Die Kennzeichnung im Editor "Bilder" ist optional. Sie aktivieren die Kennzeichnungen und legen die gewünschten Farben für Rahmen und Hintergrund in den Einstellungen des TIA-Portals unter "Extras > Einstellungen > SiVArc" fest.

6.4.3 Visualisierung generieren

Voraussetzungen

- Anwenderprogramm und Hardware wurden fehlerfrei übersetzt.
- Bildregeln wurden definiert.
- Die in den Bildregeln verwendeten Masterkopien und Bildbausteintypen sind in der Projektbibliothek oder in der globalen Bibliothek gespeichert.
- Variablen wurden definiert.
- Textlistenregeln sind definiert.
- Alle verwendeten Instanzen von Typen sind auf die neueste Version aktualisiert.

Hinweis

Änderungen an der Steuerung benötigen Übersetzung

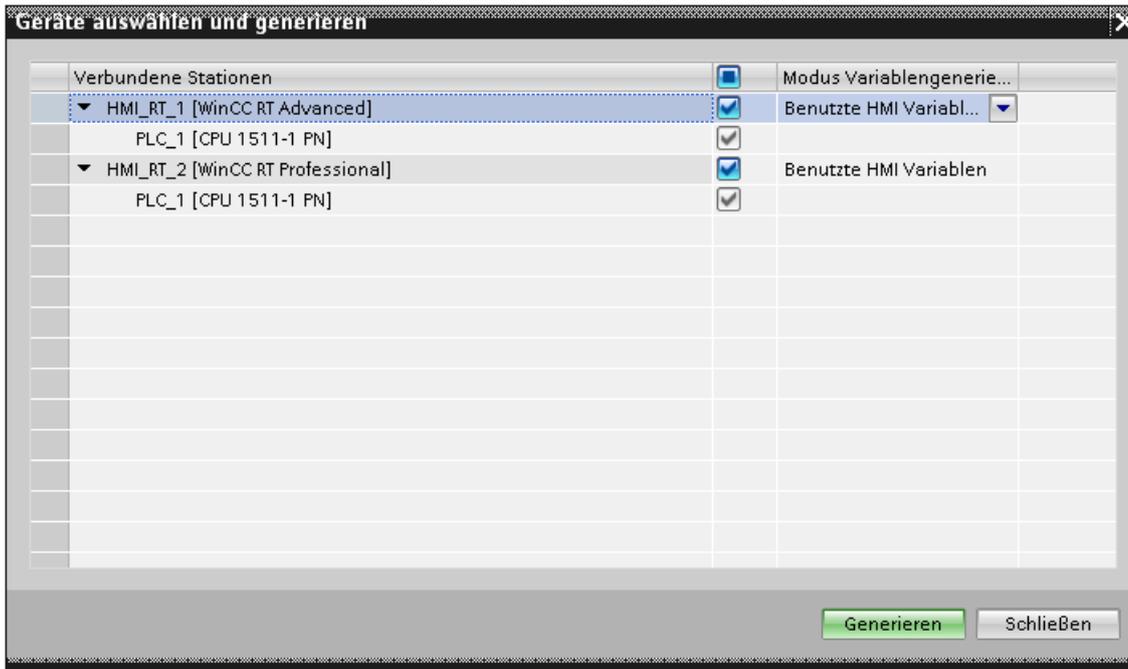
Änderungen am Anwenderprogramm oder an der Hardwarekonfiguration müssen übersetzt werden, bevor Sie die Visualisierung generieren.

Generieren ohne Stationsauswahl

1. Klicken Sie im Kontextmenü der Runtime oder des HMI-Geräts in der Projektnavigation auf "Visualisierung (SiVArc) >". Wählen Sie eine der folgenden Optionen:
 - Generieren der Visualisierung (SiVArc)
 - Generieren mit Stationsauswahl
 - Löschen der Visualisierung (SiVArc)

Generieren mit Stationsauswahl

1. Achten Sie darauf, dass die Option "PLC-Präfix" in den Laufzeiteinstellungen des HMI-Geräts für alle PLCs aktiviert ist.
2. Klicken Sie im Kontextmenü des Projekts in der Projektnavigation auf "Generieren der Visualisierung (SiVArc)> Generieren mit Stationsauswahl".
Der Dialog "Geräte auswählen und generieren" wird geöffnet.



3. Aktivieren Sie die HMI-Geräte und PLCs, für die eine Visualisierung generiert werden soll.
Um die Visualisierung für alle Geräte zu generieren, aktivieren Sie die Option in der Kopfzeile.
4. Klicken Sie auf "Generieren".

Löschen der Generierungsdaten

Sie haben die Möglichkeit, die Generierungsdaten wie folgt zu löschen:

1. Klicken Sie mit der rechten Maustaste auf ein **"HMI-Gerät>Visualisierung (SiVArc)>Löschen der Visualisierung (SiVArc)"**.
2. Wählen Sie den Regelsatz aus dem Stationsauswahl-Popup für ein ausgewähltes HMI-Gerät:
 - Vom Anwender erstellte Regeln – Das System löscht alle Vorkommnisse von mittels SiVArc generierten Objekten, die nur nach anwenderseitig erstellten Regeln generiert wurden.
 - Energy Suite-Regeln – Das System löscht SiVArc-Objekte, die nur mit Energy Suite-Regeln generiert wurden.
 - Alle Regeln – Das System löscht alle Vorkommnisse von SiVArc-Objekten, die wie oben erwähnt generiert wurden.

Hinweis

- HMI-Variablen für ein gewähltes HMI-Gerät wurden nach anwenderseitig erstellten Regeln oder nach Energy Suite-Regeln generiert. Wenn Sie die Aufgabe **"Löschen der Visualisierung (SiVArc)"** ausführen, werden alle HMI-Variablen unabhängig von der Art der Generierung gelöscht.
 - Beim Bereinigen der Generierungsdaten werden die SiVArc-Lizenz und Energy Suite-Lizenzen validiert.
-

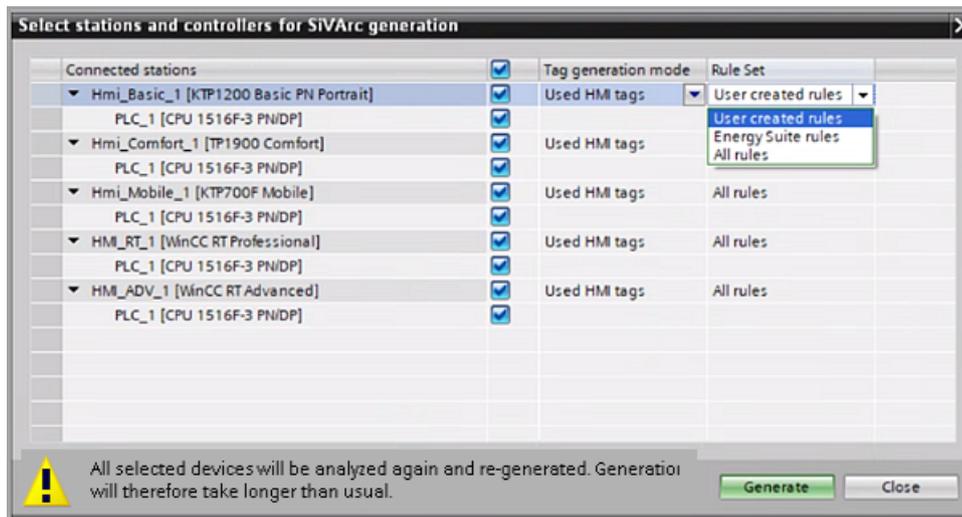
3. Das Dialogfeld für die Bereinigung von SiVArc informiert Sie mit einer Alarmmeldung darüber, dass die Daten aus "Generierübersicht" und "Generiermatrix" gelöscht werden.
4. Während der Bereinigung der Generierungsdaten können Sie den Vorgang jederzeit abbrechen, indem Sie im Dialogfeld für den Fortschritt der Bereinigung auf die Schaltfläche "Abbrechen" klicken.
5. Beim Klicken auf die Schaltfläche "Abbrechen" werden keine Daten gelöscht und die Transaktion wird rückgängig gemacht bzw. abgebrochen.

Neustart der Gesamtgenerierung

Wenn Sie Änderungen am Anwenderprogramm vorgenommen haben, kann es notwendig sein, eine Gesamtgenerierung neu zu starten. Selbst wenn im Anwenderprogramm nichts geändert wurde, durchläuft die Gesamtgenerierung das gesamte Anwenderprogramm.

Die Auswahl der verbundenen Stationen bleibt die gleiche wie in der ersten Generierung und kann nicht geändert werden.

Sie starten die Gesamtgenerierung durch Auswahl des Projekts oder des Geräts und drücken dann <Alt>+<Umschalt>+<F>.



Geräteanzeige

Geräte, die im Projekt vorhanden, aber nicht an eine Steuerung angeschlossen sind, werden in den SiVArC-Editoren nicht angezeigt.

Hinweis

IPI-Geräte

Steuerungen und Geräte, die über IPI mit dem Projekt verbunden sind, werden im Auswahlfenster nicht angezeigt.

6.5 Ergebnis prüfen

6.5.1 Prüfen des Ergebnisses

Einleitung

Umfangreiche SiVArC-Projekte erfordern eine zusätzliche Analyse und Optimierung nach der erstmaligen Generierung.

Dieses Dokument bietet eine Übersicht der Optionen für Analyse und Nachbearbeitung eines SiVArC-Projekts.

Mechanismen prüfen

SiVArc stellt verschiedene Funktionen und Editoren für die Prüfung der Generierung bereit. Darin sind die folgenden Punkte eingeschlossen:

- Welche Objekte wurden tatsächlich generiert?
Eine Übersicht finden Sie im Editor "Generierübersicht" unter "Gemeinsame Daten > SiVArc". Die Generierübersicht wird bei der ersten Generierung erstellt und bei den weiteren Generierungen jeweils aktualisiert.
- Welche Objekte wurden nicht oder mit Fehlern generiert?
Während der Generierung aufgetretene Fehler finden Sie im Inspektorfenster unter "Info" mit direktem Link zum Fehlerort.
Außerdem wird ein Protokoll der Generierung in der Projektnavigation unter "Gemeinsame Daten > Protokolle" angezeigt.
Bei der Fehlerbehandlung ist die Organisation der Regeln in den Editoren hilfreich. Auf diese Weise können Sie Ihre Projekte in Abschnitten generieren, sodass mögliche Fehler einfacher zu finden sind.

Um ein Projekt zu erhalten, das ausgeführt werden kann, übersetzen Sie das Projekt und laden es, um es zu testen. Beim Testen wird sichergestellt, dass das Projekt vollständig ist und fehlerfrei in der Runtime läuft.

Überprüfung der Vollständigkeit bei der Generierung

Um die Generierübersicht zu öffnen, doppelklicken Sie auf "Gemeinsame Daten > SiVArc > Generierübersicht" in der Projektnavigation. Sie können die Generierübersicht auch aus der Fertigstellungsmeldung heraus öffnen, um die Visualisierung im Inspektorfenster zu generieren.

Um Bausteine, Bildregeln oder generierte Anzeige- und Bedienobjekte in dem in der Generierübersicht aufgeführten Projekt zu sehen, wählen Sie den Kontextmenübefehl "Gehe zu referenziertem Objekt".

Mit den Filter- und Sortierfunktionen des Editors können verschiedene Ansichten der Generierübersicht angezeigt werden. Auf diese Weise können die Beziehungen zwischen Bildregeln, generierten Anzeige- und Bedienobjekten und Geräten angesehen werden.

Mithilfe der Generierübersicht planen und projektieren Sie nachträgliche Änderungen für eine weitere Generierung.

Die Generierübersicht steht auch an mehreren Stellen im SiVArc-Projekt zur Verfügung:

Screen	Screen object	Master copy / type	HMI device	PLC device	Program block	Screen rule	Generated by ma...
1	Plantsection1	Plantscreen	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	<input type="checkbox"/>
2	Plantsection1	Plantsection_Title	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	<input type="checkbox"/>
3	Plantsection1	PlantStatus_Symb_IO	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_St...	<input type="checkbox"/>
4	Plantsection1	Activate	HMI_RT_1	PLC_1	Activate, Activate_DB	Activate_Btn	<input type="checkbox"/>
5	Plantsection1	Productionline_title	HMI_RT_1	PLC_1	Productionline, #Produc...	Productionline_...	<input type="checkbox"/>
6	Plantsection1	Productionlin...	HMI_RT_1	PLC_1	Productionline, #Produc...	Productionline_...	<input type="checkbox"/>
7	Plantsection1	Conveyor	HMI_RT_1	PLC_1	Conveyor	Conveyor	<input type="checkbox"/>
8	Plantsection1	ProcessingUnit	HMI_RT_1	PLC_1	Processing	Processing_Unit	<input type="checkbox"/>
9	Plantsection1	Conveyor	HMI_RT_1	PLC_1	Conveyor	Conveyor	<input type="checkbox"/>
10	Plantsection1	ProcessingUnit	HMI_RT_1	PLC_1	Processing	Processing_Unit	<input type="checkbox"/>
11	Plantsection1	Stop	HMI_RT_1	PLC_1	Stop, Stop_DB	Stop_Btn	<input type="checkbox"/>
12	Plantsection2	Plantscreen	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	<input type="checkbox"/>
13	Plantsection3	Plantscreen	HMI_RT_1	PLC_1	Plantsection, Plantsecti...	Plantsection_Tit...	<input type="checkbox"/>

- WinCC
Inspektorfenster eines generierten Bilds
Alle generierten Anzeige- und Bildobjekte des ausgewählten Bilds werden im Register "Generierübersicht" angezeigt.
- STEP 7
Inspektorfenster eines Bausteins
In der Generierübersicht "Bilder" und der Generierübersicht "Textlisten" werden alle aus dem ausgewählten Programmbaustein generierten Bilder mit den zugehörigen Bildobjekten und Textlisten angezeigt.

Filter- und Sortierfunktionen im Editor "Generierübersicht" erleichtern das Erzeugen einer Übersicht nach unterschiedlichen Kriterien.

Fehlerbehandlung mit gezielter Einzelgenerierung

Sie können Ihr Projekt auch abschnittsweise überprüfen. Mit den Regeleditoren können Sie einzelne Projektabschnitte generieren.

- Sie können Regeln als Gruppe ein- und ausschalten.
- Wird eine Regel nach der Generierung deaktiviert, werden alle zugehörigen generierten Objekte aus der Generierung entfernt.
- Das Aktivieren und Deaktivieren von Regeln überschreibt die Bedingung einer Regel. Hat eine Regel zum Beispiel die Bedingung "WAHR", wird sie nur angewandt, wenn die Regel aktiviert ist. Hat eine Regel die Bedingung "FALSCH", wird sie nicht in die Generierung eingeschlossen, auch wenn die Regel aktiviert ist.
- Werden die Regeln für die nächste Generierung erneut aktiviert, werden die zugehörigen Objekte ebenfalls erneut generiert.

Änderungen in letzter Minute

Verwenden Sie die Generiermatrix, um letzte Änderungen zu übernehmen, ohne die SiVArc-Regeln analysieren und ändern zu müssen.

Siehe auch

- Einsatz der Generiermatrix (Seite 221)
- Beispiel: Generiermatrix einsetzen (Seite 224)
- SiVArc-Regeln bearbeiten und verwalten (Seite 199)
- Anzeige in den SiVArc-Editoren bearbeiten (Seite 277)

6.5.2 Einsatz der Generiermatrix

Einsatz der Generiermatrix

Die Generiermatrix ist für nachträgliche Änderungen an der Zuordnung generierter Objekte geeignet und richtet sich vor allem an Inbetriebnehmer, die kurzfristig Anpassungen im Projekt vornehmen müssen.

Den größten Nutzen aus der Generiermatrix erzielen Sie, wenn Sie während der Projektierung nach Möglichkeit ausschließlich Bildregeln zum Generieren von Bildern und Bildobjekten verwenden.

Beschreibung

Im Editor "Generiermatrix" werden nach jeder Generierung die generierten Bilder und Bildobjekte für ein HMI-Gerät oder einen HMI-Gerätetyp angezeigt.

Zusätzlich können Sie die Zuordnung wie folgt anpassen:

- Bildobjekt in anderes Bild generieren
- Bild in anderes HMI-Gerät generieren

Geänderte Zuordnungen werden bei der nächsten Generierung wirksam. Abhängig von Ihren Einstellungen wird die Bildnavigation gleichzeitig mit angepasst.

Registerkarte "Bildobjekte -> Bilder"

In der Symbolleiste des Editors wählen Sie das HMI-Gerät aus, für das die Matrix unter "Zielgerät" angezeigt werden soll. SiVArc zeigt auch den Gerätetyp für alle Geräte an.

Weisen Sie in dieser Registerkarte ein generiertes Bildobjekt einem anderen Bild zu.

GettingStartedSiVArC_V2.0_Complete_V14 ▶ Common data ▶ SiVArC ▶ Generation matrix

Target device: HMI_RT_1 [WinCC RT Advanced] (1)

Screen items -> Screens Screens -> HMI devices

Call structure	Screen rule	Name of the screen object	HMI devices	Plantsection1	Plantsection2	Plantsection3
1 PLC_1						
2 Main						
3 Plantsection, Plan...	Plantsection_Status_SymbIO	Plantsection1_DB_SymbIO	For all	X		
4 Plantsection, Plan...	Plantsection_Title	Plantsection1_DB	For all	X		
5 Plantsection, Plan...	Plantsection_Status_SymbIO	Plantsection2_DB_SymbIO	For all		X	
6 Plantsection, Plan...	Plantsection_Title	Plantsection2_DB	For all		X	
7 Plantsection, Plan...	Plantsection_Status_SymbIO	Plantsection3_DB_SymbIO	For all			X
8 Plantsection, Plan...	Plantsection_Title	Plantsection3_DB	For all			X

Screen name	Layout field
1 Plantsection1	
2 Plantsection2	
3 Plantsection3	X

Registerkarte "Bilder -> HMI-Geräte"

In der Symbolleiste des Editors wählen Sie den HMI-Gerätetyp aus, für den die Matrix unter "Geräte-Typ" angezeigt werden soll. Daraufhin zeigt der Editor die Bilder aller HMI-Geräte dieses Typs an.

Weisen Sie in dieser Registerkarte ein generiertes Bild einem anderen HMI-Gerät zu.

GettingStartedSiVArC_V2.0_Complete_V14 ▶ Gemeinsame Daten ▶ SiVArC ▶ Generier-Matrix

Geräte-Typ: [WinCC RT Advanced] (1024 x 768)

Bildobjekte -> Bilder Bilder -> HMI-Geräte

Bildname	HMI_RT_1
1 Plantsection1	<input checked="" type="checkbox"/>
2 Plantsection2	<input checked="" type="checkbox"/>
3 Plantsection3	<input checked="" type="checkbox"/>

Zuordnung generierter Bildobjekte und Bilder anpassen

1. Um die Zuordnung eines Bildobjekts zu ändern, wählen Sie auf der Registerkarte "Bildobjekte -> Bilder" in der entsprechenden Zelle das Positionierungsfeld oder "X" aus.
2. Um die Zuordnung eines Bildes zu ändern, aktivieren Sie auf der Registerkarte "Bilder -> HMI-Geräte" in der entsprechenden Zelle das Optionskästchen.
3. Generieren Sie die Visualisierung.

Navigationsschaltflächen für Bilder anpassen

Navigationsschaltflächen, die auf ein Bild führen, das über die Matrix neu generiert wird, werden entsprechend der Bildhierarchie ebenfalls neu generiert.

1. Aktivieren Sie die Option "SiVArc > SiVArc-Einstellungen > Matrix-Einstellungen > Navigationsschaltflächen generieren".
2. Ordnen Sie die Bilder neu zu.
3. Generieren Sie die Visualisierung.

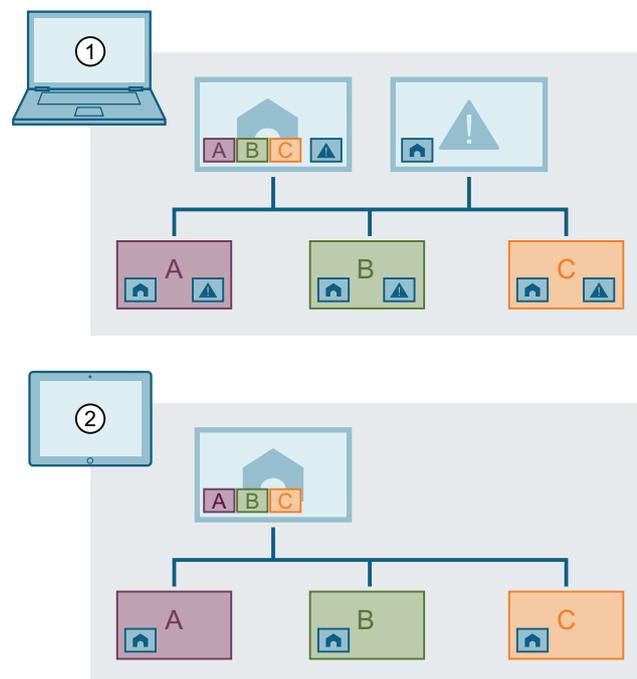
Die Bilder und Navigationsschaltflächen, die zu diesem Bild führen, werden neu generiert.

Projektieren Sie die Eigenschaft "Anzahl Überlaufbilder" eines Bildes und verschieben Sie die Eigenschaft in die Bibliothekstypen. Unter Verwendung eines Bildtyps als "Kopiervorlage/Typ eines Bildes" werden die Navigationsobjekte entsprechend generiert.

Beispiel: Verschieben von Bildern samt Navigation auf andere Geräte mit Hilfe der Generiermatrix

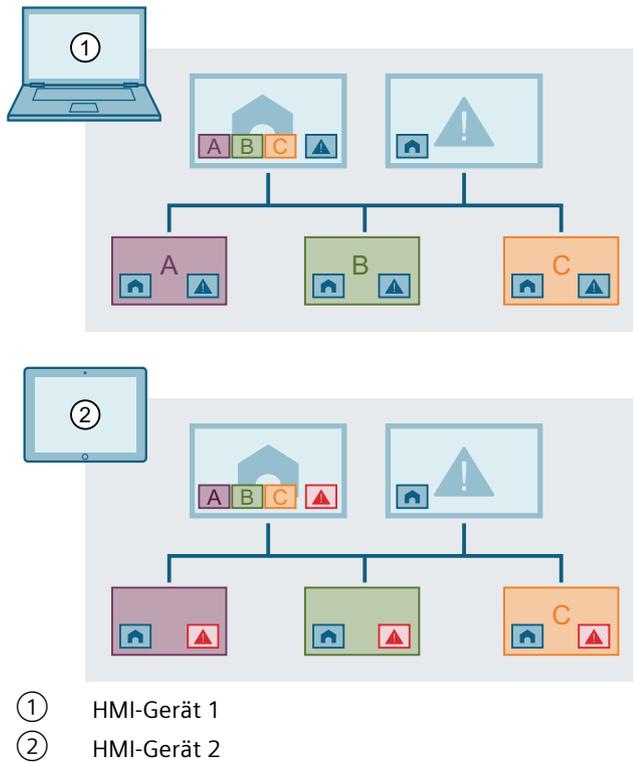
Sie haben auf dem HMI-Gerät 1 ein Startbild, ein Diagnosebild und unterlagerte Bilder generiert. Das Startbild und das Diagnosebild lassen sich von jedem unterlagerten Bild aus über Navigationsschaltflächen anzeigen.

Ein Diagnosebild wurde auf dem HMI-Gerät 2 jedoch nicht erstellt.



- ① HMI-Gerät 1
- ② HMI-Gerät 2

Wenn Sie das Diagnosebild mit der Generiermatrix auf das HMI-Gerät 2 verschieben, werden die Navigationsschaltflächen entsprechend angepasst.



Siehe auch

Beispiel: Generiermatrix einsetzen (Seite 224)

Anzeige in den SiVArC-Editoren bearbeiten (Seite 277)

6.5.3 Beispiel: Generiermatrix einsetzen

Beispielsszenario

Zum Zeitpunkt der Inbetriebnahme einer Anlage fehlt immer noch ein Bediengerät. Der Händler liefert nicht, und das Projekt droht in Verzug zu kommen. Sämtliche Inhalte des Bediengeräts werden daher in ein anderes Bediengerät integriert, bis der Händler das Bediengerät liefert.

Anforderung

Um eine reibungslose und verzugsfreie Inbetriebnahme zu gewährleisten, soll der Inbetriebnehmer die Anlagenstruktur im WinCC-Projekt kurzfristig ändern.

Umsetzungskonzept

Über die Generiermatrix verschiebt der Inbetriebnehmer generierte Anzeige- und Bedienobjekte sowie Bilder nach den neuen Anforderungen.

Bei der nächsten Generierung wird die Änderung ausgewertet. Die Navigation wird dabei automatisch hierarchisch angepasst.

Das ursprüngliche Projekt bleibt erhalten und kann erneut generiert werden, sobald das fehlende Bediengerät geliefert ist.

Siehe auch

Einsatz der Generiermatrix (Seite 221)

SiVArc Openness

7.1 Einleitung

Einleitung

Mit der TIA Portal Openness-Anwendung können Sie SiVArc instanziiieren. Sie benötigen eine Client-Anwendung für den Zugriff auf das TIA Portal und starten dann über die Openness-Funktion die SiVArc-Dienste. Weitere Einzelheiten zur Einrichtung und zum Zugriff auf Openness finden Sie im Benutzerhandbuch zum TIA Portal.

Einrichten der Anwendung

Um eine Client-Anwendung einzurichten, gehen Sie wie folgt vor:

1. Erstellen Sie eine Konsolenanwendung. Fügen Sie eine Referenz auf die Public API (Siemens.Engineering.dll) hinzu, verfügbar unter `_deployed\TIA\TIA\SP1_11010001\PublicAPI\15.1\936 Siemens.Engineering.dll` oder über die installierten Binärdateien unter `PublicAPI\15.1\937 Siemens.Engineering.dll`.
2. Fügen Sie der Konfigurationsdatei Konfigurationsdetails hinzu. Ausführliche Informationen über die Konfigurationsdetails und den Zugriff auf die Public API finden Sie im TIA Openness Wiki.
3. Für den Zugriff auf den SiVArc-Dienst verwenden Sie die nachfolgend erwähnte API:

```
using (TiaPortal tia = new
TiaPortal (TiaPortalMode.WithUserInterface))
{
    Project myProject = tia.Projects.Open(new FileInfo(@"C:\Users
\z003exve\Documents\Automation\Project_Demo\Project_Demo.ap15));
    //Wenn SiVArc nicht installiert ist, kann der Benutzer nicht auf
den SiVArc-Dienst zugreifen (Übersetzungsfehler)
    Sivarc sivarc =myproject?.GetService<Sivarc>():
    if (sivarc !=null)
    {
    }
}
```

7.2 SiVArc-Diensteigenschaften

SiVArc-Diensteigenschaften

Die nachfolgende Tabelle führt die unterstützten Eigenschaften und Methoden für SiVArc auf:

Name der Eigenschaft	Beschreibung	Datentyp
AlarmRules	Ankerobjekt für alle Alarmregelobjekte	AlarmRulesBrowsable
ScreenRules	Ankerobjekt für alle Bildregelobjekte	ScreenRulesBrowsable
TextlistRules	Ankerobjekt für alle Textlistenregelobjekte	TextlistRulesBrowsable
TagRules	Ankerobjekt für alle Variablenregelobjekte	TagRulesBrowsable
CopyRules	Ankerobjekt für alle Kopierregelobjekte	CopyRulesBrowsable
Alarm Rules	Enumeration sämtlicher direkter Alarmregeln der ersten Ebene	AlarmRuleComposition
Groups	Enumeration sämtlicher direkter Alarmregelgruppen der ersten Ebene	AlarmRuleGroupComposition
ScreenRules	Enumeration sämtlicher direkter Bildregeln der ersten Ebene	ScreenRuleComposition
ScreenRules-Groups	Enumeration sämtlicher direkter Bildregelgruppen der ersten Ebene	ScreenRuleGroupComposition
TextlistRules	Enumeration sämtlicher direkter Textlistenregeln der ersten Ebene	TextlistRuleComposition
TextlistGroups	Enumeration sämtlicher direkter Textlistenregelgruppen der ersten Ebene	TextlistRuleGroupComposition
TagRules	Enumeration sämtlicher direkter Variablenregeln der ersten Ebene	TagRuleComposition
TagRulesGroups	Enumeration sämtlicher direkter Variablenregelgruppen der ersten Ebene	TagRuleGroupComposition
CopyRules	Enumeration sämtlicher direkter Kopierregeln der ersten Ebene	CopyRuleComposition
CopyRulesGroups	Enumeration sämtlicher direkter Kopierregelgruppen der ersten Ebene	CopyRuleGroupComposition

In der folgenden Tabelle wird die Zusammensetzung von AlarmRule und AlarmRuleGroup aufgeführt. Gleiches gilt für andere SiVArc-Objekte.

Methodenname	Parameter	Beschreibung	Datentyp
Find	Zeichenkette – Alarmregel-/Regelgruppenname	Sucht die Alarmregel/Alarmregelgruppe in der Sammlung der Alarmregeln/Alarmgruppen	AlarmRule
CreateFrom	MasterCopy – Masterkopie Alarmregel/Alarmregelgruppe	Kopiert die Masterkopie der Alarmregel/Alarmregelgruppe aus der Bibliothek in das Projekt und verwendet die Standardoption zum Ersetzen	AlarmRule
CreateFrom	MasterCopy – Masterkopie Alarmregel/Alarmregelgruppe, CreateOptions – Umbenennen/ Ersetzen	Kopiert die Masterkopie der Alarmregel/Alarmregelgruppe aus der Bibliothek in das Projekt und verwendet die Option zum Erstellen	AlarmRule

7.3 Kopieren von Regeln oder Gruppen aus der Bibliothek

Voraussetzung

- Starten Sie die TIA Portal Openness-Anwendung. Weitere Informationen zu Verbindungen finden Sie im Benutzerhandbuch zum TIA Portal.
- Ein TIA Portal-Projekt mit Bildregeleditor, Bildregelgruppen und Masterkopie ist vorhanden.

Fall 1: Wenn Sie Regeln/Regelgruppen aus der Masterkopie in den Bildregeleditor kopieren

Über die globale Bibliothek "CreateFrom" können Regeln und Regelgruppen aus der globalen Bibliothek in den SiVArc-Regeleditor kopiert werden. Wenn erfolgreich, gibt die API-Funktion die ScreenRule/ScreenRuleGroup zurück. Der folgende Code zeigt, wie Regeln oder Regelgruppen aus der Masterkopie in den SiVArc-Editor kopiert werden:

```
// Finds screen rule master copy "Screen rule_1"
MasterCopy screenRuleMasterCopy =
    myProject.ProjectLibrary.MasterCopyFolder.MasterCopies.Find("Screen rule_1");

if (screenRuleMasterCopy != null)
{
    var rule = sivarcs.ScreenRules.Rules.CreateFrom(screenRuleMasterCopy);
    if (rule != null)
    {
        Console.WriteLine("Copied Screen Rule Name: " + rule.Name);
        Console.WriteLine("Copied Screen Rule Comment: " + rule.Comment);
        Console.WriteLine("Copied Screen Rule Enabled: " + rule.Enabled);
        Console.WriteLine("Copied Screen Rule Condition: " + rule.Condition);
    }
}
}
```

Standardmäßig ist das Verhalten "Ersetzen".

Fall 2: Wenn Sie Regeln/Regelgruppen aus der Masterkopie kopieren, können die vorhandenen Regeln/Regelgruppen basierend auf der zweiten Parametervorgabe umbenannt oder ersetzt werden.

Wenn Regeln/Regelgruppen der Masterkopie bereits im SiVArc-Editor vorhanden sind und Sie versuchen, diese zu kopieren, werden die Regeln/Regelgruppen umbenannt. Die API "CreateOptions" erstellt die Regeln/Regelgruppen im SiVArc-Editor, sofern sie nicht vorhanden sind, und ersetzt andernfalls bereits vorhandene Regeln/Regelgruppen. Wenn erfolgreich, benennt die API-Funktion die ScreenRule/ScreenRuleGroup um. Das folgende Code-Snippet zeigt das Ersetzen von Regeln/Regelgruppen:

```
// Finds screen rule master copy "Screen rule_1"
MasterCopy screenRuleMasterCopy =
    myProject.ProjectLibrary.MasterCopyFolder.MasterCopies.Find("Screen rule_1");

if (screenRuleMasterCopy != null)
{
    var rule = sivarcs.ScreenRules.Rules.CreateFrom(screenRuleMasterCopy, CreateOptions.Rename);
    if (rule != null)
    {
        Console.WriteLine("Copied Screen Rule Name: " + rule.Name);
        Console.WriteLine("Copied Screen Rule Comment: " + rule.Comment);
        Console.WriteLine("Copied Screen Rule Enabled: " + rule.Enabled);
        Console.WriteLine("Copied Screen Rule Condition: " + rule.Condition);
    }
}
}
```

7.4 Suchen von Bildregeln oder Bildregelgruppen

Voraussetzung

- Die TIA Portal Openness-Anwendung ist mit dem TIA Portal verbunden. Weitere Informationen zu Verbindungen finden Sie im Benutzerhandbuch zum TIA Portal.
- Ein TIA Portal-Projekt mit Bildregeln und Bildregelgruppen ist vorhanden.

Suchen von Bildregeln in Bildgruppen

Fall 1: Suchen von Bildregeln im Bildregeleditor.

Mit der API `sivarc.ScreenRules.Rules` können Sie die verfügbaren Regeln im SiVArc-Bildregeleditor wie nachfolgend gezeigt finden:

```
// Collection of all immediate first level screen rules
ScreenRuleComposition screenRules = sivarc.ScreenRules.Rules;
if(screenRules != null && screenRules.Count > 0)
{
    // Finds screen rule
    ScreenRule rule = screenRules.Find("screen rule_7");
    if(rule != null)
    {
        Console.WriteLine("Screen Rule Name: " + rule.Name);
        Console.WriteLine("Screen Rule Comment: " + rule.Comment);
        Console.WriteLine("Screen Rule Enabled: " + rule.Enabled);
        Console.WriteLine("Screen Rule Condition: " + rule.Condition);
    }
}
```

Fall 2: Suchen von Bildregelgruppen im Bildregeleditor.

Mit der API `sivarc.ScreenRules.Groups` können Sie die verfügbaren Regeln und Regelgruppen im SiVArc-Bildregeleditor wie nachfolgend gezeigt finden:

```

var groups = sivarC.ScreenRules.Groups;
if (groups != null && groups.Count > 0)
{
    var rule = groups.Find("Screen rule group").Rules.Find("Screen rule_2");
    if (rule != null)
    {
        Console.WriteLine("Found Screen Rule Name: " + rule.Name);
        Console.WriteLine("Found Screen Rule Comment: " + rule.Comment);
        Console.WriteLine("Found Screen Rule Enabled: " + rule.Enabled);
        Console.WriteLine("Found Screen Rule Condition: " + rule.Condition);
    }

    var group = groups.Find("Screen rule group").Groups.Find("Screen rule group_2");
    if (group != null)
    {
        Console.WriteLine("Found Screen Rule Group Name: " + group.Name);
        Console.WriteLine("Found Screen Rule Group Comment: " + group.Comment);
        Console.WriteLine("Found Screen Rule Group Enabled: " + group.Enabled);
        Console.WriteLine("Found Screen Rule Group Condition: " + group.Condition);
    }
}

```

7.5 Löschen von Regeln und Regelgruppen

Voraussetzung

- Die TIA Portal Openness-Anwendung ist mit dem TIA Portal verbunden. Weitere Informationen zu Verbindungen finden Sie im Benutzerhandbuch zum TIA Portal.
- Ein TIA Portal-Projekt mit Bildregeln und Bildregelgruppen ist vorhanden.

Löschen von Regeln und Regelgruppen

Um eine Regel oder Regelgruppe zu löschen, verwenden Sie die folgende API:

```
sivarC.ScreenRules.Rules.Find("Screen rule_1").Delete();
```

ScreenRules ist ein Ankerobjekt für alle Bildregelobjekte. Um eine Löschung durchzuführen, ist es obligatorisch, die Regel im Regeleditor zu finden. Weitere Informationen zum Suchen der Bildregel finden Sie im Abschnitt "Suchen von Bildregeln oder Bildregelgruppen".

Um Bilder aus einer Bildgruppe zu löschen, verwenden Sie die folgende API:

```
sivarC.ScreenRules.Rules.Find("Screen rule group_1").Delete();
```

7.6 UMAC-Einrichtung für Openness

Wissenswertes zu UMAC

Um über Openness auf UMAC zuzugreifen, stellen Sie sicher, dass Sie die erforderlichen UMAC-Anmeldedaten und Zugriffsrechte haben. Wenn Sie kein gültiger UMAC-Benutzer sind, gibt die Anwendung für alle SiV Arc-Ankerregelobjekte den Wert `NULL` zurück. Weitere Informationen zu UMAC finden Sie im Thema "UMAC" im SiV Arc-Benutzerhandbuch.

7.7 SiV Arc-Generierung

Voraussetzung

- Starten Sie die TIA Portal Openness-Anwendung. Weitere Informationen zu Verbindungen finden Sie im Benutzerhandbuch zum TIA Portal.
- Ein mit einem HMI-Gerät verbundenes TIA Portal-Projekt ist vorhanden und der PLC konfiguriert.

Wichtige Hinweise

- Stellen Sie sicher, dass die SiV Arc-Lizenz auf Ihrem PC installiert ist, andernfalls wird während der Generierung eine Ausnahme gemeldet - *"SiV Arc-Lizenz fehlt; eine SiV Arc-Lizenz ist für die Änderung von Daten obligatorisch"*.
- Stellen Sie sicher, dass ein gültiger Gerätenamen verwendet wird, andernfalls wird eine Ausnahme gemeldet - *"HMI-Gerät 'Gerätenamen' nicht gefunden"*.
- Stellen Sie sicher, dass ein gültiger PLC-Name aufgerufen wird, andernfalls wird eine Ausnahme gemeldet - *"PLC-Gerät 'PLC-Gerätenamen' nicht gefunden"*.
- Stellen Sie sicher, dass der Name eines unterstützten Geräts verwendet wird, andernfalls wird eine Ausnahme gemeldet - *"HMI-Gerät 'Gerätenamen' wird nicht unterstützt"*.
- Stellen Sie sicher, dass der Name eines unterstützten PLC verwendet wird, andernfalls wird eine Ausnahme gemeldet - *"PLC-Gerät 'PLC-Gerätenamen' wird nicht unterstützt"*.
- Stellen Sie sicher, dass Sie einen gültigen Parameter `GenerationOption` übergeben. Wird kein Parameter übergeben, erfolgt die SiV Arc-Generierung, und standardmäßig werden dafür die TIA Portal-Projekteinstellungen verwendet.
- Stellen Sie sicher, dass Sie einen gültigen PLC-Namen verwenden, der nicht bei der vorherigen Generierung verwendet wurde, da ansonsten das System einfriert.

Generierungsoptionen - Enum (Merker)

SiV Arc unterstützt Enum-Optionen, und Sie können in der API `Generate` eine Kombination aus zwei Werten übergeben. In der folgenden Tabelle werden die Enum-Optionen aufgeführt:

SN	Werte	Beschreibung
1	None	Keine Auswahl getroffen, für die Generierung werden die Standardeinstellungen übernommen
2	AllTags	Es werden alle Variablen generiert
3	UsedHmi- Tags	Es werden nur relevante (verwendete) Variablen generiert
4	FullGenerati- on	Wenn die Option FullGeneration nicht ausgewählt ist, entscheidet SiVArc intern basierend auf der Konfiguration, ob eine vollständige Generierung oder eine Delta-Generierung durchgeführt werden muss. Wenn Sie den Parameter FullGeneration übergeben, führt SiVArc eine erzwungene vollständige Generierung durch.

Um SiVArc zu generieren, verwenden Sie die folgende API:

```
sivarc.Generate("HMI_1", new List<string> {PLC_1},
GenerateOptions.AllTags | GenerateOptions.FullGeneration);
```

SiVArcGenerationResult und SivarcFeedbackMessage

Bei der SiVArc-Generierung wird bei erfolgreicher Generierung auf die folgenden Eigenschaften zugegriffen:

- IsGenerationSuccessful - Informiert darüber, ob die SiVArc-Generierung erfolgreich ist.
- WarningCount - Gesamtzahl von Warnungen nach der SiVArc-Generierung
- ErrorCount - Gesamtzahl der Fehler nach der SiVArc-Generierung
- Messages - Zusammensetzung der Rückmeldung

Um das SiVArc-Ergebnis zu generieren, verwenden Sie die folgende API:

- Print Sivarc generated feedback messages

```
private void WriteSivarcGenerationResults(SivarcGenerationResult result)
{
    sb.Append("Is SiV Arc generation successful:" +
        result.IsGenerationSuccessful);
    sb.Append(Environment.NewLine);
    sb.Append("Total Warning Count:" + result.WarningCount);
    sb.Append(Environment.NewLine);
    sb.Append("Total Error Count:" + result.ErrorCount);
    sb.Append(Environment.NewLine);

    RecursivelyWriteMessages(result.Messages);
}
```

Bei der SiV Arc-Generierung wird bei erfolgreicher Generierung auf die folgenden Rückmeldungen zugegriffen:

- Path: Header-Text der Rückmeldung (Header-Meldungen haben immer ein leeres Beschreibungsfeld)
- DateTime: Datum und Uhrzeit der Rückmeldung
- MessageType: Typ der Rückmeldung
- Description: Beschreibung/Inhalt der Rückmeldung (nur wenn der Pfad leer ist, um sicherzustellen, dass es sich nicht um eine Header-Meldung handelt)
- WarningCount: Anzahl der Warnungen für eine Header-Meldung
- ErrorCount: Anzahl der Fehler für eine Header-Meldung
- Messages: Zusammensetzung der Rückmeldung (SivarcFeedbackMessage)

Mit dem folgenden Code-Snippet können Sie rekursive Rückmeldungen anzeigen:

```
private void RecursivelyWriteMessages(SivarcFeedbackMessageComposition messages)
{
    foreach (SivarcFeedbackMessage message in messages)
    {
        sb.Append("Path: " + message.Path);
        sb.Append(Environment.NewLine);
        sb.Append("DateTime: " + message.DateTime);
        sb.Append(Environment.NewLine);
        sb.Append("State: " + message.MessageType);
        sb.Append(Environment.NewLine);
        sb.Append("Description: " + message.Description);
        sb.Append(Environment.NewLine);
        sb.Append("Warning Count: " + message.WarningCount);
        sb.Append(Environment.NewLine);
        sb.Append("Error Count: " + message.ErrorCount);
        sb.Append(Environment.NewLine);
        sb.Append(Environment.NewLine);

        RecursivelyWriteMessages(message.Messages, indent);
    }
}
```


Referenz

8.1 SiVArc-Objekte

8.1.1 PLC-Variablen

Beschreibung

PLC-Variablen stellen Variablen dar, die von der PLC unterstützt werden.

Verwendung

Sie können das PLC-Variablenobjekt verwenden, um generierte externe PLC-unterstützte Variablen in der Projektnavigation in strukturierter Form zu speichern.

Verwenden Sie das Objekt "PLC-Variable" wie folgt:

- Objekteigenschaft "SymbolicName"
`PLCTag.DB.SymbolicName`
Zugriff auf den benutzerdefinierten Namen des Datenbausteins.

8.1.2 Peripheriegerät

Beschreibung

Das I-Device ist als IO-Gerät mit einem übergeordneten IO-Controller verbunden.

Verwendung

Sie können das Objekt IODevice für den Zugriff auf ein Peripheriegerät im Projekt verwenden.

Verwenden Sie das Objekt "Device.name" wie folgt:

- Objekteigenschaft "Name"
`IODevice.Name`
Zugriff auf den benutzerdefinierten Namen eines IO-Geräts, z. B. HMI_1.

8.1.3 Software Units

Beschreibung

Sie können Ausdrücke für verschiedene Eigenschaften von HMI-Objekten konfigurieren, die in den Namen der Software Unit mit dem PLC-Baustein, der in der SiVArc-Regelerstellung verwendet wird, aufgelöst werden.

Verwendung

Sie können das Objekt softwareunit für den Zugriff auf das Software Unit-Gerät im Projekt verwenden.

Verwenden Sie das Objekt "Device.name" wie folgt:

- Objekteigenschaft "Name"
SoftwareUnit.Name
Zugriff auf den benutzerdefinierten Namen einer Software Unit.

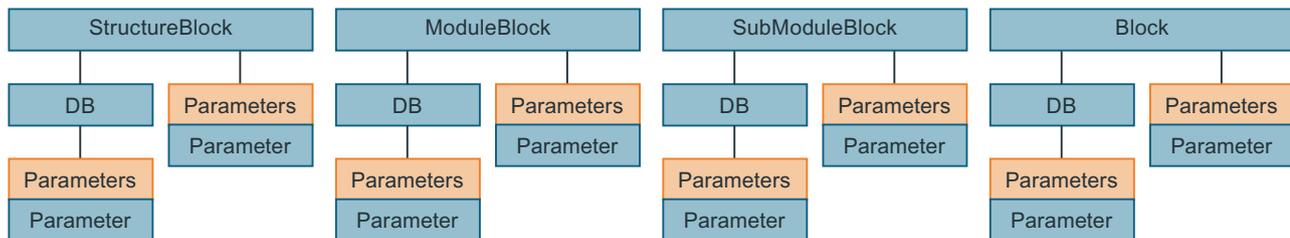
8.1.4 Objekthierarchie

Einleitung

Mit SiVArC-Ausdrücken können Sie Daten aus verschiedenen Bereichen des TIA Portals direkt adressieren.

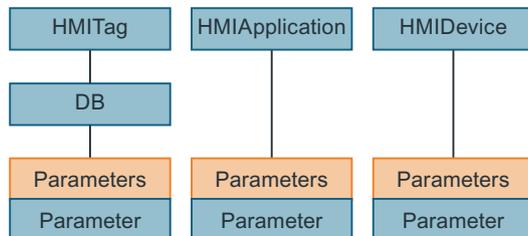
Programmaufruf in STEP7

Über Schlüsselworte greifen Sie auf die Bausteine im Anwenderprogramm, die dazugehörigen Datenbausteine und deren Parameter zu.



WinCC-Daten

Über folgende Schlüsselworte greifen Sie auf externe Variablen, Geräte und Anwendungen der Visualisierung zu.



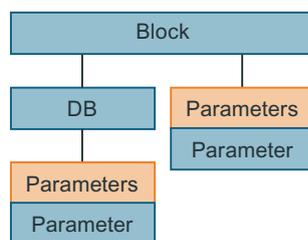
Bibliotheks-Daten

Über das Schlüsselwort `LibraryObject` greifen Sie auf den Ablageort einer Generiervorlage in der Bibliothek zu.

LibraryObject

8.1.5 Block (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt den Programmbaustein dar, der gerade von SiVArC bearbeitet wird - unabhängig von dessen Position innerhalb der Aufrufhierarchie.

Verwendung

Sie verwenden das Objekt "Block" folgendermaßen:

- Objekteigenschaft "FolderPath"
`Block.FolderPath`
 Greift auf den Ablagepfad des Bausteins in der Projektnavigation innerhalb des Ordners "Programmbausteine" zu, z. B. "Plant\Plantsection\Productionline"
- Objekteigenschaft "Name"
`Block.Name`
 Greift auf den internen Namen des Bausteins zu, z. B. "FB1".
- Objekteigenschaft "SymbolicName"
`Block.SymbolicName`
 Greift auf den benutzerdefinierten Namen des Bausteins zu.
- Objekteigenschaft "NetworkComment"
`Block.NetworkComment`
 Greift auf den Kommentar zu, der im Netzwerk des Bausteins eingetragen ist.
- Objekteigenschaft "NetworkTitle"
`Block.NetworkTitle`
 Greift auf den Titel des Netzwerks zu, in dem der Baustein instanziiert ist.
- Objekteigenschaft "Number"
`ModuleBlock.DB.Number`
 Greift auf die Bausteinnummer in den Bausteineigenschaften zu.

8.1 SiVArC-Objekte

- Auflistung "Parameters"
`ModuleBlock.Parameters("Activate").Value`
 Greift auf einen Bausteinparameter zu.
- Objekteigenschaft "SymbolComment"
`Block.SymbolComment`
 Greift auf den benutzerdefinierten Kommentar in den Bausteineigenschaften zu.
- Objekteigenschaft "Title"
`Block.Title`
 Greift auf die Überschrift des Bausteins in den Bausteineigenschaften zu.
- Objekteigenschaft "Version"
`Block.Version`
 Wenn der Block eine Instanz eines Bausteintyps ist, greift dieser Ausdruck auf die Typ-Version des Bausteintyps in der Bibliothek zu.
- Auflistung "Parameters"
`Block.Parameters(<Name Parameter>).AssignedTag.Comment`
 Greift auf den Kommentar einer Variable zu, die dem Bausteinparameter zugewiesen ist.

8.1.6 DB (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Stellt den Datenbaustein eines Bausteins dar. Das DB-Objekt ist ein SiVArC-Objekt der zweiten Hierarchieebene. Dem DB-Objekt ist immer ein Baustein aus der Auftrufhierarchie oder das Objekt `HMITag` vorangestellt.

Verwendung

Sie verwenden das Objekt "DB" folgendermaßen:

- Objekteigenschaft "Comment"
`ModuleBlock.DB.Comment`
 Greift auf den Kommentar in den Bausteineigenschaften zu.
- Objekteigenschaft "FolderPath"
`HMITag.DB.FolderPath`
 Greift auf den Ablagepfad des Bausteins in der Projektnavigation innerhalb des Ordners "Programmbausteine" zu, z. B. "DBs\Plant"
- Objekteigenschaft "Number"
`SubModuleBlock.DB.Number`
 Greift auf die Bausteinnummer in den Bausteineigenschaften zu.
- Objekteigenschaft "SymbolicAddress"
`StructureBlock.DB.SymbolicAddress`
 Greift auf den benutzerdefinierten Namen des Datenbausteins zu.
 Wenn der Datenbaustein eine Multiinstanz ist, wird die symbolische Adresse des Bausteins zurückgegeben.

- Objekteigenschaft "TagPrefix"
`StructureBlock.DB.TagPrefix`
Greift auf den benutzerdefinierten Namen des Datenbausteins zu.
Wenn der Datenbaustein eine Multiinstanz ist, wird die symbolische Adresse im HMI-Format zurückgegeben. Als Trennzeichen wird "_" statt "." zwischen dem Namen des Datenbausteins und dem Namen Variablen verwendet.
- Objekteigenschaft "SymbolicName"
`HMI_Tag.DB.SymbolicName`
Greift auf den benutzerdefinierten Namen des Datenbausteins zu.
- Objekteigenschaft "Type"
`ModuleBlock.DB.Type`
Greift auf den Typ des Datenbausteins zurück: Einzelinstanz (IDB) oder Multiinstanz (MDB).

Siehe auch

Einfluss der Mehrsprachigkeit auf eine Generiervorlage (Seite 124)

8.1.7 HMIApplication (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

HMIApplication

Stellt die Runtime-Software auf einem Bediengerät dar.

Verwendung

Über das HMIApplication-Objekt greifen Sie auf eine Runtime-Anwendung eines Bediengeräts zu.

Sie verwenden das Objekt "HMIApplication" folgendermaßen:

- Objekteigenschaft "Name"
`HMIApplication.Name`
Greift auf den benutzerdefinierten Namen der Runtime-Software eines Bediengeräts zu, z. B. RT_HMI_1.
- Objekteigenschaft "Type"
`HMIApplication.Type`
Greift auf die Art der Runtime-Software zu, z. B. WinCC RT Advanced.

Hinweis

Wenn Ihr Bediengerät ein Panel ist, sind die Objekte HMIDevice und HMIApplication gleichbedeutend.

8.1.8 HMIDevice (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt ein Bediengerät im Projekt dar.

Verwendung

Über das HMIDevice-Objekt greifen Sie auf ein Bediengerät im Projekt zu.

Sie verwenden das Objekt "HMIDevice" folgendermaßen:

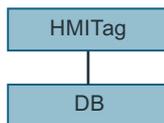
- Objekteigenschaft "Name"
`HMIDevice.Name`
Greift auf den benutzerdefinierten Namen eines Bediengeräts zu, z. B. HMI_1.
- Objekteigenschaft "Type"
`HMIDevice.Type`
Greift auf die Art des Bediengeräts zu, z. B. KTP400.

Hinweis

Wenn Ihr Bediengerät ein Panel ist, sind die Objekte HMIDevice und HMIApplication gleichbedeutend.

8.1.9 HMI Tag (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt eine externe Variable dar.

Verwendung

Über das HMITag-Objekt legen Sie generierte externe Variablen in der Projektnavigation strukturiert ab.

Hinweis

Verwendungsmöglichkeiten

Sie verwenden das HMITag-Objekt ausschließlich im Editor "Variablenregeln".

Sie verwenden das Objekt "HMITag" folgendermaßen:

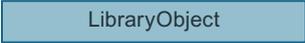
- Objekteigenschaft "FolderPath"
`HMITag.DB.FolderPath`
Greift auf den Ablagepfad des Blocks in der Projektnavigation innerhalb des Ordners "Programmbausteine" zu, z. B. "Plant\Plantsection\Productionline"
- Objekteigenschaft "SymbolicName"
`HMITag.DB.SymbolicName`
Greift auf den benutzerdefinierten Namen des Datenbausteins zu.

Siehe auch

Ablagestrategien für generierte Objekte (Seite 126)

8.1.10 LibraryObject (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

A rectangular icon with a light blue background and a thin black border. The text "LibraryObject" is centered within the rectangle in a dark blue, sans-serif font.

Stellt einen Bildtyp in der Projektbibliothek dar.

Verwendung

Sie verwenden das `LibraryObject`-Objekt ausschließlich in der SiVArc-Eigenschaften "Name" und "Bildgruppe" einer Generiervorlage für ein Bild.

- Objekteigenschaft "FolderPath"

`LibraryObject.FolderPath`

Referenziert den Ablagepfad des Bildtyps in der Bibliothek. Wenn Sie den SiVArc-Ausdruck in der SiVArc-Eigenschaft "Bildgruppe" verwenden, wird der Ablagepfad aus der Bibliothek in der Projektnavigation erzeugt. Wenn Sie den SiVArc-Ausdruck in der SiVArc-Eigenschaft "Name" verwenden, wird das generierte Bild nach dem Ordner benannt, in dem der Bildtyp abgelegt ist. .

Hinweis

Sie können diesen Ausdruck unter "Name" nur in Bezug auf eine einstufige Ablagehierarchie in der Bibliothek verwenden. Wenn Sie mehrstufige Ablagehierarchie nutzen möchten, können Sie als Ersatz für den Backslash den Ausdruck `LibraryObject.FolderPath` nutzen.

- Objekteigenschaft "Name"

`LibraryObject.Name`

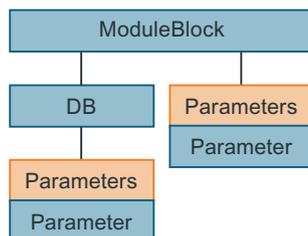
Referenziert den Namen des Bildtyps der Bibliothek. Wenn Sie den SiVArc-Ausdruck in der SiVArc-Eigenschaft "Bildgruppe" verwenden, wird das Bild in einem Ordner mit dem Namen des Bildtyps in der Projektnavigation abgelegt. Wenn Sie den SiVArc-Ausdruck in der SiVArc-Eigenschaft "Name" verwenden, wird das Bild nach dem Bildtyp benannt.

Siehe auch

Ablagestrategien für generierte Objekte (Seite 126)

8.1.11 ModuleBlock (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt den Programmbaustein der zweiten Ebene der Aufrufhierarchie dar. Mit dem `ModuleBlock`-Objekt adressieren Sie den Baustein der zweiten Ebene absolut.

Verwendung

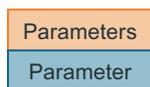
Über das `ModuleBlock`-Objekt können Sie auf verschiedene Eigenschaften des Bausteins und des dazugehörigen Datenbausteins zugreifen.

Sie verwenden das Objekt "ModuleBlock" folgendermaßen:

- Objekteigenschaft "FolderPath"
`ModuleBlock.FolderPath`
 Greift auf den Ablagepfad des Bausteins in der Projektnavigation innerhalb des Ordners "Programmbausteine" zu, z. B. "Plant\Plantsection\Productionline"
- Objekteigenschaft "Name"
`ModuleBlock.Name`
 Greift auf den internen Namen des Bausteins zu, z. B. "FB1".
- Objekteigenschaft "NetworkComment"
`ModuleBlock.NetworkComment`
 Greift auf den Kommentar zu, der im Netzwerk des Bausteins eingetragen ist.
- Objekteigenschaft "NetworkTitle"
`ModuleBlock.NetworkTitle`
 Greift auf den Titel des Netzwerks zu, in dem der Baustein instanziiert ist.
- Objekteigenschaft "Number"
`ModuleBlock.DB.Number`
 Greift auf die Bausteinnummer in den Bausteineigenschaften zu.
- Auflistung "Parameters"
`ModuleBlock.Parameters("Activate").Value`
 Greift auf einen Bausteinparameter zu.
- Objekteigenschaft "SymbolComment"
`ModuleBlock.SymbolComment`
 Greift auf den benutzerdefinierten Kommentar in den Bausteineigenschaften zu.
- Objekteigenschaft "SymbolicName"
`ModuleBlock.SymbolicName`
 Greift auf den benutzerdefinierten Namen des Bausteins zu.
- Objekteigenschaft "Title"
`ModuleBlock.Title`
 Greift auf die Überschrift des Bausteins in den Bausteineigenschaften zu.
- Objekteigenschaft "Version"
`ModuleBlock.Version`
 Wenn der Block eine Instanz eines Bausteintyps ist, greift dieser Ausdruck auf die Typ-Version des Bausteintyps in der Bibliothek zu.

8.1.12 Parameters (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Das Parameters-Objekt ist eine Auflistung aller Parameter am Baustein. Das Parameter-Objekt stellt einen Parameter im angegebenen Datenbaustein oder Baustein dar.

Verwendung

Über das Parameters-Objekt können Sie auf einen bestimmten Parameterwert am Baustein zugreifen.

Sie verwenden das Objekt "Parameters" folgendermaßen:

- Objekteigenschaft "Assigned"
`StructureBlock.Parameters("<Name Parameter>").Value`
 Gibt TRUE zurück, wenn der Parameter zugewiesen ist.
- Objekteigenschaft "Comment"
`Parameters("<Name Parameter>").Comment`
 Greift auf den Kommentar des Parameters zu.
- Objekteigenschaft "InitialValue"
`Parameters("<Name Parameter>").InitialValue`
 Greift auf den Standardwert des Parameters zu.
- Objekteigenschaft "Value"
`Parameters("<Name Parameter>").Value`
 Greift auf den Wert des Parameters zu.

8.1.13 S7Control (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Stellt eine PLC im Projekt dar.

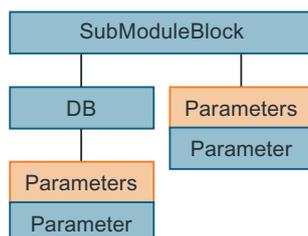
Verwendung

Das S7Control-Objekt verwenden Sie, um auf den Namen einer PLC zuzugreifen:

- Objekteigenschaft "Name"
`S7Control.Name`

8.1.14 SubModuleBlock (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt den Programmbaustein der dritten Ebene der Aufrufhierarchie dar. Mit dem `SubModuleBlock`-Objekt adressieren Sie den Baustein der dritten Ebene absolut.

Verwendung

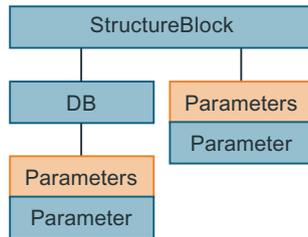
Über das `SubModuleBlock`-Objekt können Sie auf verschiedene Eigenschaften des Bausteins und seines Datenbausteins zugreifen.

Sie verwenden das Objekt "SubModuleBlock" folgendermaßen:

- Objekteigenschaft "FolderPath"
`SubModuleBlock.FolderPath`
Greift auf den Ablagepfad des Bausteins in der Projektnavigation innerhalb des Ordners "Programmbausteine" zu, z. B. "Plant\Plantsection\Productionline"
- Objekteigenschaft "Name"
`SubModuleBlock.Name`
Greift auf den internen Namen des Bausteins zu, z. B. "FB1".
- Objekteigenschaft "NetworkComment"
`SubModuleBlock.NetworkComment`
Greift auf den Kommentar zu, der im Netzwerk des Bausteins eingetragen ist.
- Objekteigenschaft "NetworkTitle"
`SubModuleBlock.NetworkTitle`
Greift auf den Titel des Netzwerks zu, in dem der Baustein instanziiert ist.
- Objekteigenschaft "Number"
`SubModuleBlock.DB.Number`
Greift auf die Bausteinnummer in den Bausteineigenschaften zu.
- Auflistung "Parameters"
`SubModuleBlock.Parameters("Activate").Value`
Greift auf einen Bausteinparameter zu.
- Objekteigenschaft "SymbolComment"
`SubModuleBlock.SymbolComment`
Greift auf den benutzerdefinierten Kommentar in den Bausteineigenschaften zu.
- Objekteigenschaft "SymbolicName"
`SubModuleBlock.SymbolicName`
Greift auf den benutzerdefinierten Namen des Bausteins zu.
- Objekteigenschaft "Title"
`SubModuleBlock.Title`
Greift auf die Überschrift des Bausteins in den Bausteineigenschaften zu.
- Objekteigenschaft "Version"
`SubModuleBlock.Version`
Wenn der Block eine Instanz eines Bausteintyps ist, greift dieser Ausdruck auf die Typ-Version des Bausteintyps in der Bibliothek zu.

8.1.15 StructureBlock (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung



Stellt den Programmbaustein der ersten Ebene der Aufrufhierarchie dar. Mit dem StructureBlock-Objekt adressieren Sie den Baustein der 1. Ebene absolut.

Verwendung

Über das StructureBlock-Objekt können Sie auf verschiedene Eigenschaften des Bausteins und seines Datenbausteins zugreifen.

Sie verwenden das Objekt "StructureBlock" folgendermaßen:

- Objekteigenschaft "FolderPath"
`SubModuleBlock.FolderPath`
 Greift auf den Ablagepfad des Bausteins in der Projektnavigation innerhalb des Ordners "Programmbausteine" zu, z. B. "Plant\Plantsection\Productionline"
- Objekteigenschaft "Name"
`SubModuleBlock.Name`
 Greift auf den internen Namen des Bausteins zu, z. B. "FB1".
- Objekteigenschaft "NetworkComment"
`SubModuleBlock.NetworkComment`
 Greift auf den Kommentar zu, der im Netzwerk des Bausteins eingetragen ist.
- Objekteigenschaft "NetworkTitle"
`SubModuleBlock.NetworkTitle`
 Greift auf den Titel des Netzwerks zu, in dem der Baustein instanziiert ist.
- Objekteigenschaft "Number"
`SubModuleBlock.DB.Number`
 Greift auf die Bausteinnummer in den Bausteineigenschaften zu.
- Auflistung "Parameters"
`SubModuleBlock.Parameters("Activate").Value`
 Greift auf einen Bausteinparameter zu.
- Objekteigenschaft "SymbolComment"
`SubModuleBlock.SymbolComment`
 Greift auf den benutzerdefinierten Kommentar in den Bausteineigenschaften zu.
- Objekteigenschaft "SymbolicName"
`SubModuleBlock.SymbolicName`
 Greift auf den benutzerdefinierten Namen des Bausteins zu.

- Objekteigenschaft "Title"
SubModuleBlock.Title
Greift auf die Überschrift des Bausteins in den Bausteineigenschaften zu.
- Objekteigenschaft "Version"
SubModuleBlock.Version
Wenn der Block eine Instanz eines Bausteintyps ist, greift dieser Ausdruck auf die Typ-Version des Bausteintyps in der Bibliothek zu.

8.1.16 TagNaming (Panels, Comfort Panels, RT Advanced, RT Professional)

Beschreibung

Stellt die Runtime-Einstellungen für Variablen dar.

Verwendung

Über das TagNaming-Objekt greifen Sie auf die in den Runtime-Einstellungen für Variablen gewählten Ersatz-Trennzeichen für die Unterebenen des Pfades der PLC-Variablen zu.

Sie verwenden das Objekt "TagNaming" folgendermaßen:

- Objekteigenschaft "SeparatorChar"
TagNaming.SeparatorChar
- Objekteigenschaft "IndexStartChar"
TagNaming.IndexStartChar
- Objekteigenschaft "IndexEndChar"
TagNaming.IndexEndChar

Rückgabewerte

Die Steuerung "PLC1" enthält den strukturierten Datenbaustein "DB1". Das Datenbausteinelement "Db1.a[1].b.c[3]" wird in einem Bild verwendet. Abhängig von Ihren Einstellungen gibt das TagNaming-Objekt folgende Werte zurück:

Rückgabewerte	WinCC-Variablenname	Gewählte Runtime-Einstellung
TagNaming.SeparatorChar = "."	Db1_a[1].b.c[3]	Kompatibilitätsmodus
TagNaming.IndexStartChar = "["	Plc1.Db1.a[1].b.c[3]	PLC-Präfix
TagNaming.IndexEndChar = "]"	Db1.a[1].b.c[3]	Trennzeichen ersetzen ohne Zeichenauswahl An der Verwendungsstelle im Bild wird der Variablenname in Anführungszeichen gesetzt: "Db1.a[1].b.c[3]"

Rückgabewerte	WinCC-Variablenname	Gewählte Runtime-Einstellung
TagNaming.SeparatorChar = ; TagNaming.IndexStartChar = "(" TagNaming.IndexEndChar = ")"	Db1;a(1);b;c(3)	Punkt und Klammer ersetzen mit ; ()
TagNaming.SeparatorChar = "_" TagNaming.IndexStartChar = "{" TagNaming.IndexEndChar = "}"	Plc1_Db1_a{10}_b_c{3}	Punkt und Klammer ersetzen mit _ { } PLC-Präfix

8.2 SiVArC-Objekteigenschaften

8.2.1 Assigned

Beschreibung

Gibt TRUE zurück, wenn eine Zuweisung am angegebenen Bausteinparameter vorhanden ist.

Syntax

```
<Object>.Assigned
```

Object

- Parameter

8.2.2 Comment

Beschreibung

Gibt den eingetragenen Kommentar zurück.

Syntax

```
<Object>.Comment
```

Object

- Parameter
- DB

Bemerkung

Wenn Sie den Kommentar eines Datenbausteins abfragen, wird der Kommentar aus den Bausteineigenschaften zurückgegeben.

Wenn Sie den Kommentar eines Parameters abfragen, wird der Kommentar aus der Symboltabelle zurückgegeben.

Mehrsprachigkeit

Der SiVArc-Ausdruck "DB.Comment" ist mehrsprachig projektierbar.

Siehe auch

Einfluss der Mehrsprachigkeit auf eine Generiervorlage (Seite 124)

8.2.3 FolderPath

Beschreibung

Gibt den Ablagepfad zurück.

Syntax

```
<Object>.FolderPath
```

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB
- LibraryObject

Bemerkung

Wenn Sie den Ablagepfad eines Programmbausteins abfragen, wird der Ablagepfad innerhalb des Ordners "Programmbausteine" zurückgegeben.

Wenn Sie den Ablagepfad eines Bibliotheksobjekts abfragen, wird der Ablagepfad innerhalb der Ordner "Mastercopies" oder "Types" zurückgegeben.

Als Trennzeichen zwischen der Ordnerhierarchie wird ein "\" zurückgegeben.

Siehe auch

Ablagestrategien für generierte Objekte (Seite 126)

8.2.4 HMITagPrefix

Beschreibung

Gibt den Wert der Eigenschaft "TagPrefix" eines Bildfensters zurück.

Die Eigenschaft "TagPrefix" ist z. B. der Namen des zugehörigen Datenbausteins des Programmbausteins, den SiVArc gerade auswertet.

Syntax

<Object>.HMITagPrefix

Object

- DB

8.2.5 IndexEndChar

Beschreibung

Gibt die in den Runtime-Einstellungen gesetzte schließende Klammer bei strukturieren externen Variablen zurück.

Syntax

<Object>.IndexEndChar

Object

- TagNaming

8.2.6 IndexStartChar

Beschreibung

Gibt die in den Runtime-Einstellungen gesetzte öffnende Klammer bei strukturieren externen Variablen zurück.

Syntax

<Object>.IndexStartChar

Object

- TagNaming

8.2.7 InitialValue

Beschreibung

Gibt den Standardwert eines Parameters zurück.

Syntax

```
<Object>.InitialValue
```

Object

- Parameter

8.2.8 Name

Beschreibung

Gibt den internen Namen zurück, z. B. "FB1"

Syntax

```
<Object>.Name
```

Object

- S7Control
- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- HMIApplication
- HMIDevice

Siehe auch

Ablagestrategien für generierte Objekte (Seite 126)

8.2.9 NetworkComment

Beschreibung

Gibt den Netzwerkkommentar zurück.

Syntax

```
<Object>.NetworkComment
```

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

Mehrsprachigkeit

Die Objekteigenschaft "NetworkComment" ist mehrsprachig projektierbar.

Siehe auch

Einfluss der Mehrsprachigkeit auf eine Generiervorlage (Seite 124)

8.2.10 NetworkTitle

Beschreibung

Gibt den Netzwerktitel zurück.

Syntax

```
<Object>.NetworkTitle
```

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

Mehrsprachigkeit

Die Objekteigenschaft "NetworkTitle" ist mehrsprachig projektierbar.

Siehe auch

Einfluss der Mehrsprachigkeit auf eine Generiervorlage (Seite 124)

8.2.11 Number

Beschreibung

Gibt die Bausteinnummer zurück.

Syntax

`<Object>.Number`

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB

8.2.12 SeparatorChar

Beschreibung

Gibt das in den Runtime-Einstellungen gesetzte Trennzeichen zurück.

Das Trennzeichen wird zwischen den Unterebenen des Pfades der PLC-Variable gesetzt, die im synchronisierten Namen der externen Variable enthalten sind.

Syntax

`<Object>.SeparatorChar`

Object

- TagNaming

8.2.13 SymbolComment

Beschreibung

Gibt den benutzerdefinierten Kommentar in den Bausteineigenschaften zurück.

Syntax

`<Object>.SymbolComment`

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB

Mehrsprachigkeit

Die Objekteigenschaft "SymbolComment" ist mehrsprachig projektierbar.

Siehe auch

Einfluss der Mehrsprachigkeit auf eine Generiervorlage (Seite 124)

8.2.14 SymbolicName

Beschreibung

Gibt den benutzerdefinierten Namen eines Bausteins oder einer Variablen zurück.

Syntax

`<Object>.SymbolicName`

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block
- DB
- HMITag

Bemerkungen

Wenn Sie den benutzerdefinierten Namen eines Datenbaustein abfragen, der als Multiinstanz (MDB) aufgerufen wird, dann wird der in der Bausteinschnittstelle abgelegte Name des Bausteins aufgerufen. Der Bausteinname für MDBs ist in den unter den statischen Lokaldaten abgelegt.

8.2.15 Title

Beschreibung

Gibt den Bausteintitel zurück.

Syntax

```
<Object>.Title
```

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

Mehrsprachigkeit

Die Objekteigenschaft "Title" ist mehrsprachig projektierbar.

Siehe auch

Einfluss der Mehrsprachigkeit auf eine Generiervorlage (Seite 124)

8.2.16 Type

Beschreibung

Gibt den Typ zurück.

Syntax

```
<Object>.Type
```

Object

- DB
- HMIApplication
- HMIDevice

Bemerkung

Wenn Sie den Typ eines Datenbausteins abfragen, wird der Typ "MDB" (Multiinstanzbaustein) oder "IDB" (Instanzbaustein) als String zurückgegeben.

8.2 SiVArC-Objekteigenschaften

Wenn Sie den Typ eines Bediengeräts abfragen, wird die Geräteart als String zurückgegeben, z. B. "KTP400".

Wenn Sie den Typ einer Runtime-Software abfragen, wird die Art der Software als String zurückgegeben, z. B. "WinCC RT Advanced".

8.2.17 Value

Beschreibung

Gibt den Wert zurück.

Syntax

```
<Object>.Value
```

Object

- Parameter

8.2.18 Version

Beschreibung

Gibt die Version eines Bausteintyps zurück.

Syntax

```
<Object>.Version
```

Object

- StructureBlock
- ModuleBlock
- SubModuleBlock
- Block

Bemerkung

Die Eigenschaft wird nur ausgewertet, wenn der gerade von SiVArC ausgewertete Baustein eine Instanz eines Bausteintyps in der Bibliothek ist.

8.3 SiVArc-Objekteigenschaften

Zugriff auf Bediengeräte

Mithilfe der folgenden Variablen können Sie in der Projektnavigation auf Bediengeräte zugreifen.

SiVArc-Objekteigenschaft	Adressierte -Eigenschaft
<code>HmiDevice.Name</code>	Name des Bediengeräts in der Projektnavigation Beispiel: "HMI_1", "PC_system_1"
<code>HmiDevice.Type</code>	Art des Bediengeräts in der Projektnavigation Beispiel: "KTP700 Mobile", "SIMATIC PC-Station"
<code>HmiApplication.Name</code>	Name der Anwendung Beispiel: "HMI_1", "HMI_RT_40"
<code>HmiApplication.Type</code>	Typ der Anwendung Beispiel: "WinCC RT Advanced", "WinCC RT Professional"
<code>LayoutFieldIndex</code>	Indexwert des bei der Generierung verwendeten Positionierungsfeldes Zur Verwendung in SiVArc-Ausdrücken

Handelt es sich bei dem Bediengerät um ein Panel, sind `HmiDevice` und `HmiApplication` identisch.

SiVArc-Objekteigenschaften für den Namen der Steuerung und für externe Variablen

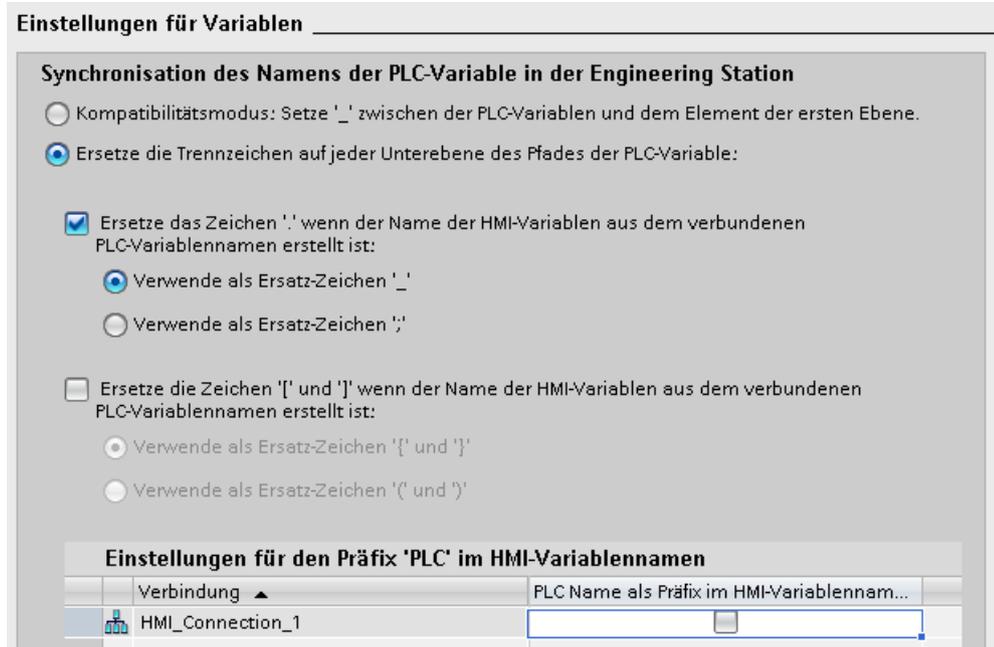
Um den Namen der S7-Steuerung zu referenzieren oder externe Variablen zu generieren, verwenden Sie die SiVArc-Objekteigenschaften `Name` und `SymbolicName`:

Die SiVArc-Ausdrücke `HmiTag.SymbolicName` und `HmiTag.DB.SymbolicName` verwenden Sie ausschließlich im Editor "Variablenregeln".

SiVArc-Objekteigenschaft	Referenziertes Objekt	Formulierung im SiVArc-Ausdruck
<code>Name</code>	Name der S7-PLC	<code>S7Control.Name</code>
<code>SymbolicName</code>	Name der externen Variablen (Variablenname)	<code>HmiTag.SymbolicName</code>
<code>DB.SymbolicName</code>	Name des DB	<code>HmiTag.DB.SymbolicName</code>
<code>DB.FolderPath</code>	Pfad der DBs	<code>HmiTag.DB.FolderPath</code>

SiVArc-Objekteigenschaften für den Namensabgleich von externen Variablen

Wie die Namen von PLC-Variablen und externen Variablen beim Synchronisieren abgeglichen werden, legen Sie in den Runtime-Einstellungen des Bediengeräts fest.



Um die Namen von externen Variablen entsprechend den Einstellungen für Variablen im TIA Portal mit SiVArc abzugleichen, verwenden Sie TagNaming-Variablen.

SiVArc-Objekteigenschaft	Referenziertes Objekt	Formulierung im SiVArc-Ausdruck
SeparatorChar	Die folgenden Trennzeichen auf den einzelnen Unterebenen des PLC-Variablenpfads: ". " " _ ";"	TagNaming.SeparatorChar
IndexStartChar	Die folgenden Trennzeichen auf den einzelnen Unterebenen des PLC-Variablenpfads: "[" "(" "{"	TagNaming.IndexStartChar
IndexEndChar	Die folgenden Trennzeichen auf den einzelnen Unterebenen des PLC-Variablenpfads: "]" ")" "}"	TagNaming.IndexEndChar

Siehe auch

Editor "Variablenregeln" (Seite 29)

8.4 Funktionen

8.4.1 Funktionen in SiVArc

In SiVArc sind die im folgenden Abschnitt aufgeführten Funktionen definiert.

Funktionen verwenden Sie in SiVArc-Ausdrücken. Die Funktionsnamen können Sie nicht ändern.

8.4.2 Funktion "Contains"

Funktion `Contains`

Die Funktion `Contains` ermittelt, ob eine Zeichenkette in einer anderen Zeichenfolge enthalten ist. Die Funktion berücksichtigt Groß- und Kleinschreibung sowie Leerzeichen.

Funktion	Ergebnis
<code>Contains("ButtonText", "Text")</code>	True
<code>Contains("ButtonText", "ttonT")</code>	True
<code>Contains("ButtonText", "butt")</code>	False
<code>Contains("ButtonText", "txeT")</code>	False
<code>Contains("ButtonText", " Text")</code>	False
<code>Contains("ButtonText", "Text ")</code>	False
<code>Contains("ButtonText", "Te xt")</code>	False
<code>Contains("Butto nText", "on")</code>	False
<code>Contains("ButtonText 1", "ButtonText 2")</code>	False

8.4.3 Funktion "EndsWith"

Funktion `EndsWith`

Die Funktion `EndsWith` ermittelt, ob das Ende einer Zeichenkette mit einer angegebenen Zeichenfolge übereinstimmt. Die Funktion berücksichtigt Groß- und Kleinschreibung sowie Leerzeichen.

Funktion	Ergebnis
<code>EndsWith("ButtonText", "Text")</code>	True
<code>EndsWith("ButtonText", "ButtonText")</code>	True
<code>EndsWith("ButtonText", "butt")</code>	False
<code>EndsWith("ButtonText", "Butt")</code>	False
<code>EndsWith("ButtonText", " Text")</code>	False
<code>EndsWith("ButtonText", "Text ")</code>	False

Funktion	Ergebnis
<code>EndsWith("ButtonText", "Text")</code>	False
<code>EndsWith("ButtonText", "t")</code>	True
<code>EndsWith("ButtonText", "T")</code>	False
<code>EndsWith("ButtonText ", "Text")</code>	False
<code>EndsWith("ButtonText 1", "ButtonText 2")</code>	False

8.4.4 Funktion "Format"

Funktion `Format`

Die Funktion `Format` gibt einen formatierten String zurück. Anweisungen innerhalb eines Format-String geben vor, in welcher Form der String zurückgegeben wird.

Die Funktion besitzt zwei Funktionsparameter:

- String, der formatiert zurückgegeben wird.
- Format-String, der die Formatierung des String vorgibt.
Um das Ergebnis als Binärcode anzuzeigen, verwenden Sie den Format-String "b". Wenn das Ergebnis eines Ausdrucks eine Gleitzahl ist, wird das Ergebnis im Binärformat gerundet angezeigt.

Funktion	Ergebnis
<code>Format(5, "0.00")</code>	5,00
<code>Format((VAR_1 Or 2#111100), "b")</code>	2#11101

Weiterführende Informationen zu Format-String finden Sie im Microsoft Developer Network unter dem Suchbegriff "Strings.Format-Methode".

8.4.5 Funktion "FormatNumber"

Funktion `FormatNumber`

Die Funktion `FormatNumber` gibt einen als Zahl formatierten String zurück.

Die Funktion besitzt fünf Funktionsparameter:

Position	Parameter	Beschreibung	Anmerkungen
1	Ausdruck	String, der als Zahl formatiert zurückgegeben wird	Wenn der String nicht als Zahl formatierbar ist (z. B. "hello world"), wird ein Fehler angezeigt.
2	AnzahlZiffernNachDezimalpunkt	Zahl, die angibt, wie viele Nachkommastellen rechts vom Dezimaltrennzeichen angezeigt werden: <code>FormatNumber ("12,4", 3, -2, -2, -2) ("12 Komma 4") = 12,400</code>	Der Standardwert -1 gibt an, dass die Ländereinstellungen des Computers verwendet werden.
3	FührendeZifferAufnehmen	Zahl, die angibt, ob bei Bruchzahlen eine führende 0 angezeigt wird oder nicht: <code>FormatNumber ("0,4", 3, -1, -2, -1) = 0,400</code>	Die möglichen Einstellungen sind unter "Liste der Konstanten" aufgeführt.
4	ÜbergeordneteAlsNegativeZahlenVerwenden	Zahl, die angibt, ob negative Werte in Klammern gesetzt werden oder nicht: <code>FormatNumber ("-12", 1, -2, -1, 0) = (12,0)</code>	Wenn die Zahl in Klammern gesetzt wird, fällt das Minuszeichen weg. Die möglichen Einstellungen sind unter "Liste der Konstanten" aufgeführt.
5	ZiffernGruppieren	Zahl, die angibt, ob große Zahlen mit dem Tausendertrennzeichen gruppiert werden oder nicht: <code>FormatNumber ("1288,4", 3, -2, -2, 0) = 1288,400</code>	Die Form des Tausendertrennzeichens (z. B. Punkt, Komma oder Leerzeichen) ist in den Ländereinstellungen des Computers festgelegt. Die möglichen Einstellungen sind unter "Liste der Konstanten" aufgeführt.

Liste der Konstanten

Tabelle 8-1 FührendeZifferAufnehmen

FührendeZifferAufnehmen	Wert	
TRUE	-1	Führende 0 anzeigen.
FALSE	0	Führende 0 nicht anzeigen.
UseDefault	-2	Ländereinstellungen des Computers verwenden.

Tabelle 8-2 ÜbergeordneteAlsNegativeZahlenVerwenden

FührendeZiffer-Aufnehmen	Wert	
TRUE	-1	Negative Werte in Klammern setzen. Das Minuszeichen fällt weg.
FALSE	0	Negative Werte ohne Klammern ausgeben. Das Minuszeichen wird angezeigt.
UseDefault	-2	Ländereinstellungen des Computers verwenden.

Tabelle 8-3 ÜbergeordneteAlsNegativeZahlenVerwenden

FührendeZiffer-Aufnehmen	Wert	
TRUE	-1	Zahlen mit dem Tausendertrennzeichen gruppieren.
FALSE	0	Zahlen nicht mit dem Tausendertrennzeichen gruppieren.
UseDefault	-2	Ländereinstellungen des Computers verwenden.

Beispiele

Die folgende Tabelle gilt für deutsche Ländereinstellungen. Das Tausendertrennzeichen ist der Punkt und das Dezimaltrennzeichen ist das Komma.

Funktion	Ergebnis
<code>FormatNumber ("12,4", 3, -2, -2, -2)</code> ("12 Komma 4")	12,400
<code>FormatNumber ("12.4", 3, -2, -2, -2)</code> ("12 Punkt 4")	124,000
<code>FormatNumber ("1288,4", 3, -2, -2, -1)</code>	1.288,400
<code>FormatNumber ("1288,4", 3, -2, -2, 0)</code>	1288,400
<code>FormatNumber ("-12", 1, -2, -2, 0)</code>	-12,0
<code>FormatNumber ("-12", 1, -2, -1, 0)</code>	(12,0)

8.4.6 Funktion "InStr"

Funktion InStr

Die Funktion `InStr` prüft, ob ein String vollständig in einem anderen String enthalten ist. Groß-/Kleinschreibung wird berücksichtigt. Die Funktion gibt einen Boole'schen Wert ("True" oder "False") zurück.

Die Funktion besitzt zwei Funktionsparameter:

- String, in dem geprüft wird.
- String, der den Vergleichstext enthält.

Die folgenden Beispiele zeigen, welche Werte die Funktion `InStr` ausgibt:

Funktion	Ergebnis
<code>InStr("Hallo", "Hallo")</code>	True
<code>InStr("Hallo", "hallo")</code>	False
<code>InStr("Hallo", "al")</code>	True
<code>InStr("12345", 3)</code>	True
<code>InStr("12345", "6")</code>	False

8.4.7 Funktion "IsDefined"

Funktion `IsDefined`

Die Funktion `IsDefined` überprüft anhand eines String als Parameter, ob eine Variable mit dem eingegebenen String als Name existiert.

Diese Funktion können Sie für folgende Syntax-Elemente anwenden:

- SiVArc-Variablen
- SiVArc-Objekteigenschaft
- Arrays vom Datentyp "String"

Beispiel: Folgende benutzerdefinierte Variable ist angelegt:

```
ButtonText "Takt_zeiten"
```

Funktion	Ergebnis
<code>IsDefined("ButtonText")</code>	True
<code>IsDefined("ButtonText[0]")</code>	True
<code>IsDefined("ButtonText[1]")</code>	True
<code>IsDefined("ButtonText[2]")</code>	False

8.4.8 Funktion "LBound"

Funktion `LBound`

Die Funktion `LBound` erwartet ein Array als Parameter und gibt den kleinsten möglichen Index zurück.

Funktion	Ergebnis
<code>LBound(Split("SG19_FG97_ST090", "_"))</code>	0
<code>LBound(Split("SG19_FG97", "_"))</code>	0

8.4.9 Funktion "Left"

Funktion `Left`

Die Funktion `Left` gibt eine Zeichenfolge zurück, die eine angegebene Anzahl von Zeichen ab dem am weitesten links stehenden Zeichen einer Zeichenfolge enthält.

Die Funktion besitzt zwei Funktionsparameter:

- String, aus dem ein Substring zurückgegeben wird.
- Zahl, welche die Zeichenlänge des Substring angibt
Wenn die Zahl 0 ist, wird ein leerer String zurückgegeben.
Wenn die Zahl größer ist als die Anzahl Zeichen im angegebenen String, wird ein Fehler angezeigt.

Funktion	Ergebnis
<code>Left("ButtonText", 6)</code>	"Button"
<code>Left("ButtonText", 0)</code>	"" (Leerer String)
<code>Left("ButtonText", "10")</code>	"ButtonText"
<code>Left("ButtonText", 11)</code>	Fehler (Zahl ist größer als die Zeichenanzahl im String)

8.4.10 Funktion "Len"

Funktion `Len`

Die Funktion `Len` gibt die Zeichenanzahl eines String zurück. Die Funktion erwartet als Funktionsparameter einen String.

Funktion	Ergebnis
<code>Len("ButtonText")</code>	10
<code>Len("")</code>	0
<code>Left("ButtonText", Len("ButtonText"))</code>	"ButtonText"

8.4.11 Funktion "LTrim"

Funktion LTrim

Die Funktion `LTrim` entfernt führende Leerzeichen eines String. Die Funktion erwartet als Funktionsparameter einen String.

Funktion	Ergebnis
<code>LTrim (" ButtonText")</code>	"ButtonText"
<code>LTrim ("ButtonText")</code>	"ButtonText"

8.4.12 Funktion "Max"

Funktion Max

Die Funktion `Max` erwartet zwei Zahlen als Parameter und gibt die größere der beiden zurück.

Funktion	Ergebnis
<code>Max (12, 3)</code>	12
<code>Max (3, 123)</code>	123

8.4.13 Funktion "Mid"

Funktion Mid

Die Funktion `Mid` gibt eine Teilzeichenfolge innerhalb eines String ab einer bestimmten Position zurück.

Die Funktion besitzt drei Funktionsparameter:

- String, aus welchem die Teilzeichenfolge kopiert wird.
- Zahl, welche die Startposition im String angibt.
Wenn die Startposition größer ist als die Zeichenlänge des String, wird ein Fehler angezeigt.
- Zahl, welche die Länge der Teilzeichenfolge ab der Startposition angibt.
Wenn die angegebene Länge größer ist als die längste mögliche Teilzeichenkette ab der Startposition im String, wird ein Fehler angezeigt.

Funktion	Ergebnis
<code>Mid("ButtonText", 5, 3)</code>	"nTe"
<code>Mid("ButtonText", 0, 10)</code>	"ButtonText"

Funktion	Ergebnis
Mid("ButtonText", 10, 3)	Fehler (Startposition ist größer als die Zeichenlänge des String)
Mid("ButtonText", 7, 10)	Fehler (Angegebene Länge ist größer als die längste mögliche Teilzeichenkette ab Position 7)

8.4.14 Funktion "Min"

Funktion Min

Die Funktion `Min` erwartet zwei Zahlen als Parameter und gibt die kleinere der beiden zurück.

Funktion	Ergebnis
Min(12, 3)	3
Min(3, 123)	3

8.4.15 Funktion "Replace"

Funktion Replace

Die Funktion `Replace` sucht in einem String von links nach rechts nach einem Substring und ersetzt diesen durch einen anderen Substring. Die Suchfunktion berücksichtigt Groß-/Kleinschreibung. Der geänderte String wird zurückgegeben.

Die Funktion besitzt drei Funktionsparameter:

- String, in welchem ein Substring gesucht und ersetzt wird.
- String, der den gesuchten Substring angibt.
Wenn der gesuchte Substring ein leerer String ist, wird der zuerst übergebene String unverändert zurückgegeben.
- String, der anstelle des gesuchten Substring eingefügt wird.

Das Suchen und Ersetzen wird nach dem neu eingefügten Substring fortgesetzt.

Funktion	Ergebnis
Replace("ButtonText", "Text", "Button")	"ButtonButton"
Replace("ButtonText", "ButtonText", "Hello World")	"Hello World"
Replace("aaa", "aa", "bb")	"bba"
Replace("a", "a", "a")	"a"
Replace("a", "", "b")	"a"
Replace("aA", "a", "b")	"bA"

8.4.16 Funktion "Right"

Funktion `Right`

Die Funktion `Right` gibt einen Substring aus das am weitesten rechts stehenden Zeichen eines String zurück. Die Anzahl der Zeichen wird beim Funktionsaufruf angegeben.

Die Funktion besitzt zwei Funktionsparameter:

- String, aus dem der Substring erzeugt und zurückgegeben wird.
- Zahl, welche die Anzahl der am weitesten rechts stehenden Zeichen angibt, die zurückgegeben werden.
Wenn die Zahl 0 ist, wird ein leerer String zurückgegeben.
Wenn die Zahl größer ist als die Anzahl der Zeichen im String, wird ein Fehler angezeigt.

Funktion	Ergebnis
<code>Right("ButtonText", 4)</code>	"Text"
<code>Right("ButtonText", 0)</code>	"" (Leerer String)
<code>Right("ButtonText", 10)</code>	"ButtonText"
<code>Right("ButtonText", 11)</code>	Fehler (Zahl ist größer als die Anzahl der Zeichen im String)

8.4.17 Funktion "RTrim"

Funktion `RTrim`

Die Funktion `RTrim` entfernt Leerzeichen am Ende eines String. Der resultierende String wird zurückgegeben.

Wenn der String keine Leerzeichen am Ende enthält, wird der String unverändert zurückgegeben.

Funktion	Ergebnis
<code>RTrim("ButtonText ")</code>	"ButtonText"
<code>RTrim("ButtonText")</code>	"ButtonText"

8.4.18 Funktion "Split"

Funktion `Split`

Die Funktion `Split` teilt einen String auf. Das dafür erforderliche Trennzeichen ist frei definierbar.

Die Funktion besitzt zwei Funktionsparameter:

- String
- Trennzeichen

Abhängig von der Syntax wird ein Substring zurückgegeben oder die Anzahl der enthaltenen Substrings:

- Substring als Rückgabewert
`Split("<String>", "<Separator>") (<Index>)`
 Den Substring referenzieren Sie über einen Index, der bei Null beginnt.
- Anzahl enthaltener Substrings als Rückgabewert
`Split("<String>", "<Separator>").Length`

Die folgenden Beispiele zeigen, welche Zahlenwerte die Funktion `Split` ausgibt:

Funktion	Ergebnis
<code>Split("SG19_FG97_ST090", "_") (0)</code>	SG19
<code>Split("SG19.FG97.ST090", ".") (1)</code>	FG97
<code>Split("42", ".") (0)</code>	42
<code>Split(".", ".") (0)</code>	"" (Leerer String)

Die folgenden Beispiele zeigen, welche Anzahl an Substrings Funktion `Split` ausgibt:

Funktion	Ergebnis
<code>Split("SG19_FG97_ST090", "_").Length</code>	3
<code>Split("SG19.FG97.ST090", ".").Length</code>	3

8.4.19 Funktion "StartsWith"

Funktion `StartsWith`

Die Funktion `StartsWith` ermittelt, ob der Anfang einer Zeichenkette mit einer angegebenen Zeichenfolge übereinstimmt. Die Funktion berücksichtigt Groß- und Kleinschreibung sowie Leerzeichen.

Funktion	Ergebnis
<code>StartsWith("ButtonText", "Butt")</code>	True
<code>StartsWith("ButtonText", "butt")</code>	False
<code>StartsWith("ButtonText", "Text")</code>	False
<code>StartsWith("ButtonText", "ButtonText")</code>	True
<code>StartsWith("ButtonText", " Butt")</code>	False
<code>StartsWith("ButtonText", "Butt ")</code>	False
<code>StartsWith("ButtonText", "Bu tt")</code>	False
<code>StartsWith("ButtonText", "B")</code>	True
<code>StartsWith("ButtonText", "b")</code>	False

Funktion	Ergebnis
<code>StartsWith(" ButtonText", "Butt")</code>	False
<code>StartsWith("B uttonText", "Butt")</code>	False
<code>StartsWith("ButtonText 1", "ButtonText 2")</code>	False

8.4.20 Funktion "StrComp"

Funktion `StrComp`

Die Funktion `StrComp` vergleicht zwei Strings. Die Funktion sortiert die Strings alphanumerisch, beginnend mit dem ersten Zeichen, und berücksichtigt Groß-/Kleinschreibung. Abhängig von der Sortierung der Strings wird eine Zahl zurückgegeben.

Folgende Fälle sind möglich:

- Der erste String ist vor dem zweiten String einsortiert. Der Rückgabewert ist -1.
`StrComp("ABCD", "Abcd") = -1`
`StrComp("A", "a") = -1` ("A" liegt im Alphabet vor "a")
- Der zweite String ist vor dem ersten String einsortiert. Der Rückgabewert ist 1.
`StrComp("ABCD", "AAcd") = 1`
- Beide Strings sind identisch. Der Rückgabewert ist 0.
`StrComp("Abcd", "Abcd") = 0`

8.4.21 Funktion "TrailNum"

Funktion `TrailNum`

Die Funktion `TrailNum` gibt den letzten positiven Zahlenwert aus einem String zurück, z. B. die Nummer im Namen eines Programmbausteins.

Die folgenden Beispiele zeigen, welche Zahlenwerte die Funktion `TrailNum` ausgibt:

Funktion	Ergebnis
<code>TrailNum("42")</code>	42
<code>TrailNum("Anzahl42")</code>	42
<code>TrailNum("Anzahl0042")</code>	42
<code>TrailNum("Anzahl-42")</code>	42
<code>TrailNum("Minimum42_Maximum84")</code>	84

Die folgenden Beispiele zeigen, wie Sie die Funktion `TrailNum` in einem SiVArc-Ausdruck verwenden.

Im TIA Portal ist ein Funktionsbaustein mit dem symbolischen Namen "SG19_FG97_ST090+IR001_FB" programmiert.

SiVArc-Ausdruck	Ergebnis
"MyBlock_"&TrailNum (ModuleBlock.SymbolicName)	"MyBlock_1"
"MyBlock_"&TrailNum (ModuleBlock.SymbolicName[0])	"MyBlock_19"

Wenn Sie keine String-Indizierung angeben, wird der letzte Zahlenwert im String ausgegeben.

8.4.22 Funktion "Trim"

Funktion Trim

Die Funktion `Trim` entfernt alle Leerzeichen am Anfang und am Ende eines String. Der resultierende String wird zurückgegeben.

Wenn der String weder am Anfang und Ende Leerzeichen enthält, wird der String unverändert zurückgegeben.

Funktion	Ergebnis
<code>Trim(" ButtonText")</code>	"ButtonText"
<code>Trim("ButtonText ")</code>	"ButtonText"
<code>Trim(" ButtonText ")</code>	"ButtonText"
<code>Trim("ButtonText")</code>	"ButtonText"

8.4.23 Funktion "UBound"

Funktion UBound

Die Funktion `UBound` erwartet ein Array als Parameter und gibt den größten möglichen Index zurück.

Funktion	Ergebnis
<code>UBound(Split("SG19_FG97_ST090", "_"))</code>	2
<code>UBound(Split("SG19_FG97", "_"))</code>	1
<code>UBound(Split("", "."))</code>	0

8.5 Operatoren

Sie können folgende Operatoren in SiVArc-Ausdrücken verwenden.

Achten Sie bei der Verwendung der Operatoren auf die Groß-/Kleinschreibung. Dies betrifft zum einen bei den logischen und den bitweisen Operatoren die Groß-/Kleinschreibung der Operatoren selbst. Zum anderen müssen Sie bei Vergleichsoperatoren die Groß-/

Kleinschreibung der in Relation gesetzten Strings beachten, z. B. wenn Sie zwei Strings auf Namensgleichheit vergleichen.

Arithmetische Operatoren

Arithmetischer Operator	Beispiel	Ergebnis
+	4+2	6
-	4-2 -4+2	2 -2
*	4*2	8
/	4/2	2

Vergleichsoperatoren

Vergleichsoperator	Beispiel	Ergebnis
=	4=4 4=2	True False
<> ("verschieden von")	4<>4 4<>2	False True
>	4>2 2>4	True False
>=	4>=2 4>=4	True True
<	4<2 2<4	False True
<=	4<=2 4<=4	False True

Logische Operatoren

Logischer Operator	Beispiel	Ergebnis
And	True And True True And False False And False	True False False
Or	True Or True True Or False False Or False	True True False
Not	Not True Not False	False True

Bitweise Operatoren

Bitweiser Operator	Beispiel	Ergebnis
And	16 And 16	16
Or	8 Or 4	12
Xor	3 Xor 1	2

Operatoren für String-Ketten

Verknüpfungsoperator	Beispiel	Ergebnis
&	"Tool"&"Bar"	ToolBar

Priorität der Operatoren

Die folgende Tabelle gibt an, mit welcher Priorität die Operatoren verarbeitet werden, falls Sie mehrere Operatoren in einem SiVArC-Ausdruck verwenden. Dabei hat 1 die höchste Priorität.

Operator	Not - (unär)	*, /	+, -	&	=, <> >, >= <, <=	And	Or	Xor
Priorität	1	2	3	4	5	6	7	8

Um die Verarbeitungsreihenfolge zu ändern, verwenden Sie Klammern.

8.6 String-Indizierung

Verwendung

Innerhalb eines Strings werden Substrings durch das Zeichen `_` getrennt. Um auf einen Substring zuzugreifen, verwenden Sie den Indizierungsoperator `[]`.

Die Zählung der Substrings beginnt bei 0. Über die Zahl im Indizierungsoperator greifen Sie auf den Substring zu.

Beispiel

Im TIA Portal ist die Variable "FB_Name" mit dem Wert "SG19_FG97_ST090+IR001_FB" definiert.

String-Indizierung im SiVArC-Ausdruck	Ergebnis
FB_Name[0]	SG19
FB_Name[1]	FG97
FB_Name[2]	ST090+IR001
FB_Name[3]	FB

8.7 If-Bedingungen

Logische Bedingungen in SiVArC-Ausdrücken formulieren Sie mit dem If-Operator.

If-Operator

Der If-Operator hat folgende Syntax:

```
If(<condition>, <thenExpression>, <elseExpression>)
```

<condition> Boolean oder Integer

<thenExpression> wird ausgegeben, wenn <condition> entweder True oder ein Integer-Wert außer 0 ist

<elseExpression> wird ausgegeben, wenn <condition> entweder False oder 0 ist

Sie können die Bedingungen auch verschachteln und eine If-Bedingung in einer If-Bedingung verwenden.

Beispiele

If-Bedingung	Ergebnis
If(True, "An", "Aus")	An
If(0, "An", "Aus")	Aus
If(42, "An", "Aus")	An
If(4>2, If(False, 4, 2), 42)	2

8.8 Unterstützte Datentypen für PLC-Variablen

SiVArC unterstützt alle elementaren Datentypen, die in WinCC V13.1 von der PLC auf das Bediengerät abgebildet werden können.

Zusätzlich unterstützt SiVArC die strukturierten Datentypen ARRAY, STRUCT und UDT.

Elementare Datentypen

Name	Datentyp
BOOL	Boolescher Wert
BYTE	Dual- und Hexadezimalzahlen mit 8 Bits
CHAR	ASCII-Zeichen
DINT	Doppelinteger, Ganze Zahlen mit Vorzeichen
DTL	Datum und Uhrzeit (Jahr-Monat-Tag-Stunde:Minute:Sekunde.Nanosekunde)
DWORD	Dual- und Hexadezimalzahlen mit 32 Bits
DATE	IEC-Datum in Schritten von 1 Tag
DATE_AND_TIME	Datum und Uhrzeit (Jahr-Monat-Tag-Stunde:Minute:Sekunde; Festpunktzahl)

8.8 Unterstützte Datentypen für PLC-Variablen

Name	Datentyp
INT	Integer, Ganze Zahlen mit Vorzeichen
LDT	Datum und Uhrzeit (Jahr-Monat-Tag-Stunde:Minute:Sekunde)
LINT	
LREAL	
LTIME	
LTIME_OF_DAY	
LWORD	
REAL	Reelle Zahlen (IEEE Gleitpunktzahlen)
S5TIME	Zeitdauer im S5T#-Format, Step7-Zeit in Schritten von 10 ms
SINT	
STRING	Zeichenkette
TIME	Zeitdauer im IEC-Format, IEC-Zeit in Schritten von 1 ms, Ganze Zahl mit Vorzeichen
TIME_OF_DAY	Uhrzeit (Tageszeit) in Schritten von 1 ms
UDINT	
UINT	
ULINT	
USINT	
WORD	Dual- und Hexadezimalzahlen mit 16 Bits
WString	Unicode-Zeichenkette mit variabler Länge
WChar	Unicode-Zeichen mit 16 Bits

Strukturierte Datentypen

SiVArC unterstützt strukturierte PLC-Variablen und alle dazugehörigen Elemente, die für WinCC freigegeben sind. Beim Generieren erzeugt SiVArC strukturierte externe Variablen und Elemente entsprechend der PLC-Variable. Variablen und Elemente werden automatisch mit der PLC-Variablen und deren Elementen verschaltet.

Name	Datentyp
ARRAY	Feld
ARRAY DBs	
ARRAY DB STRUCT	
STRUCT	Struktur
UDT	User Defined Data Type (PLC-Datentyp)

Hinweis**Bedingung für PLC-Datentypen (UDTs)**

Wenn ein PLC-Datentyp ein Array eines strukturierten Datentyps (STRUCT oder UDT) ist, zerlegt SiVArc in WinCC das Array in einzelne Variablen diesen Datentyps. Wenn ein PLC-Datentyp Arrays von strukturierten Datentypen als Elemente enthält, werden diese als strukturierte Elemente im Editor "HMI-Variablen" abgebildet.

8.9 Unterstützte Systemfunktionen für Bildbausteine

Systemfunktionen

Abhängig davon, für welches Bediengerät generiert wird, verwenden Sie an SiVArc-Ereignissen folgende Systemfunktionen:

Systemfunktion	RT Advanced	RT Professional
ActivateScreen	x	x
DecreaseTag	x	x
IncreaseTag	x	x
InvertBit	x	x
InvertBitInTag	x	x
SetBit	x	x
SetBitInTag	x	x
SetTag	x	x
ResetBit	x	x
ResetBitInTag	x	x
ActivateScreenInScreenWindow	---	x
ActivatePreviousScreen	x	---
ShiftAndMask	x	---

8.10 Anzeige in den SiVArc-Editoren bearbeiten

Einleitung

Sie können SiVArc-Regeln im Editor oder in der Generierübersicht filtern und sortieren, ohne dabei die Generierreihenfolge zu beeinflussen. Die neue Anordnung speichern Sie bei Bedarf bis zum nächsten Start des TIA Portals. Außerdem können Sie in allen SiVArc-Editoren die Ansicht nach Spalten gruppieren. Die Filterfunktionen sind dann deaktiviert.

Während die Liste gefiltert oder sortiert ist, können Sie die SiVArc-Regeln weiter bearbeiten oder neue Regeln erstellen. Die aktiven Filterkriterien werden auch auf neue und bearbeitete Regeln angewandt.

Hinweis

Neue Regeln im gefilterten Editor

Wenn Sie im gefilterten Editor eine neue Regel anlegen, ist die neue Regel eine Kopie der an unterster Stelle angezeigten Regel. Wenn die Liste nach den Namen der SiVArc-Regeln gefiltert ist, wird die neue SiVArc-Regel nicht angezeigt.

Inhalte der Editoren zur Ansicht filtern

Wenn die Schaltfläche "Gruppieren" deaktiviert ist, können Sie die Inhalte der Editoren filtern.

Um SiVArc-Regeln im Editor zu filtern, gehen Sie folgendermaßen vor:

1. Klicken Sie in der Werkzeugleiste des Editors auf die Schaltfläche "Filter".
Unter der Kopfzeile des Editors wird eine Filterzeile angezeigt.
2. Öffnen Sie in der Filterzelle der gewünschten Spalte den Auswahldialog.
3. Selektieren Sie im Auswahldialog die Objekte, die Sie im Editor anzeigen wollen.
Die Regeln sind entsprechend Ihrer Auswahl gefiltert.

Um die Filterzeile auszublenden, klicken Sie erneut auf die Schaltfläche "Filter".

Inhalte der Editoren zur Ansicht sortieren

Wenn die Schaltfläche "Gruppieren" deaktiviert ist, können Sie die Inhalte der Editoren sortieren.

Sie können SiVArc-Regeln auch neu sortieren, während die Liste gefiltert dargestellt ist und umgekehrt.

Um SiVArc-Regeln im Editor zu sortieren, gehen Sie folgendermaßen vor:

- Klicken Sie auf die Spaltenüberschrift, nach der Sie die Anzeige sortieren wollen.
Die Anzeige ist alphabetisch absteigend nach der gewählten Spalte sortiert. Wenn der Regel-Editor Gruppen enthält, werden die Regeln innerhalb der Gruppen ebenfalls nach dieser Spalte sortiert.

Sortierung und Filter speichern

Um den Filter oder die Sortierung der Regeln bis zum nächsten Start des TIA Portals zu erhalten, gehen Sie folgendermaßen vor:

- Klicken Sie in der Werkzeugleiste des Editors auf die Schaltfläche "Anordnung merken".
Beim erneuten Öffnen des TIA Portals sind die SiVArc-Regeln wie zuletzt angeordnet und gefiltert.

Anzeige neu gruppieren

Beim ersten Öffnen der Anzeige werden die Inhalte gruppiert nach der ersten Spalte angezeigt.

Um die Inhalte im Editor neu zu gruppieren, gehen Sie folgendermaßen vor:

1. Um die Gruppierfunktion zu aktivieren, klicken Sie die Schaltfläche "Gruppieren". Die Schaltfläche "Gruppieren" wird gedrückt dargestellt.
2. Klicken Sie auf die Spaltenüberschrift, für deren Inhalte Sie die Anzeige gruppieren wollen. In der Anzeige werden alle SiVArc-Regeln oder SiVArc-Objekte nach den Inhalten der gewählten Spalte gruppiert.

8.11 Bildlegende

Übersicht

In den Illustrationen der SiVArc Dokumentation werden Symbole und abstrahierte Darstellungen wie folgt verwendet:

Symbol/Darstellung	Beschreibung
	Funktionsbaustein
	Instanz im OB1 (Aufruf im OB1)
	Netzwerk im OB1
	Typ eines Funktionsbausteins (Bibliothekstyp)
	Instanz eines Funktionsbaustein-Typs (Bibliothekstyp)
	Datenbaustein
SIV	SiVArc-Projektierung
	SiVArc-Ausdruck, der auf einen Datenbaustein zugreift
	SiVArc-Ausdruck, der auf einen Funktionsbaustein zugreift
	SiVArc-Ausdruck, der auf einen Eintrag am Netzwerk zugreift

Tooltipps

9.1 UMAC

9.1.1 CTHMISivarc

Siehe auch

[Erstellen von SiVArc-Regeln \(Seite 165\)](#)

[SiVArc-Regeln \(Seite 161\)](#)

[User Management Control \(UMAC\) in SiVArc \(Seite 26\)](#)

Index

A

- Ablagestrukturen, 182
- Aktualisierungszyklus, 182
- Ändern
 - Größe, 210
 - Rotationswinkel, 210
 - SiVArc-Regeln, 212
- Animation, 54
- Anlagenstruktur, 160
- Assigned-Eigenschaft (SiVArc), 250
- Aufbau
 - Registerkarte "SiVArc-Eigenschaften", 94
- Aufrufhierarchie, 121
- Ausdrücke
 - Bearbeiten, 107
- Auswertung
 - Regeln, 170

B

- Basisinstallation, 24
- Bausteinparameter, 97
- Bearbeiten
 - Ausdrücke, 107
- Bediengerät
 - Bildregel, 188
- Bedingung
 - Bildregel, 27
 - Textlistenregel, 30
 - Variablenregel, 29
- Bedingungen, 180
- Beispiel
 - Bildregel, 28
- Benennungskonvention, 175
- Benutzerdefiniertes Positionierungsschema, 63
- Bibliothek
 - SiVArc-Regeln, 213
- Bibliothekobjekt
 - Kopierregel, 35
- Bild
 - Bildregel, 27
- Bildablage, 160
- Bildbaustein, 188
- Bildbausteine
 - Systemfunktionen, 141
- Bilder, 162, 177

- Bildfenster
 - Bild ändern, 211
- Bildobjekt
 - Bildregel, 27
- Bildregel, 188
 - Bediengerät, 188
 - Bedingung, 27
 - Beispiel, 28
 - Bild, 27
 - Bildobjekt, 27
 - Generierübersicht, 44
 - Generiervorlage eines Bilds, 27
 - HMI-Gerät, 27
 - HMI-Objekt, 188
 - Kommentar, 27
 - Name, 27
 - Positionierungsfeld, 27, 188
 - Programmbaustein, 27
 - Steuerung, 27
- Bildregeln, 170
- Bildtyp, 188
- Bitmaske
 - Überlaufbilder, 77
- Block-Objekt (SiVArc), 239

C

- Comment-Eigenschaft (SiVArc), 250
- Contains-Funktion, 261

D

- DB-Objekt (SiVArc), 240
- Definition
 - Variablengenerierung, 173

E

- Eigenschaften (SiVArc)
 - Assigned, 250
 - Comment, 250
 - FolderPath, 251
 - HMI TagPrefix, 252
 - IndexEndChar, 252
 - IndexStartChar, 252
 - InitialValue, 253
 - NamGe, 253
 - NetworkComment, 253

- NetworkTitle, 254
- Number, 255
- SeparatorChar, 255
- SymbolComment, 255
- SymbolicName, 256
- Title, 257
- Type, 257
- Value, 258
- Version, 258
- Einsatz
 - Generiermatrix, 221
- Einstellungen für Variablen, 181
- EndsWith-Funktion, 261
- Erfassungsart, 182
- Exportieren
 - SiVArc-Regeln, 202

F

- Feste Positionierung, 63, 71, 210
- FolderPath-Eigenschaft (SiVArc), 251
- Format-Funktion, 262
- FormatNumber-Funktion, 262
- Formulierungsregeln
 - SiVArc-Ausdruck, 108
- Funktion, 53
- Funktionen, 261
 - Contains, 261
 - EndsWith, 261
 - Format, 262
 - FormatNumber, 262
 - InStr, 264
 - IsDefined, 265
 - LBound, 265
 - Left, 266
 - Len, 266
 - LTrim, 267
 - Max, 267
 - Mid, 267
 - Min, 268
 - Replace, 268
 - Right, 269
 - RTrim, 269
 - Split, 269
 - StartsWith, 270
 - StrComp, 271
 - TrailNum, 271
 - Trim, 272
 - UBound, 272
- Funktionsumfang, 13

G

- Generiermatrix, 211, 221, 224
 - Einsatz, 221
- Generierte Objekte
 - Tiefenstaffelung, 74
- Generierübersicht
 - Bildregel, 44
 - Programmbaustein, 44
- Generierung
 - Visualisierung, 49, 94
- Generiervorlage
 - Bildregel, 27
- Generiervorlagen, 180
- Geräte
 - Unterstützt, 79
- Globaler Datenbaustein, 186
- Grafiklisten, 177
- Größe
 - ändern, 210
- Gruppe, 160

H

- HMIApplication-Objekt (SiVArc), 241
- HMIDevice-Objekt (SiVArc), 242
- HMI-Gerät
 - Bildregel, 27
 - Kopierregel, 35
- HMI-Gerätetyp
 - Kopierregel, 35
- HMI-Objekt
 - Bildregel, 188
 - Textlistenregel, 189
- HMI-Objekte, 99
- HMI Tag-Objekt (SiVArc), 242
- HMI TagPrefix-Eigenschaft (SiVArc), 252

I

- Importieren
 - Regelgruppen, 203
 - SiVArc-Regeln, 204
- Importoptionen, 185, 203
- Index, 29
- IndexEndChar-Eigenschaft (SiVArc), 252
- IndexStartChar-Eigenschaft (SiVArc), 252
- InitialValue-Eigenschaft (SiVArc), 253
- Instanz-Datenbaustein, 186
- InStr-Funktion, 264

Interne HMI-Variablen, 173
 Interne Variablen, 177
 IsDefined-Funktion, 265

K

Know-how-Schutz
 einrichten, 205
 Kommentar
 Bildregel, 27
 Kopierregel, 35
 Textlistenregel, 30
 Variablenregel, 29
 Kopierregel
 Bibliotheksojekt, 35
 HMI-Gerät, 35
 HMI-Gerätetyp, 35
 Kommentar, 35
 Name, 34
 Kopierregeln, 172, 178
 Namenskonflikte, 179
 Kopiervorlage
 Textlistenregel, 30
 Kopiervorlagen von SiVArc-Regeln, 213

L

LBound-Funktion, 265
 Leerzeichen in Variablennamen, 175
 Left-Funktion, 266
 Len-Funktion, 266
 LibraryObject-Objekt (SiVArc), 243
 LTrim-Funktion, 267

M

Manuell erstellte Textlisteneinträge, 211
 Manuell erzeugte HMI-Objekte, 211
 Manuelle Änderungen, 209
 Max-Funktion, 267
 Mid-Funktion, 267
 Min-Funktion, 268
 ModuleBlock-Objekt (SiVArc), 244

N

Name
 Bildregel, 27
 Kopierregel, 34

Textlistenregel, 30
 Variablenregel, 29
 Name-Eigenschaft (SiVArc), 253
 Namensänderungen, 211
 Namenskonflikte
 Kopierregeln, 179
 Priorität, 173
 Navigationsschaltflächen, 77
 NetworkComment-Eigenschaft (SiVArc), 253
 NetworkTitle-Eigenschaft (SiVArc), 254
 Neupositionierung, 210
 Number-Eigenschaft (SiVArc), 255

O

Objekte (SiVArc)
 Block, 239
 DB, 240
 HMIApplication, 241
 HMIDevice, 242
 HMITag, 242
 LibraryObject, 243
 ModuleBlock, 244
 Parameters, 245
 S7Control, 246
 StructureBlock, 248
 SubModuleBlock, 246
 TagNaming, 249

P

Parameter
 Systemfunktion, 52
 Parameters-Objekt (SiVArc), 245
 Passwort, 204
 Positionierungsfeld
 Bildregel, 27, 188
 Positionierungsmethoden, 60
 Positionierungsschema, 72
 Priorität
 Namenskonflikte, 173
 Programmbaustein
 Bildregel, 27
 Generierübersicht, 44
 Textlistenregel, 30
 Programmbausteine, 106
 Sprachen, 106
 Projektierung
 Textlisteneinträge, 97

R

- Regelgruppen
 - Importieren, 203
- Regeln
 - Auswertung, 170
- Registerkarte "SiVArc-Animationen"
 - Layout, 55
- Registerkarte "SiVArc-Eigenschaften"
 - Aufbau, 94
- Replace-Funktion, 268
- Right-Funktion, 269
- Rotationswinkel
 - ändern, 210
- RTrim-Funktion, 269

S

- S7Control-Objekt (SiVArc), 246
- Schaltfläche
 - SiVArc-Ereignis, 141
- SeparatorChar-Eigenschaft (SiVArc), 255
- SiVArc
 - Verwendung, 13
- SiVArc-Ausdruck, 49, 53, 94
 - Formulierungsregeln, 108
 - Syntax, 109
- SiVArc-Bezug, 210
- SiVArc-Eigenschaft, 49, 94
- SiVArc-Ereignis, 141
 - Schaltfläche, 141
- SiVArc-Objekte, 108, 109
- SiVArc-Objekteigenschaften, 109
- SiVArc-Positionierungsschema, 63
- SiVArc-Regeln
 - Ändern, 212
 - Bibliothek, 213
 - Exportieren, 202
 - Importieren, 204
- SiVArc-Texte, 97
- SiVArc-Variablen, 109, 173
- Skript, 141
- Skripte, 162, 177
- Speicherstrukturen, 173
- Split-Funktion, 269
- Sprachen
 - Programmbausteine, 106
- StartsWith-Funktion, 270
- Steuerung, (Bildregel)
 - Bildregel, 188

- StrComp-Funktion, 271
- StructureBlock-Objekt (SiVArc), 248
- SubModuleBlock-Objekt (SiVArc), 246
- SymbolComment-Eigenschaft (SiVArc), 255
- SymbolicName-Eigenschaft (SiVArc), 256
- Symboltabelle, 97
- Syntax
 - SiVArc-Ausdruck, 109
- Systemfunktion, 141
 - Parameter, 52
- Systemfunktionen, 277
 - Bildbausteine, 141

T

- TagNaming-Objekt (SiVArc), 249
- Textliste
 - Textlistenregel, 30
- Textlisten, 162, 177
- Textlisteneinträge, 211
 - Projektierung, 97
- Textlistenregel
 - Bedingung, 30
 - HMI-Objekt, 189
 - Kommentar, 30
 - Kopiervorlage einer Textliste, 30
 - Name, 30
 - Programmbaustein, 30
 - Textliste, 30
- Textlistenregeln, 172
- Textquellen, 180
- Tiefenstaffelung
 - generierte Objekte, 74
- Title-Eigenschaft (SiVArc), 257
- TrailNum-Funktion, 271
- Trim-Funktion, 272
- Type-Eigenschaft (SiVArc), 257

U

- Überlaufbilder
 - Bitmaske, 77
 - mit Bildobjekten, 76
- UBound-Funktion, 272
- Umfang
 - Variablengenerierung, 182
- Unterstützt
 - Geräte, 79

V

- Value-Eigenschaft (SiVArc), 258
- Variablengenerierung, 174
 - Definition, 173
 - Umfang, 182
- Variablengruppe
 - Variablenregel, 29
- Variablenregel, 180
 - Bedingung, 29
 - Kommentar, 29
 - Name, 29
 - Variablengruppe, 29
 - Variablentabelle, 29
- Variablenregeln, 29, 127, 171
 - Index, 29
- Variablentabelle
 - Variablenregel, 29
- Variablentabellen, 162
- Verschaltungen, 174
- Version-Eigenschaft (SiVArc), 258
- Verwendung
 - SiVArc, 13
- Visualisierung
 - Generierung, 49, 94

