

SIMATIC/SIMOTION Projektgenerator

Applikationshandbuch

Vorwort

Applikationsbeschreibung

1

System- und
Fehlermeldungen

2

Tipps und Hilfestellung

3




Kontakt

A

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.

Vorwort

Allgemeine Hinweise

Hinweis

Die Standardapplikationen sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Die Standardapplikationen stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Diese Standardapplikationen entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieser Standardapplikationen erkennen Sie an, dass wir über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden können. Wir behalten uns das Recht vor, Änderungen an diesen Standardapplikationen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesen Standardapplikationen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Gewährleistung, Haftung und Support

Im Falle kostenloser Überlassung der Applikation gilt folgendes:

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.

Andere und alle sonstigen Rechte und Ansprüche gegen die Siemens AG, gleich aus welchem Rechtsgrund, sind ausgeschlossen. Insbesondere Ansprüche gegen die Siemens AG auf Schadenersatz, insbesondere wegen Produktionsausfalls, Nutzungsausfalls, entgangenen Gewinns, direkter, indirekter oder Folgeschäden sind ausgeschlossen.

Dies gilt nicht, soweit zwingend gehaftet wird, z. B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, grober Fahrlässigkeit von Vorgesetzten und leitenden Angestellten der Siemens AG oder in Fällen des arglistigen Verschweigens von Mängeln.

Diese Haftungsbeschränkung findet auch Anwendung zu Gunsten von Subunternehmern, Zulieferern, Beauftragten, Vorgesetzten, leitenden Angestellten und Angestellten der Siemens AG.

Auf diese Vereinbarung findet für Kunden mit Firmensitz in Deutschland das deutsche Recht Anwendung für Kunden mit Firmensitz außerhalb Deutschlands findet das schweizerische Recht Anwendung. Die Anwendbarkeit des Übereinkommens der Vereinten Nationen über Verträge über den internationalen Warenkauf vom 11.04.1980 (CISG) ist ausgeschlossen.

Im Falle entgeltlicher Überlassung der Applikation gilt die für das jeweilige Geschäft zutreffende Alternative:

- Alternative 1: (Internes Geschäft)

Es gelten, sofern nicht unten etwas Abweichendes geregelt ist, die "Bedingungen für Lieferungen und Leistungen im Siemens-internen Geschäft" in der jeweils zum Zeitpunkt der Überlassung gültigen Fassung.

- Alternative 2: (Inlandsgeschäft der Siemens AG)

Es gelten, sofern nicht unten etwas Abweichendes geregelt ist, die "Allgemeine Bedingungen zur Überlassung von Software für Automatisierungs- und Antriebstechnik an Lizenznehmer mit Sitz in Deutschland" in der jeweils zum Zeitpunkt der Überlassung gültigen Fassung.

- Alternative 3: (Direktexportgeschäft der Siemens AG)

Es gelten, sofern nicht unten etwas Abweichendes geregelt ist, die "Allgemeine Bedingungen zur Überlassung von Softwareprodukten für Automation and Drives an Lizenznehmer mit Sitz außerhalb Deutschlands" in der jeweils zum Zeitpunkt der Überlassung gültigen Fassung.

Die Weitergabe oder Vervielfältigung dieser Applikationsbeispiele oder Auszüge daraus in unbearbeiteter Form sind nicht gestattet, soweit nicht ausdrücklich von Siemens Industry Sector gestattet.

Hinweis auf Exportkennzeichen

AL: N

ECCN: N

Über dieses Dokument

Ziel

Dieses Dokument beschreibt die Anwendung des Projektgenerators für SIMATIC/SIMOTION, mit dem Sie schnell und komfortabel ein SIMOTION SCOUT-/SIMATIC STEP 7-Projekt erstellen und Bibliotheken und Module in einem existierenden Projekt aktualisieren können.
Kenntnisse beim Umgang mit STEP 7 und SIMOTION SCOUT werden vorausgesetzt.

Hinweis

Diese Dokumentation erhebt nicht den Anspruch, alle Gerätedetails oder Gerätevarianten zu erfassen oder jeden denkbaren Fall des Betriebs oder der Anwendung zu berücksichtigen.

Sollten Sie weitere Informationen benötigen oder sollten spezielle Probleme auftreten, die für das Anwendungsgebiet nicht ausführlich genug behandelt werden, wenden Sie sich bitte an die örtliche Siemens-Niederlassung.

Zielgruppe

Das vorliegende Dokument wendet sich an Programmierer und Inbetriebnehmer.

Inhaltsverzeichnis

	Vorwort	3
1	Applikationsbeschreibung	7
1.1	Allgemeines	7
1.1.1	Aufgabe des Projektgenerators	7
1.1.2	Lieferumfang	10
1.1.3	Voraussetzungen	14
1.2	Bedienen des Projektgenerators	15
1.2.1	Vorbereitungen	15
1.2.2	Starten des Projektgenerators	15
1.2.3	Darstellung des Projektierungsablaufs	17
1.2.4	Beispielprojekt anlegen	20
1.2.4.1	Voraussetzungen	20
1.2.4.2	Ein neues Projekt mit SIMATIC Geräten anlegen	20
1.2.4.3	Ein bestehendes Projekt mit SIMOTION Geräten anpassen/erweitern	28
1.2.4.4	Übertragen von SIMOTION IT Seiten	36
1.3	Anwenderspezifische Erweiterungen	40
1.3.1	Geräte und Standardmodule hinzufügen	40
1.3.1.1	Projektgenerator erweitern	40
1.3.1.2	Geräte hinzufügen	40
1.3.1.3	Standardmodule hinzufügen	42
1.3.2	Technologieobjekte hinzufügen (nur SIMOTON)	43
1.3.3	Anwenderspezifische Module erstellen und hinzufügen	44
1.4	Architektur des Projektgenerators	46
1.4.1	Allgemeines	46
1.4.2	Struktur des Projektgenerators	46
1.4.3	Verwendung des Projektgenerators im Silent Mode	48
2	System- und Fehlermeldungen	51
2.1	Allgemeines	51
2.2	Systeminformationen	51
2.3	Warnungen	56
2.4	Fehlermeldungen	57
3	Tipps und Hilfestellung	65
3.1	Sonderzeichen in XML	65
3.2	Kommentare im Quellcode	65
3.3	MessageBox ausgeben	66
3.4	Verwendbare Parameter, Events und Objekte	66
3.5	Textdateien verwenden	67
3.6	Quelle mehrfach verwenden	69

3.7	Oberfläche neu laden.....	71
3.8	Generisch auf Oberflächenelemente zugreifen	72
3.9	Generisch die Oberfläche verändern.....	73
3.10	Performante Oberflächenelemente.....	79
A	Kontakt	81
A.1	Ansprechpartner.....	81
A.2	Internetadressen	82

Applikationsbeschreibung

1.1 Allgemeines

1.1.1 Aufgabe des Projektgenerators

Automatisierte Erstellung eines SIMATIC/SIMOTION Projekts

Mit dem Projektgenerator können Sie einfach und komfortabel ein neues oder ein bestehendes SIMATIC/SIMOTION Projekt auf Basis von Vorlagen erstellen oder aktualisieren.

Funktionssammlung von Standardmodulen und Technologischen Modulen

Zu den Vorlagen gehören vorkonfigurierte bzw. vorprojektierte SIMATIC/SIMOTION Geräte und Module, die alle mit dem Projektgenerator ausgeliefert werden. Diese können mit dem Projektgenerator automatisiert in ein Projekt eingebunden werden.

Die Funktionssammlung ist auch um anwenderspezifische Module erweiterbar.

Konfiguration und Parametrierung über Bedienoberfläche

Der Projektgenerator stellt für die Konfiguration und Parametrierung der Module eine Oberfläche zur Verfügung, die modulspezifische Anpassungen ermöglicht.

Die Module können erweitert werden, ohne dass der Quellcode des Projektgenerators geändert werden muss.

Standardisierte Prozesshierarchie gemäß ISA-88

Der Projektgenerator nutzt international etablierte Industriestandards wie IEC, XML und ISA-88. Durch den Einsatz der standardisierten Hierarchien und Schnittstellen auf Basis internationaler Industriestandards können die einzelnen Module des Projektgenerators zum Datenaustausch innerhalb einer Maschine sowie auch zwischen den Maschinen in einer Linie integriert werden. Die Module können auch mit Komponenten anderer Hersteller kombiniert und in bestehende Infrastrukturen eingebunden werden.

Der Projektgenerator eignet sich für die Verwendung in den vier unteren Ebenen des Hierarchiemodells der Norm ISA-88:

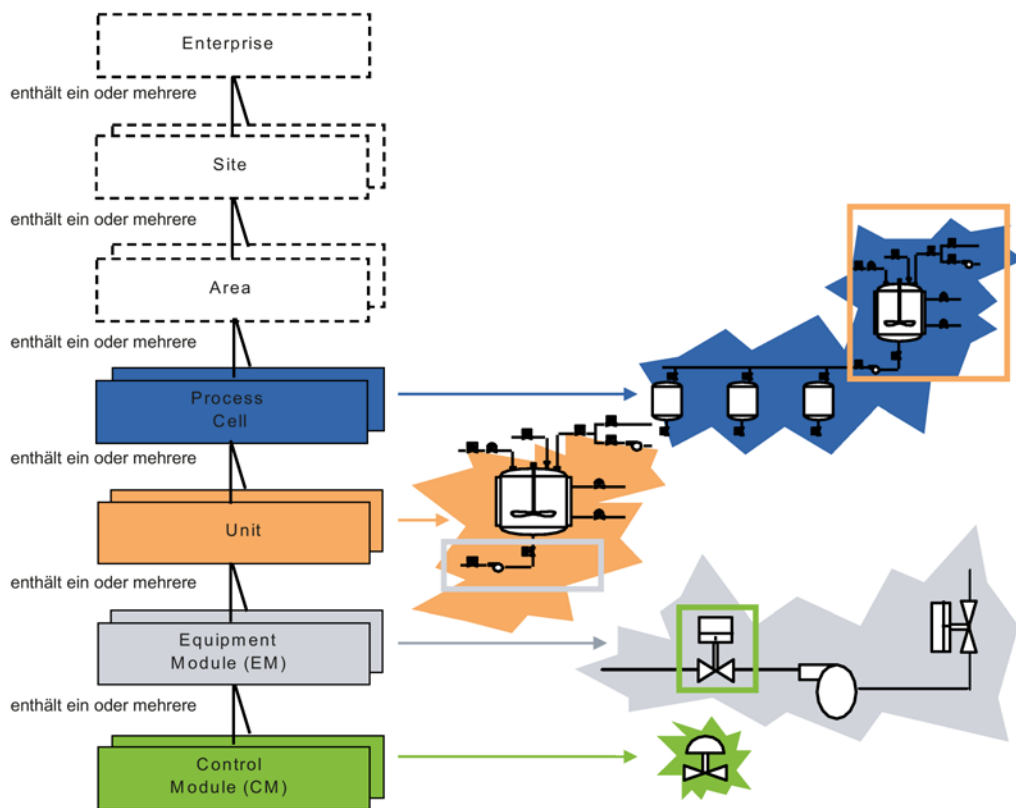


Bild 1-1 Standardisierte Prozesshierarchie gemäß ISA-88

Die Equipment Modules (EM) und Control Modules (CM) enthalten Hard- und Softwarekomponenten, die nur ein einziges Mal geschrieben werden müssen und dann beliebig oft in der im Bild dargestellten Hierarchien verwendet werden können.

Eine Unit (Maschine) kann nach ISA-88 aus einem Equipment Module, z. B. dem Maschinenmodul *Abwicklung* oder mehreren gleich- oder verschiedenartigen Equipment Modules bestehen. Ein Equipment Module kann wiederum mehrere Equipment Modules und/oder Control Modules enthalten. Control Modules sind Module mit Prozessanbindung, wie z. B. eine Achse oder der Temperaturregler. Ein Control Module kann wiederum mehrere Control Modules enthalten.

Der Projektgenerator unterstützt heute eine Ebene Equipment Modules und darunter eine Ebene Control Modules.

Funktionen des Projektgenerators im Überblick

Mit dem Projektgenerator können Sie folgende Funktionen ausführen:

Tabelle 1- 1 Funktionen des Projektgenerators

Funktion	Projektgenerator für SIMATIC	Projektgenerator für SIMOTION
Anlegen eines neuen Projekts bzw. Bearbeiten und Öffnen eines vorhandenen Projekts	✓	✓
Anlegen eines neuen Geräts bzw. Öffnen und Bearbeiten eines vorhandenen Geräts	✓	✓
Automatische Konfiguration der Module anhand der ausgewählten Hardware	✓	✓
Kopieren von Dateien/Verzeichnissen im Dateisystem	✓	✓
XML-Schnittstelle für anwenderspezifische Erweiterungen	✓	✓
<ul style="list-style-type: none"> • Generierung der Oberflächenelemente • generische Funktionsaufrufe 	<ul style="list-style-type: none"> ✓ ✓ 	<ul style="list-style-type: none"> ✓ ✓
Zuweisung von Programmen ins Ablaufsystem		✓
FTP-Transport von Dateien zum SIMOTION Gerät (z. B. SIMOTION IT Seiten)		✓
Import von Technologieobjekten (TO)		✓
Verschalten von Technologieobjekten (Gleichlaufverschaltung)		✓
Schreiben von Konfigurationsdaten und Systemvariablen in Technologieobjekte		✓
Import von Bibliotheken:	✓	✓
<ul style="list-style-type: none"> • Code ändern • Versionskontrolle und Versionsupdate • Setzen und wiederherstellen von Konstanten • Erzeugen und wiederherstellen von Userbereichen 	<ul style="list-style-type: none"> ✓ ✓ 	<ul style="list-style-type: none"> ✓ ✓ ✓ ✓
Import von Units		✓
<ul style="list-style-type: none"> • Code ändern • Versionskontrolle und Versionsupdate • Setzen und wiederherstellen von Konstanten • Erzeugen und wiederherstellen von Userbereichen 		<ul style="list-style-type: none"> ✓ ✓ ✓ ✓
Importieren von Quellen	✓	
Importieren/Hinzufügen von Netzwerken in Organisationsbausteinen	✓	
Hinzufügen von Variablen in den Deklarationsbereich eines Organisationsbausteins	✓	

1.1.2 Lieferumfang

Der Projektgenerator ist über einen Link auf dem Speichermedium **Utilities & Applications** abrufbar. **Utilities & Applications** ist Teil des Engineeringssystem SIMOTION SCOUT.

Verzeichnisstruktur des Projektgenerators

Der Projektgenerator wird in der Version 1.2 mit folgender Verzeichnisstruktur ausgeliefert:

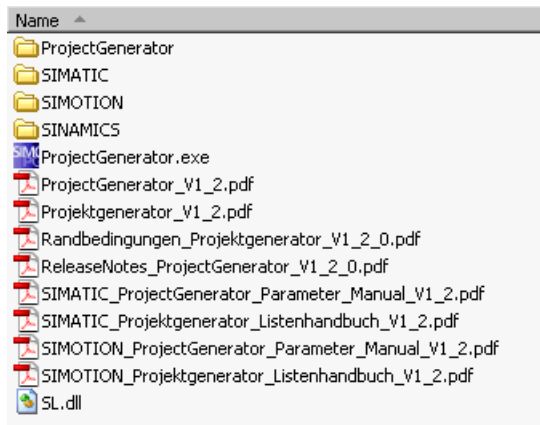


Bild 1-2 Struktur des Projektgenerators

Auf oberster Ebene finden Sie

- die Datei **ProjectGenerator.exe**, mit der Sie den **Projektgenerator** starten.
- die **Dokumentation** für den Projektgenerator in Form von pdf-Dokumenten jeweils in den Sprachen Deutsch und Englisch.

Verzeichnis ProjectGenerator

Hier finden Sie Dateien, die für den Betrieb des Projektgenerators notwendig sind. Die Dateien sind systeminterne Dateien und nicht für die Änderung durch den Anwender vorgesehen.

Verzeichnisse SIMATIC, SIMOTION und SINAMICS

In den Verzeichnissen SIMATIC und SIMOTION finden Sie Basisdaten (Verzeichnisse und Dateien), die im Zusammenhang mit der Generierung eines SIMATIC bzw. SIMOTION Projekts gebraucht werden.

Im Verzeichnis SINAMICS finden Sie importierbare SINAMICS Geräte.

Die Verzeichnisse SIMATIC, SIMOTION und SINAMICS beinhalten die folgenden Unterverzeichnisse:

- **Devices**

Das Verzeichnis **Devices** enthält die Dateien der SIMATIC, SIMOTION und SINAMICS Geräte.

Es ist auch möglich, in dieses Verzeichnis eigene vorprojektierte Geräte (Devices) mit aufzunehmen.

Tabelle 1- 2 Beispiele Devices

Ordner	Unterordner	Geräte
SIMATIC	Devices	S7300 SIMATIC S7315 SIMATIC S7317 SIMATIC S7319 ...
		S7400 SIMATIC S7414 SIMATIC S7416 ...
SIMOTION	Devices	V4.2: SIMOTION C240PN SIMOTION D410PN SIMOTION D425 SIMOTION D435 ...
		V4.3: SIMOTION C240PN SIMOTION D410-2PN SIMOTION D425-2 SIMOTION D435-2 ...
SINAMICS	Devices	1_DriveObjects 2_DriveObjects 3_DriveObjects ... In der zusätzlich anzulegenden Datei <i>IOConfig.XML</i> können die IO Adressen der Drive Objekte für HW Konfig hinterlegt werden.

- **EquipmentModules**

Das Verzeichnis **EquipmentModules** enthält für SIMATIC bzw. für SIMOTION versionsabhängig die vorgefertigten Standardmodule des Projektgenerators.

Dies sind z. B.:

Tabelle 1-3 Beispiele SIMATIC und SIMOTION EquipmentModules

Ordner	Unterordner	EquipmentModules
SIMATIC	EquipmentModules	Communication_LCom Der in der Bibliothek <i>LCom</i> zur Verfügung gestellte Funktionsbaustein FBLComMachineCom stellt die Funktion eines datensatzorientierten Transportprotokolls LCom-Protokoll zur Verfügung. Für die Übertragung der Daten über Ethernet wird das Transportprotokoll TCP genutzt.
		OMACV30_LPML Die Software-Bibliothek <i>LPMLV30</i> stellt eine komfortable Basis zum Aufbau einer OMAC-konformen Betriebsartenverwaltung und Datenschnittstelle für ein SIMATIC/SIMOTION Gerät zur Verfügung.
		Weihenstephan Die Bibliothek <i>LWeihenstephan</i> stellt die Funktionalität des Weihenstephaner Standards für Steuerungen vom Typ SIMATIC S7 zur Verfügung.
		...
SIMOTION	EquipmentModules	Communication_LCom Der in der Bibliothek <i>LCom</i> zur Verfügung gestellte Funktionsbaustein FBLComMachineCom stellt die Funktion eines datensatzorientierten Transportprotokolls LCom-Protokoll zur Verfügung. Für die Übertragung der Daten über Ethernet wird das Transportprotokoll TCP genutzt.
		OMACV30_LPML Die Software-Bibliothek <i>LPMLV30</i> stellt eine komfortable Basis zum Aufbau einer OMAC-konformen Betriebsartenverwaltung und Datenschnittstelle für ein SIMATIC/SIMOTION Gerät zur Verfügung.
		StartupCheck Die Hochlaufprüfung stellt Funktionen zur Prüfung von Peripheriebaugruppen und Technologieobjekten vom Typ Achse und Externer Geber beim Hochlauf der Maschine zur Verfügung. So werden dem Anwender bei fehlerhaftem Hochlauf Detailinformationen über die Fehlerquelle bereit gestellt. Bei erfolgreichem Hochlauf wird ein Signal bzw. eine Meldung ausgegeben.
		Messagehandling_LMsgHdl Die Bibliothek <i>LMsgHdl</i> stellt grundlegende Funktionen zur Anzeige und Verwaltung aller Arten von Meldungen des SIMOTION Systems zur Verfügung (z. B. Störungen und Warnungen).
		...

- **Projects (nur SIMOTION)**

Hier finden Sie das Ursprungsprojekt, auf das der Projektgenerator bei der Auswahl **Projekt Neu** zurückgreift. Es beinhaltet ein bereits vernetztes Programmiergerät für ein späteres Onlinegehen.

Die Ethernet-Schnittstelle ist bereits mit der Adresse 169.254.11.99 vorbelegt und muss lediglich der verwendeten Netzwerkkarte zugeordnet werden. Dieser Vorgang kann nicht durch den Projektgenerator abgedeckt werden, da die Varianz an Netzwerkadaptern auf dem Markt zu groß ist.

Möchten Sie ein kundenspezifisches Basis-Projekt ablegen, müssen Sie darauf achten, das Verzeichnis **\\SIMOTION\Projects\Project_Basis** nicht umzubenennen, sondern lediglich die Projekt-Dateien in diesem Verzeichnis auszutauschen.

Wird eine SIMOTION CPU in ein Projekt eingefügt, wird sie automatisch mit dem Standard-Subnetz **Ethernet(1)** verbunden, falls dieses vorhanden ist. Ist es nicht vorhanden, wird die CPU ohne Vernetzung importiert. Als Schnittstelle wird für die SIMOTION D4x5 Komponenten immer die IE2/NET (X130) Ethernet Schnittstelle vernetzt. Bei SIMOTION D410 PN wird standardmäßig die PN-Schnittstelle X200 vernetzt. Für alle SIMOTION D4xx-2-Komponenten wird PN/IE (X127) verwendet. Jede weitere CPU verbindet sich bei ihrem Import ebenfalls mit diesem Standard-Subnetz.

Falls die CPU auf ein anderes Subnetz verbunden werden soll, kann dies nachträglich über NetPro erfolgen oder direkt in der Daten-Basis im Projektgenerator.

- **Scripts (nur SIMOTION)**

Hier finden Sie versionsabhängige Konfigurations- und Übersichtsskripten, die über den Projektgenerator importiert werden können.

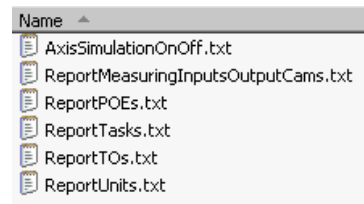


Bild 1-3 Verzeichnis Scripts

- **TechnologyObjects (nur SIMOTION)**

Das Verzeichnis **TechnologyObjects** enthält versionsabhängige XML-Exporte der Technologieobjekte vom Typ Achse, z. B.:

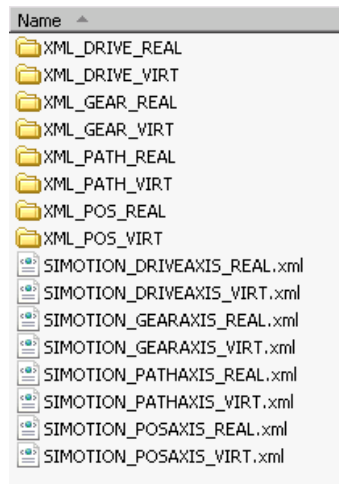


Bild 1-4 Verzeichnis TechnologyObjects

Unabhängig von den hier gelisteten Technologieobjekten kann jedes Modul seine eigenen Technologieobjekte im entsprechenden Unterverzeichnis speichern und adressieren.

- **TechnologyPackages (nur SIMOTION)**

Das Verzeichnis **TechnologyPackages** enthält versionsabhängige systeminterne Dateien mit den Kennungen der Technologiepakete und Geräte.

- **TelegramConfig (nur SINAMICS)**

Das Verzeichnis **TelegramConfig** enthält versionsabhängige systeminterne Dateien mit Telegrammprojektion und MLFBs der SINAMICS Control Units.

1.1.3 Voraussetzungen

Der Einsatz des Projektgenerators setzt eine Installation von SIMOTION SCOUT ab V4.2 und STEP 7 V5.5 voraus.

Falls Sie nur die SIMATIC-Funktionen nutzen möchten, muss SIMOTION SCOUT nicht installiert sein.

1.2 Bedienen des Projektgenerators

1.2.1 Vorbereitungen

Hinweis

Vor dem Starten des Projektgenerators müssen alle STEP 7 bzw. SIMOTION SCOUT Anwendungen geschlossen werden.

Hinweis

Um einen schnellen und stabilen Durchlauf sicherzustellen, muss der Projektgenerator von einem lokalen Laufwerk gestartet werden.

1.2.2 Starten des Projektgenerators

Der Projektgenerator kann im **Standard Mode** und im **Silent Mode** gestartet werden.

Standard Mode

Doppelklicken Sie auf die Datei **ProjectGenerator.exe**, wenn Sie den Projektgenerator im **Standard Mode** starten möchten.

Die Projektgenerator-Oberfläche wird geladen und führt Sie schrittweise vom Anlegen des Projekts bis hin zur Generierung.

Weitere Informationen sowie die dazu notwendigen einzelnen Schritte finden Sie im Kapitel Beispielprojekt anlegen (Seite 20).

Silent Mode

Sie können den Projektgenerator im **Silent Mode** ohne Dialog mit dem Anwender ausführen. Dadurch ist es möglich, den Projektgenerator in eine eigene Anwendung zu integrieren.

Sie aktivieren den **Silent Mode**, indem Sie der Datei *ProjectGenerator.exe* beim Start den Pfad zu einer gültigen externen XML-Datei des Projektgenerators übergeben. Die Befehle der Datei werden dann direkt abgearbeitet.

Hier sehen Sie einen Beispielaufruf für eine Batch-Datei mit der Endung *.xml* zum Starten des Projektgenerators im **Silent Mode**:

Tabelle 1-4 Codebeispiel Projektgenerator für SIMOTION

```
C:\SIMOTION\ProjectGenerator.exe C:\SIMOTION\PGEN_DATA_BASE.xml
if errorlevel 1 goto error1
if errorlevel 0 goto no_error
goto End

:error1
Echo Error
goto End

:no_error
Echo No Error
goto End

:End
```

Weitere Informationen zum **Silent Mode** siehe Kapitel Verwendung des Projektgenerators im Silent Mode (Seite 48).

1.2.3 Darstellung des Projektierungsablaufs

Start des Projektgenerators

Wenn Sie den Projektgenerator gestartet haben, wird nach Prüfung der Software-Voraussetzungen und der Bestätigung der Meldung Haftungsausschluss die Oberfläche des Projektgenerators geladen. Dieser Bereich ist vom Anwender **nicht beeinflussbar** (siehe Bild *Projektierungsablauf 1*):

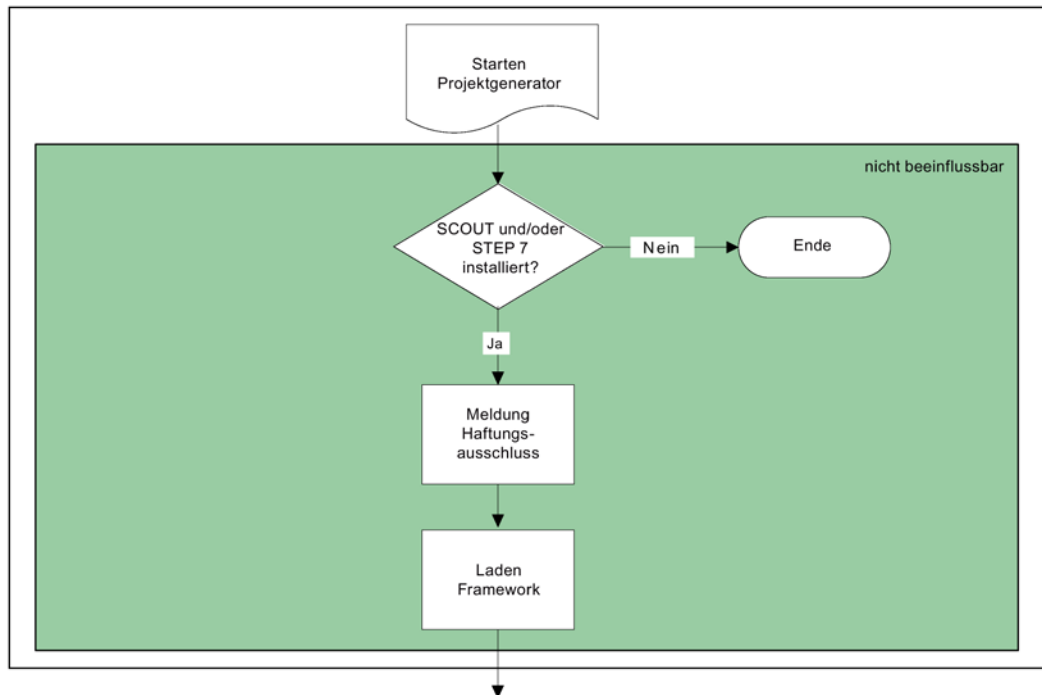


Bild 1-5 Projektierungsablauf 1

Neues Projekt anlegen oder bestehendes Projekt aktualisieren

Im nächsten Schritt legen Sie fest, ob Sie ein neues Projekt erstellen möchten oder ein vorhandenes Projekt laden und bearbeiten möchten.

Dann erfolgt das Anlegen bzw. die Auswahl des SIMATIC bzw. SIMOTION Geräts oder der SIMATIC/SIMOTION Geräte für das Projekt. Dabei wird jedes einzelne Gerät nacheinander konfiguriert.

Die Auswahl und Konfiguration der einzelnen Module stellt den **durch den Anwender beeinflussbaren Bereich dar** (hervorgehobener Bereich in der Mitte des Bildes *Projektierungsablauf 2*). Integriert werden können Standardmodule oder auch anwenderspezifische Module, die frei projiziert werden können (siehe Kapitel *Anwenderspezifische Module erstellen und hinzufügen* (Seite 44)).

Generierung des Projekts und Öffnen von SIMOTION SCOUT

Sind alle Geräte konfiguriert, erfolgt die Generierung des Projekts. Nach erfolgter Generierung kann das soeben angelegte bzw. aktualisierte Projekt geöffnet werden. Im Falle eines SIMOTION Projekts kann der SCOUT direkt aus dem Projektgenerator geöffnet werden. Im Projekt sind nun die Module, welche zuvor ausgewählt und konfiguriert wurden, enthalten.

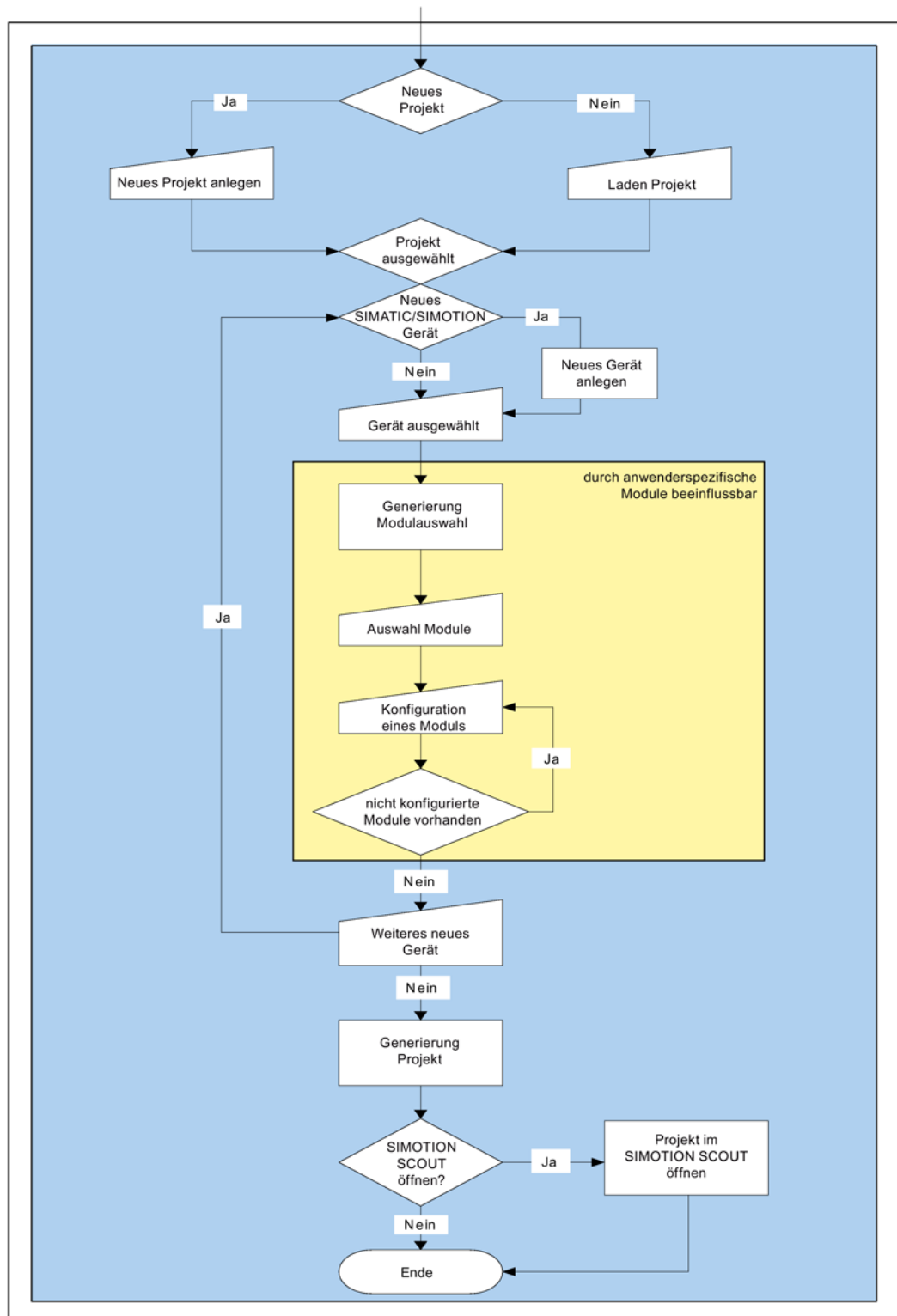


Bild 1-6 Projektierungsablauf 2

1.2.4 Beispielprojekt anlegen

1.2.4.1 Voraussetzungen

Voraussetzungen

- SIMOTION SCOUT und STEP 7 sind geschlossen
- Die Funktionen und Eigenschaften der Standardmodule sind bekannt. Die Dokumentationen der einzelnen Module werden mit in den Dateien des Projektgenerators ausgeliefert.

1.2.4.2 Ein neues Projekt mit SIMATIC Geräten anlegen

In diesem Beispielprojekt wird mit dem Projektgenerator ein neues SIMATIC Gerät mit den Standardmodulen *Ethernet Communication LCom* und *Weihenstephan Standard* angelegt.

So legen Sie ein neues SIMATIC Gerät mit Standardmodulen an

1. Starten Sie den Projektgenerator, bestätigen Sie den Haftungsausschluss und wählen Sie die Option **Create a new project**.

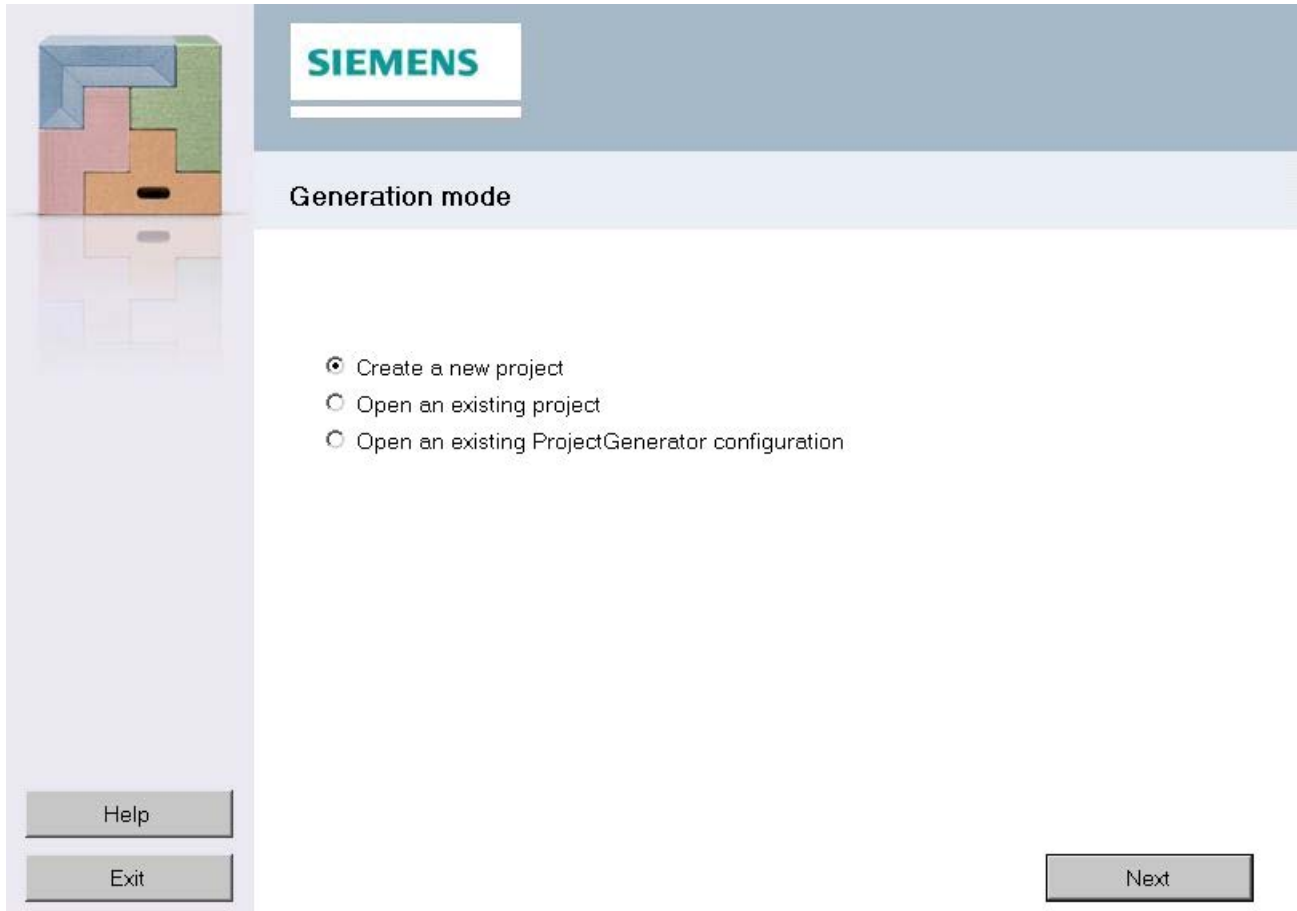


Bild 1-7 Projekt anlegen oder wählen

2. Wenn Sie **Create a new project** ausgewählt haben, geben Sie einen Projektnamen und den Ablageort des Projekts ein (der Pfad ist auch über **Browse** auswählbar) und klicken auf die Schaltfläche **Next**.

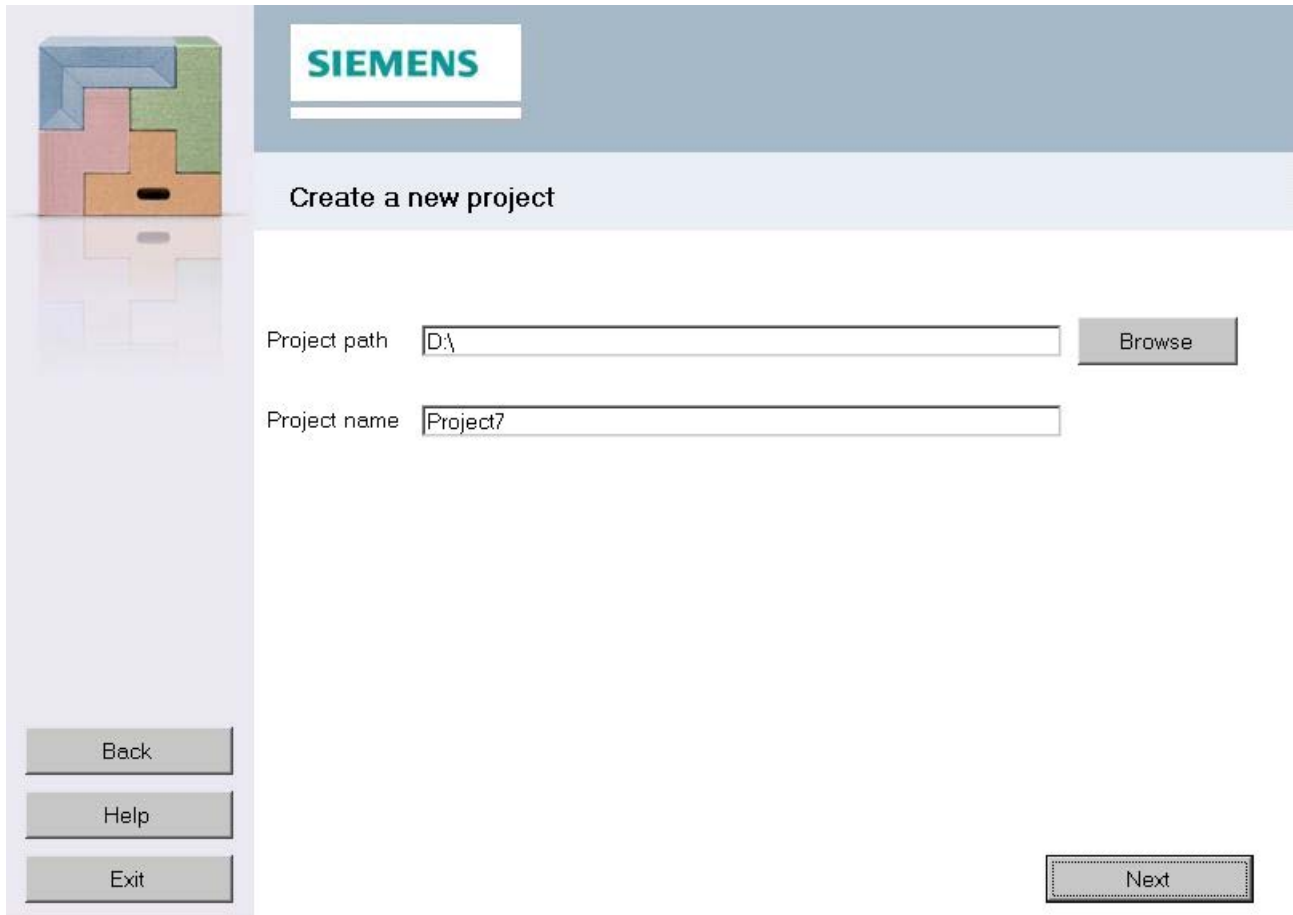


Bild 1-8 Projektname und Ablagepfad des Projekts

Das Fenster *Device selection* wird geöffnet.

3. Wählen Sie im Fenster *Device selection* unter **Select device category**, welches Gerät/welche Geräte in das Projekt integriert werden sollen:

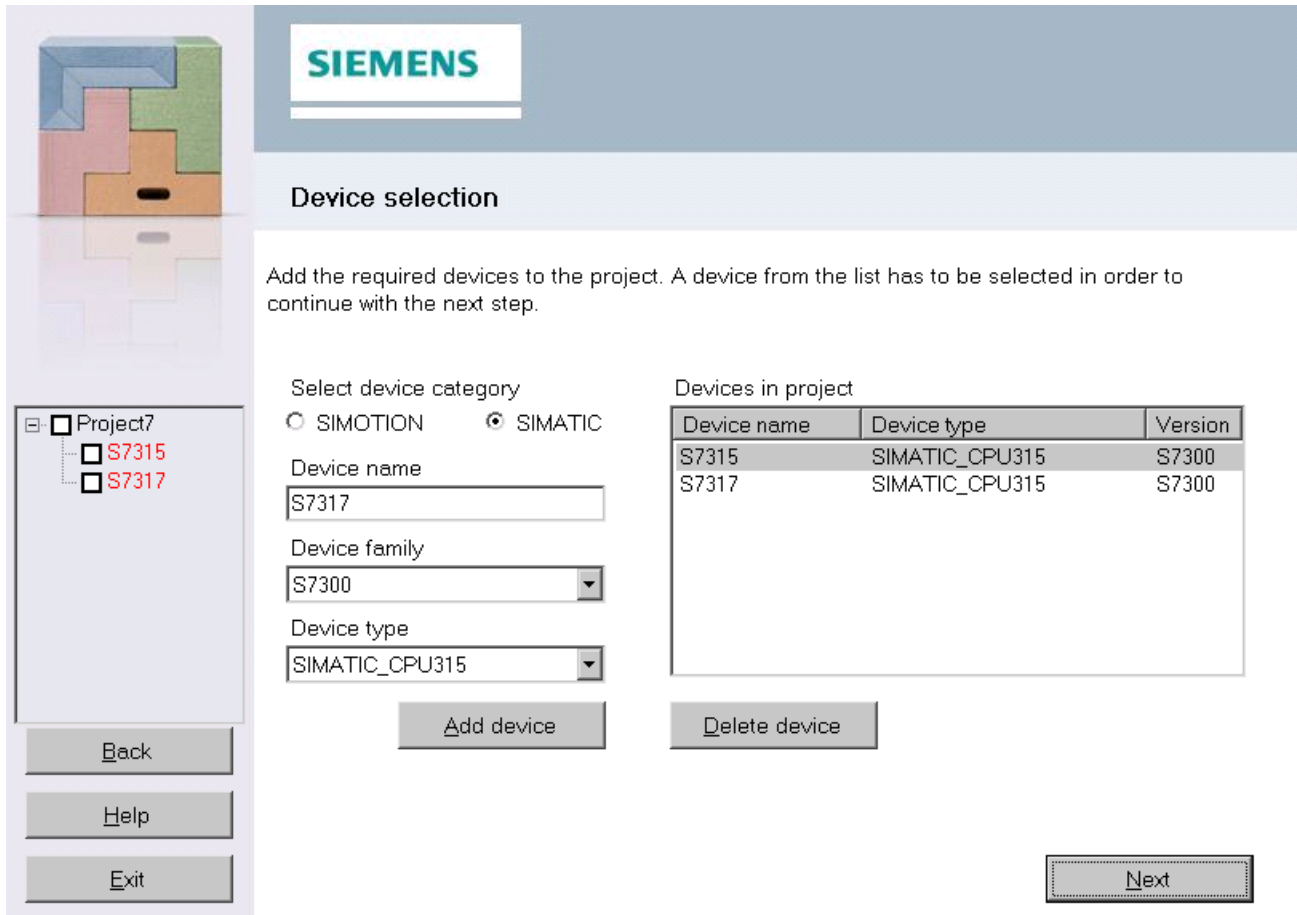


Bild 1-9 Gerät anlegen oder wählen

Abhängig von der Software Installation können Sie entweder nur SIMATIC Geräte oder SIMOTION und SIMATIC Geräte gemischt nacheinander auswählen.

Geben Sie Version bzw. Art des Geräts und den Typnamen des Geräts ein und klicken Sie auf die Schaltfläche **Add device**. Das neue Gerät wird in die Liste **Devices in project** übernommen.

Wenn Sie ein weiteres Gerät anlegen möchten, wiederholen Sie den Vorgang.

4. Markieren Sie in der Liste **Devices in project** das Gerät, das Sie konfigurieren möchten und klicken Sie auf die Schaltfläche **Next**.

Es öffnet sich das Fenster *Equipment module selection*.

5. Wählen Sie im Fenster *Equipment module selection* das oder die Standardmodule, die Sie in das gewählte Gerät integrieren möchten und klicken Sie auf die Schaltfläche **Next**.

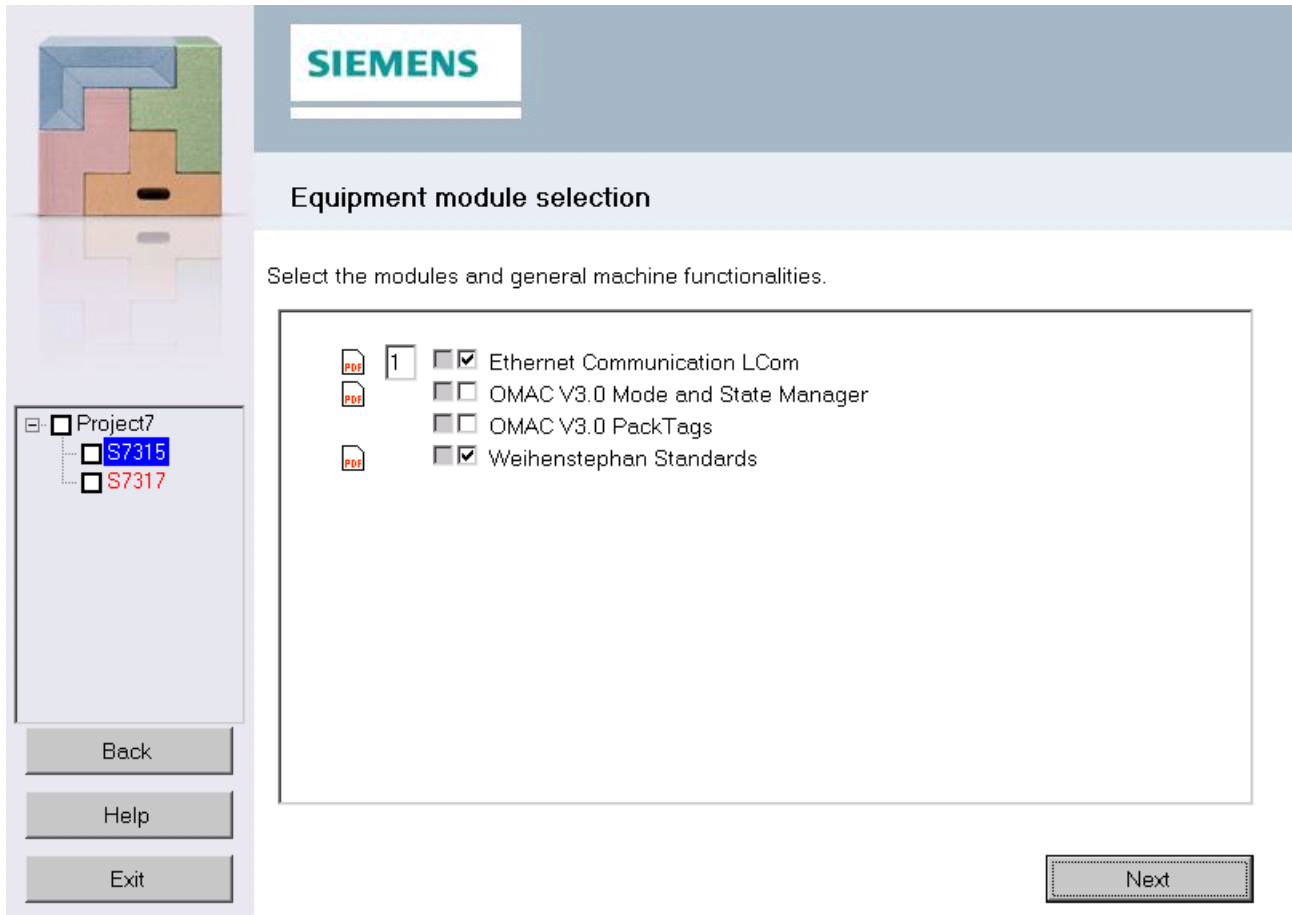


Bild 1-10 Beispiel: SIMATIC Standardmodule wählen

Wenn Sie auf die Schaltfläche **PDF** klicken, wird die Dokumentation zum jeweiligen Standardmodul geöffnet.

Bei einigen Standardmodulen, können Sie links die Anzahl der Module eingeben.

Für unser Beispiel haben wir die Module *Ethernet Communication LCom* und *Weihenstephan Standards* ausgewählt, die in den folgenden Masken nacheinander konfiguriert werden.

6. Konfigurieren Sie im Fenster *Ethernet Communication LCom - Function Block Call* den Aufruf des Funktionsbausteins mit den dazu notwendigen Datenbausteinen und Kommunikationsparametern und klicken Sie auf die Schaltfläche **Next**.

Module 1 of 2

Ethernet Communication LCom - Function Block Call

Fill in the required data and this tool will add a function block call in OB1 to the project.

Add the FB call to the project

New instance DB name	DBFBLCOMMachineCom	DB number	105
Parameter DB name	DBLComParameter	DB number	500
Send DB name	DBLComSend	DB number	501
Receive DB name	DBLComReceive	DB number	502

Communication parameters

SIMATIC is TCP client (active partner) Local port: 3456

IP address (partner): 169 | 254 | 11 | 23 Remote port: 3456

Connection ID: 105 Sender cycle time (ms): 500

Data blocks in device

Symbolic name	DB number	Data type

Next

Bild 1-11 Konfiguration des Standardmoduls *Ethernet Communication LCom*

Machen Sie hier die folgenden Eingaben:

– **Add FB call to the project**

Unter **Add FB call to the project** hinterlegen Sie die für den Aufruf des Funktionsbausteins notwendigen Namen und Nummern für die Datenbausteine (DB): **New instance DB name, Parameter DB name, Send DB name, Receive DB name.**

– **Communication parameters**

Hier legen Sie die Übergabeparameter des Funktionsbausteinaufrufs fest:

Ist das SIMATIC Gerät TCP Client, also der aktive Verbindungspartner, dann aktivieren Sie das Kästchen **SIMATIC is TCP Client**. Geben Sie in diesem Fall die **IP-Adresse** und die **Remote-Portnummer** ein.

Außerdem können Sie den **Local Port** und die **Zykluszeit** verändern.

– **Data blocks in device**

Bereits für das Gerät vorhandene Datenbausteine werden in der Tabelle **Data blocks in device** angezeigt.

- Konfigurieren Sie das nächste Standardmodul *Weihenstephan Standards* und klicken Sie dann auf die Schaltfläche **Next**.

Module 2 of 2

Weihenstephan Standards

Fill in the required data and this tool will generate the DBs and add a function block call in OB1 to the project.

Generate Weihenstephan DBs only

List DB number

Search DB number

Data DB number

Path to the PDA-Config file

Data blocks in device

Symbolic name	DB number	Data type
DBFBLComMachineCom	DB 105	FB 105
DBLComParameter	DB 500	DB 500
DBLComReceive	DB 502	DB 502
DBLComSend	DB 501	DB 501

Bild 1-12 Konfiguration des Weihenstephaner Standards

Sie können hier die folgenden Eingaben machen:

- **List DB Number/Search DB Number/Data DB Number**

Geben Sie die gewünschten Nummern der DBs ein, damit eine passende Schnittstelle für die Weihenstephaner Standards-Funktionalität erzeugt wird.

- **Browse**

Mit der Schaltfläche **Browse** können Sie den Pfad zum PDA-Config-File einstellen/ändern.

Wenn Sie mit der Konfiguration fertig sind und auf **Next** klicken, wird das Fenster *Generate the project* geöffnet.

8. Konfigurieren Sie alle weiteren Geräte nach obigem Muster, indem Sie auf die Schaltfläche **Configure devices** klicken.

Fertig konfigurierte Geräte werden links außen unterhalb des Projektnamens in grüner Farbe mit Häkchen dargestellt, noch nicht konfigurierte Geräte in roter Farbe, in Arbeit befindliche Geräte orangefarben.

9. Wenn Sie kein weiteres Gerät mehr konfigurieren (**Configure devices**) und das Projekt abschließen möchten, haben Sie die folgenden Möglichkeiten.

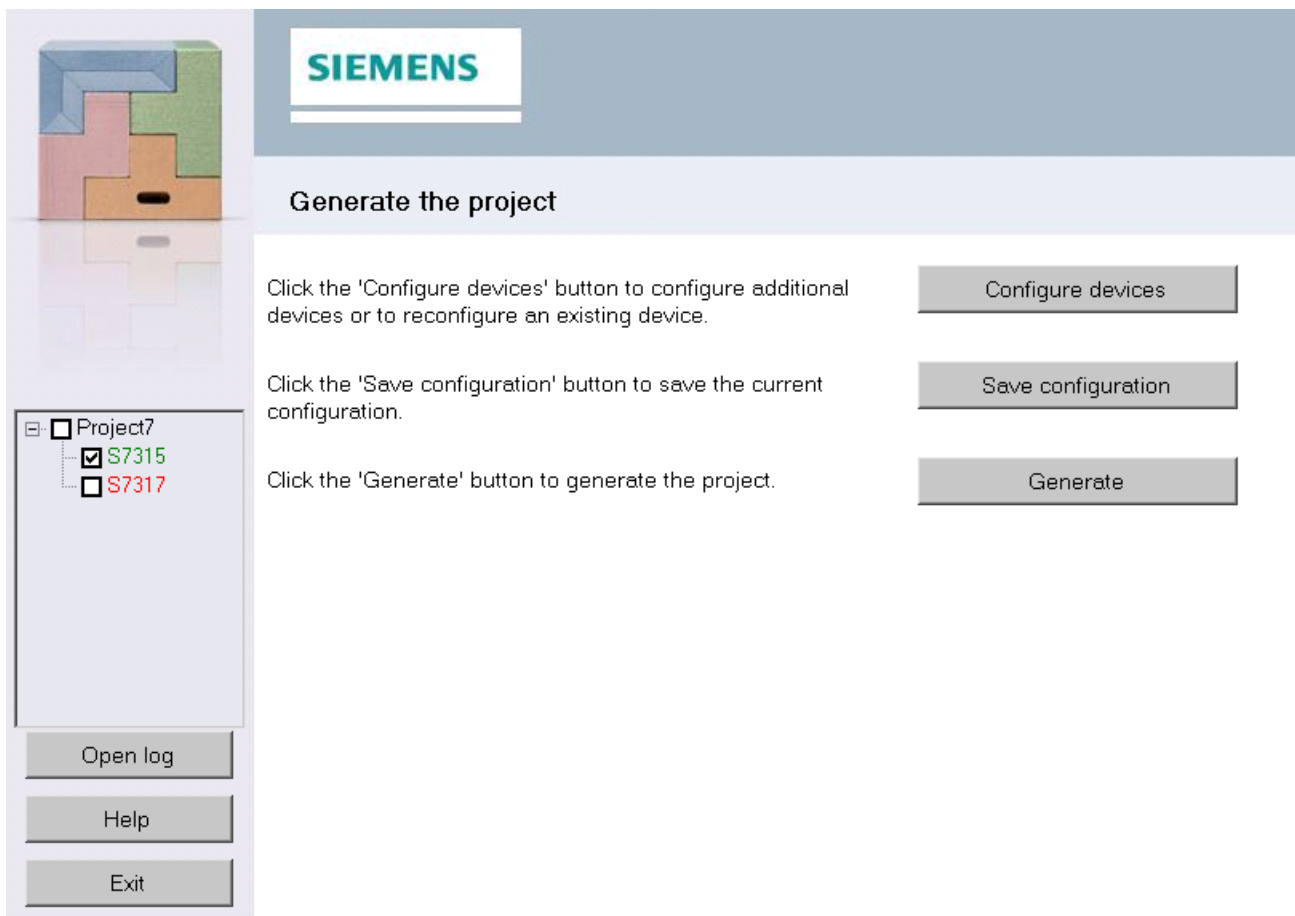


Bild 1-13 Projekt generieren

- Möchten Sie die Konfigurationen der Geräte speichern, aber erst zu einem späteren Zeitpunkt das Projekt generieren, dann klicken Sie auf die Schaltfläche **Save Configuration** und geben im Explorer den Pfad zum Speicherort ein.
- Möchten Sie kein weiteres Gerät mehr konfigurieren und das Projekt generieren, dann klicken Sie auf **Generate**.

Das Projekt wird generiert. Die Dauer richtet sich nach der Art der Konfiguration und wird mit Hilfe eines Fortschrittbalkens angezeigt.

Ist das Projekt fertig generiert, erscheint im Fenster die Meldung **Generation finished**.

10. Klicken Sie auf die Schaltfläche **Exit**, um den Projektgenerator zu schließen.

1.2.4.3 Ein bestehendes Projekt mit SIMOTION Geräten anpassen/erweitern

Beispielprojekt

In diesem Beispiel wird mit dem Projektgenerator ein bestehendes SIMOTION Projekt geöffnet und um ein neues Gerät mit den Standardmodulen *Message Handling* und *Startup Check* angelegt.

So öffnen Sie ein bestehendes SIMOTION Projekt und erweitern oder verändern es

1. Starten Sie den Projektgenerator, bestätigen Sie den Haftungsausschluss und wählen Sie die Option **Open an existing project**.

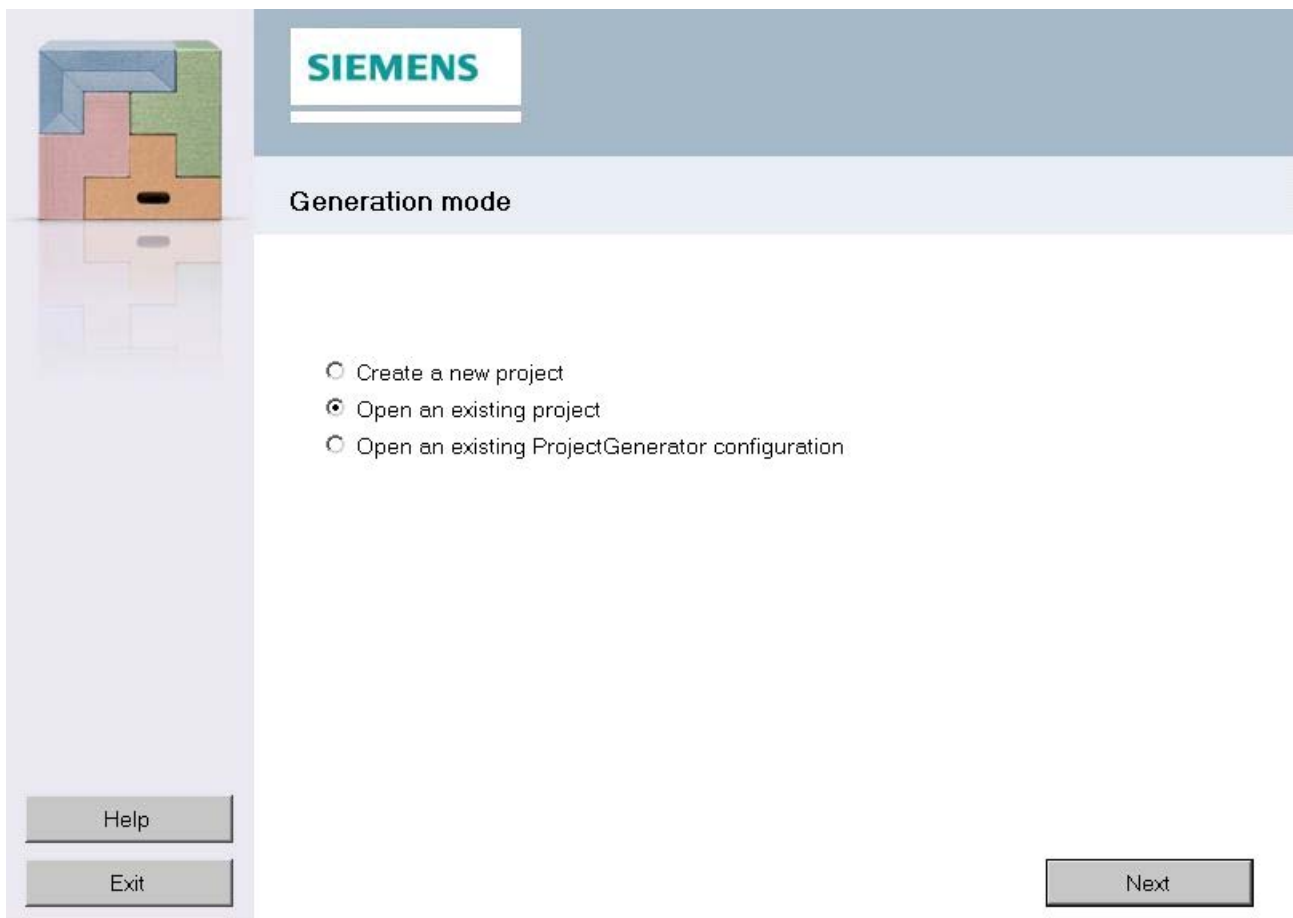


Bild 1-14 Projekt anlegen oder wählen

Das Fenster *Open an existing project* wird geöffnet.

2. Wählen Sie das Projekt aus, das Sie öffnen möchten, und klicken auf die Schaltfläche **Next**.

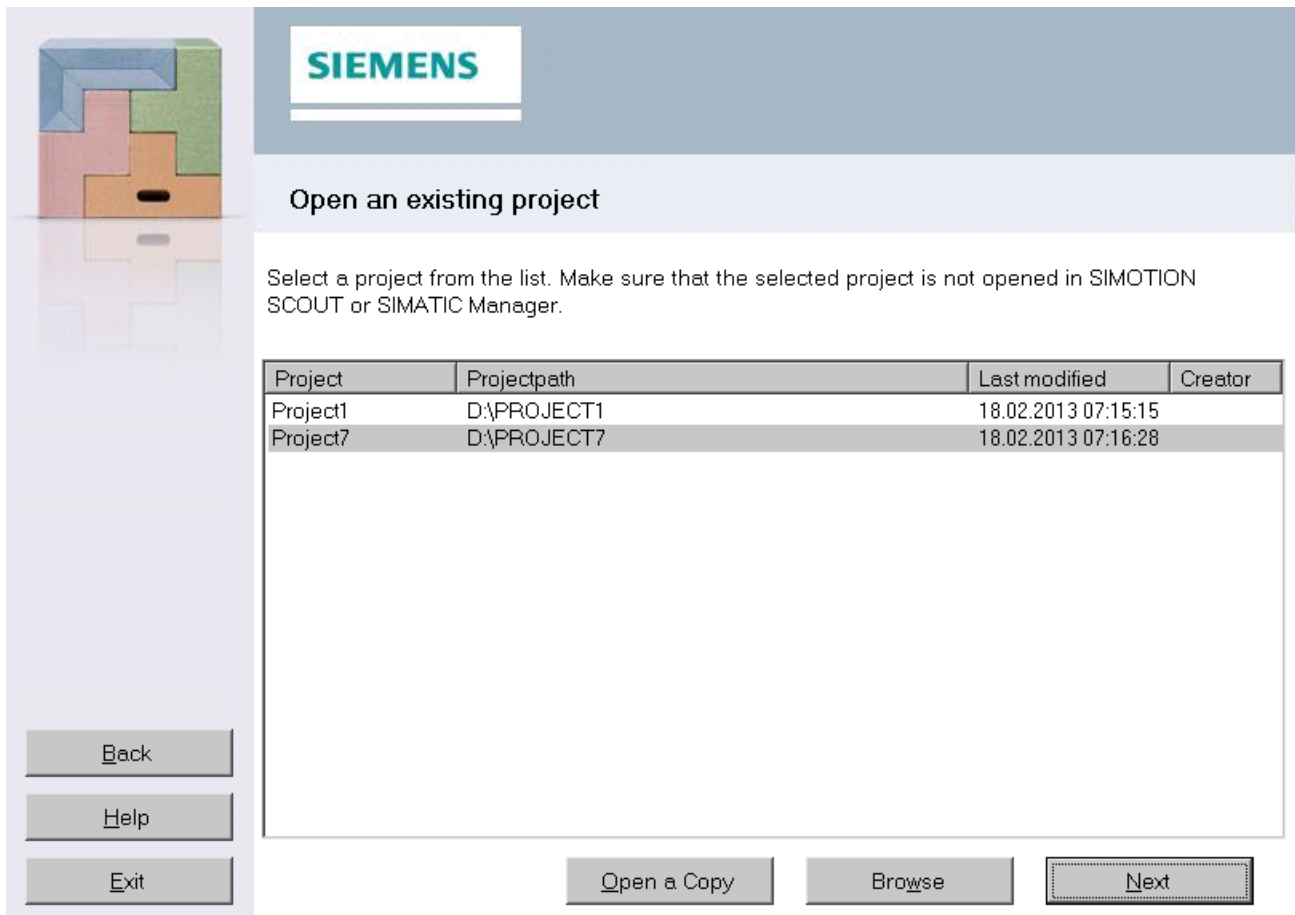


Bild 1-15 Projekt wählen

Wird das gewünschte Projekt nicht angezeigt, klicken Sie auf die Schaltfläche **Browse** und wählen über den Windows-Explorer den Pfad zum gewünschten Projekt.

Das ausgewählte Projekt wird in die Tabellenansicht übernommen. Markieren Sie das Projekt in der Tabelle und klicken Sie dann auf die Schaltfläche **Next**.

Das Fenster *Device selection* wird mit den für das Projekt angelegten Geräten geöffnet.

3. Markieren Sie im Fenster *Device selection* unter **Devices in project** das SIMOTION Gerät, welches Sie bearbeiten wollen, oder legen Sie ein neues Gerät an.

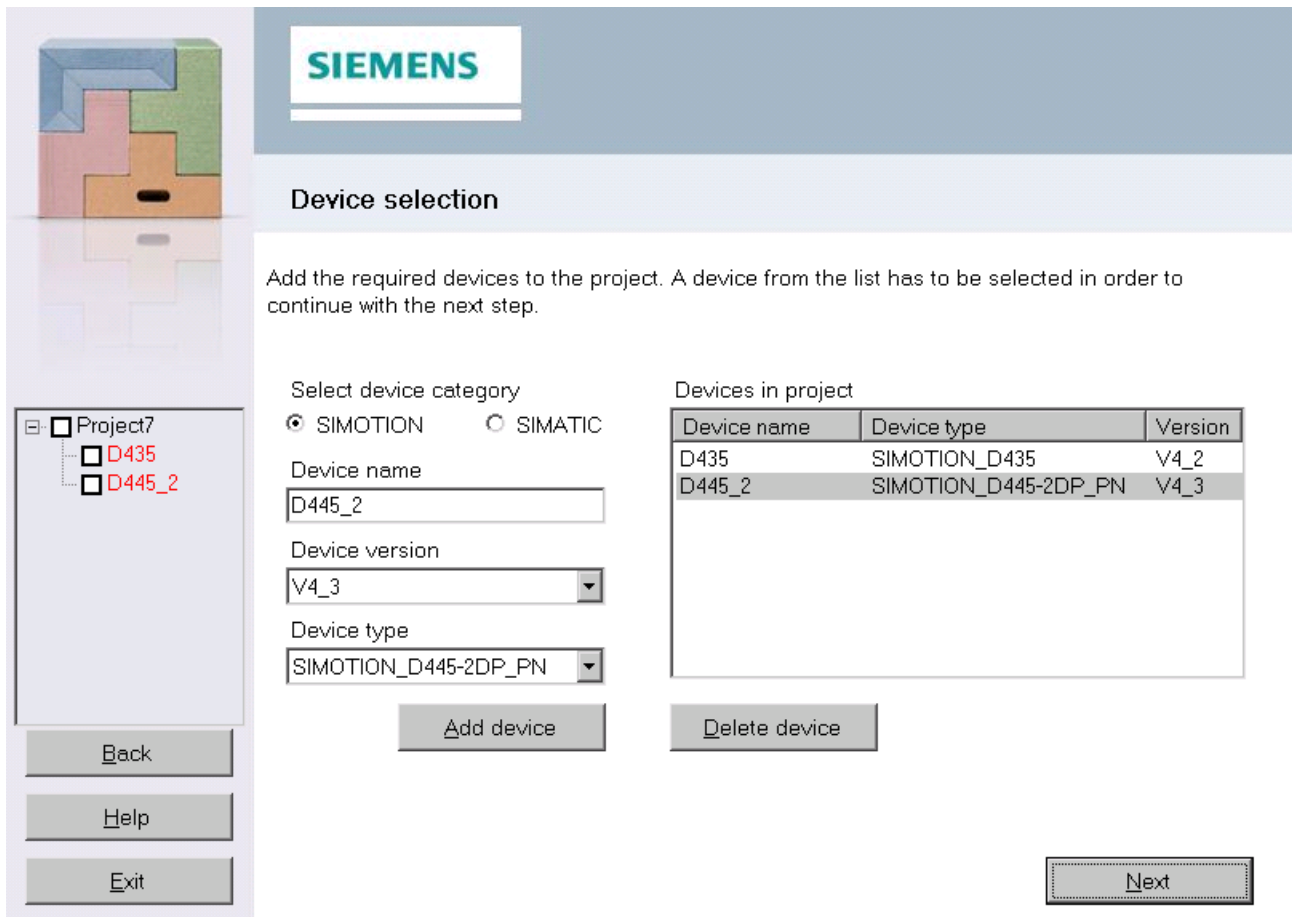


Bild 1-16 SIMOTION Gerät anlegen oder wählen

Sie haben die folgenden Möglichkeiten:

- Im linken Teil des Fensters können Sie für das Projekt weitere Geräte anlegen. Geben Sie dazu die Version bzw. die Art des Geräts und den Typnamen des Geräts ein und klicken Sie auf die Schaltfläche **Add device**. Das neue Gerät wird in die Liste **Devices in project** übernommen.
 - Sie können ein neu angelegtes Gerät löschen, indem Sie das Gerät in der Liste **Devices in project** markieren und dann auf **Delete device** klicken. Bitte beachten Sie, dass nur neu angelegte Geräte gelöscht werden können. Geräte, die beim Öffnen des Projekts schon vorhanden sind, können nicht gelöscht werden.
 - Wollen Sie lediglich ein bestehendes Gerät ändern, markieren Sie dieses im Fenster **Devices in project**.
4. Markieren Sie in der Liste **Devices** das Gerät, das Sie nachfolgend konfigurieren möchten und klicken Sie auf die Schaltfläche **Next**.

Es öffnet sich das Fenster *Equipment module selection*.

5. Wählen Sie im Fenster *Equipment module selection* das oder die Standardmodule, die Sie in das gewählte Gerät integrieren möchten und klicken Sie auf die Schaltfläche **Next**.

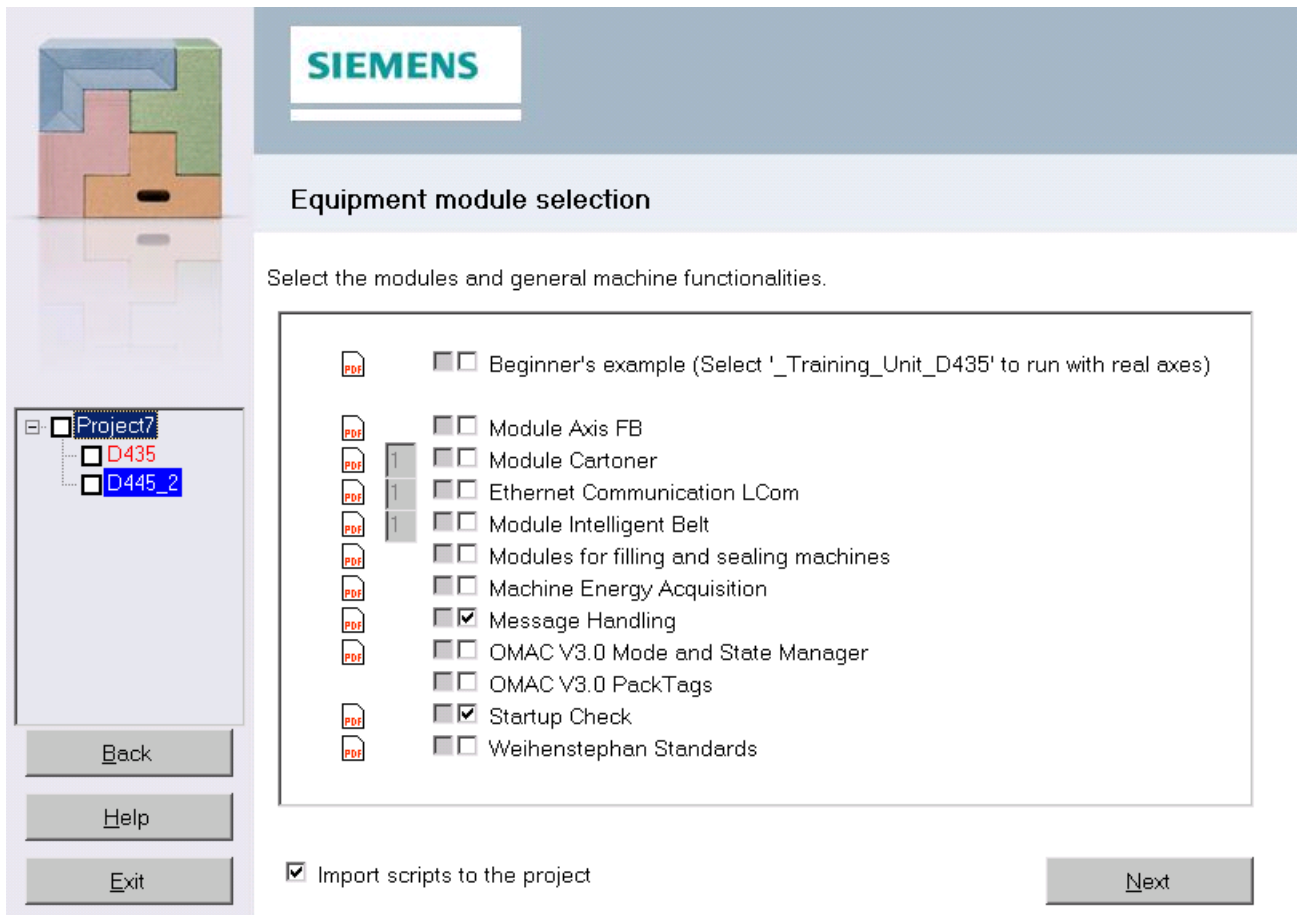


Bild 1-17 Standardmodule wählen

Wenn Sie auf die Schaltfläche **PDF** klicken, wird die Dokumentation zum jeweiligen Standardmodul geöffnet.

Bei einigen Standardmodulen, können Sie links die Anzahl der Module eingeben. Die linke senkrechte Reihe an Auswahlboxen zeigt die bereits im Projekt enthaltenen Module an.

Gilt nur für SIMOTION Geräte: Standardmäßig ist das Kästchen **Import scripts to the project** aktiviert. Damit werden alle Skripten, die sich im relativen Pfad SIMOTION\Scripts befinden, in die Projektebene des gewählten Projekts importiert.

Für unser Beispiel haben wir die Module *Message Handling* und *Startup Check* ausgewählt, die in den folgenden Masken konfiguriert werden.

6. Konfigurieren Sie im Fenster *Message Handling - Configuration* das gewählte Standardmodul **Message Handling** und klicken Sie auf die Schaltfläche **Next**.

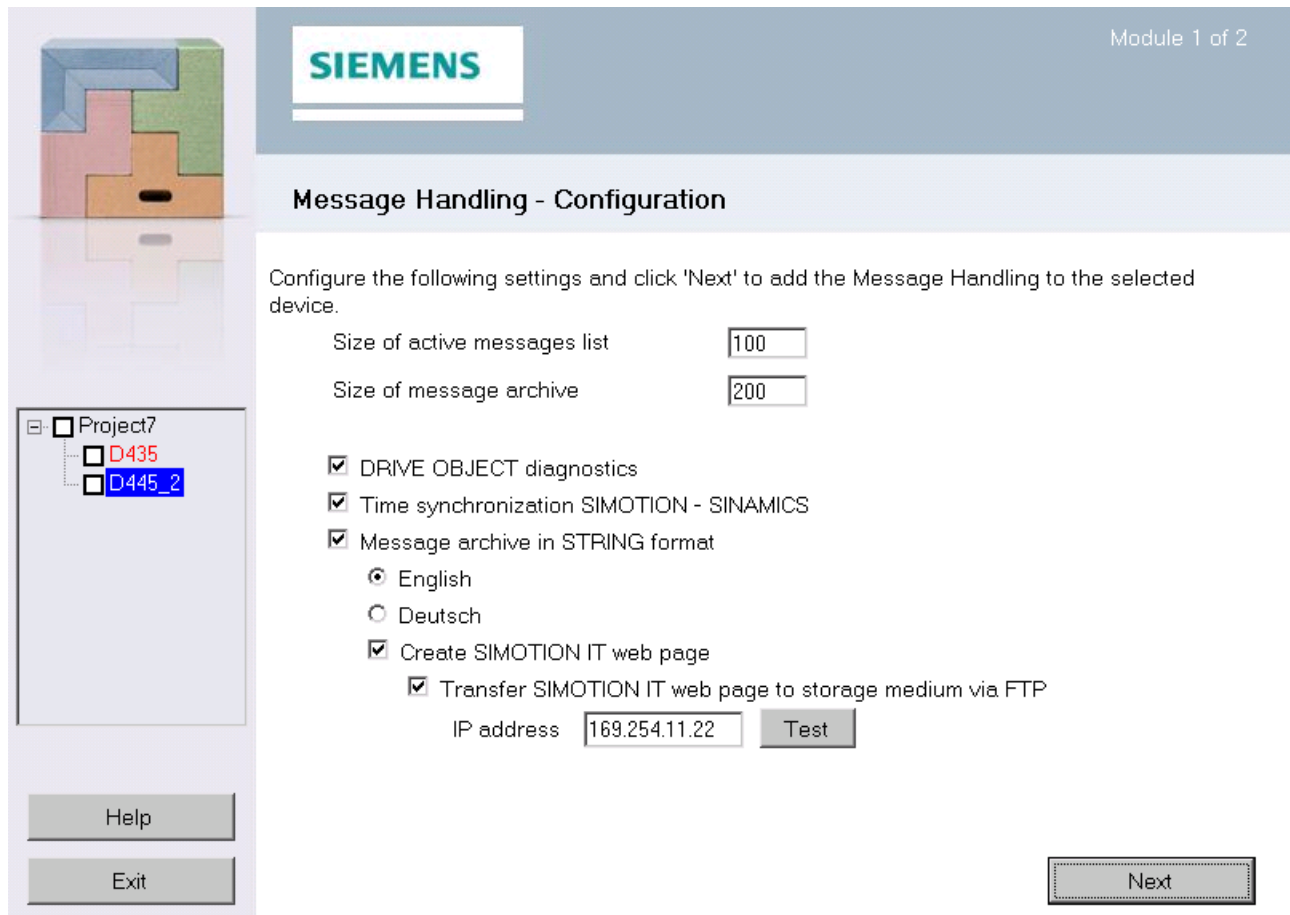


Bild 1-18 Konfiguration des Standardmoduls *Message Handling*

Sie können hier die folgenden Eingaben machen:

- **Größe der aktiven Meldungsliste** in Anzahl der Einträge
- **Größe des Meldungsarchivs** in Anzahl der Einträge
- **DRIVE OBJECT diagnostics**: Überwachung der SINAMICS Antriebsobjekte
- **Time synchronization SIMOTION – SINAMICS**: Synchronisation der Uhrzeit
- **Message archive in STRING format**:
Wählen Sie, ob Sie die Puffer im Format STRING in der Defaultsprache **Deutsch** oder **Englisch** verwenden möchten.

Wenn Sie das Optionskästchen **Create SIMOTION IT web page** zusammen mit **transfer SIMOTION IT web page to storage medium via FTP** aktivieren, wird die SIMOTION IT Seite auf das Speichermedium des SIMOTION Geräts gespeichert. Sie kann dann mit einem Standard WEB-Browser aufgerufen werden (SIMOTION IT). Dazu müssen Sie noch die IP-Adresse des SIMOTION Geräts angeben. Der Projektgenerator geht dann online und transferiert die Dateien für SIMOTION IT auf

das Speichermedium des SIMOTION Geräts. Über die Schaltfläche **Test** können Sie testen, ob die Verbindung zum SIMOTION Gerät möglich ist.

Wird das Optionskästchen **Create SIMOTION IT web page** nicht aktiviert, werden die Daten auf dem PC im Projektverzeichnis gespeichert.

7. Konfigurieren Sie das gewählte Standardmodul *Startup Check* und klicken Sie auf die Schaltfläche **Next**.

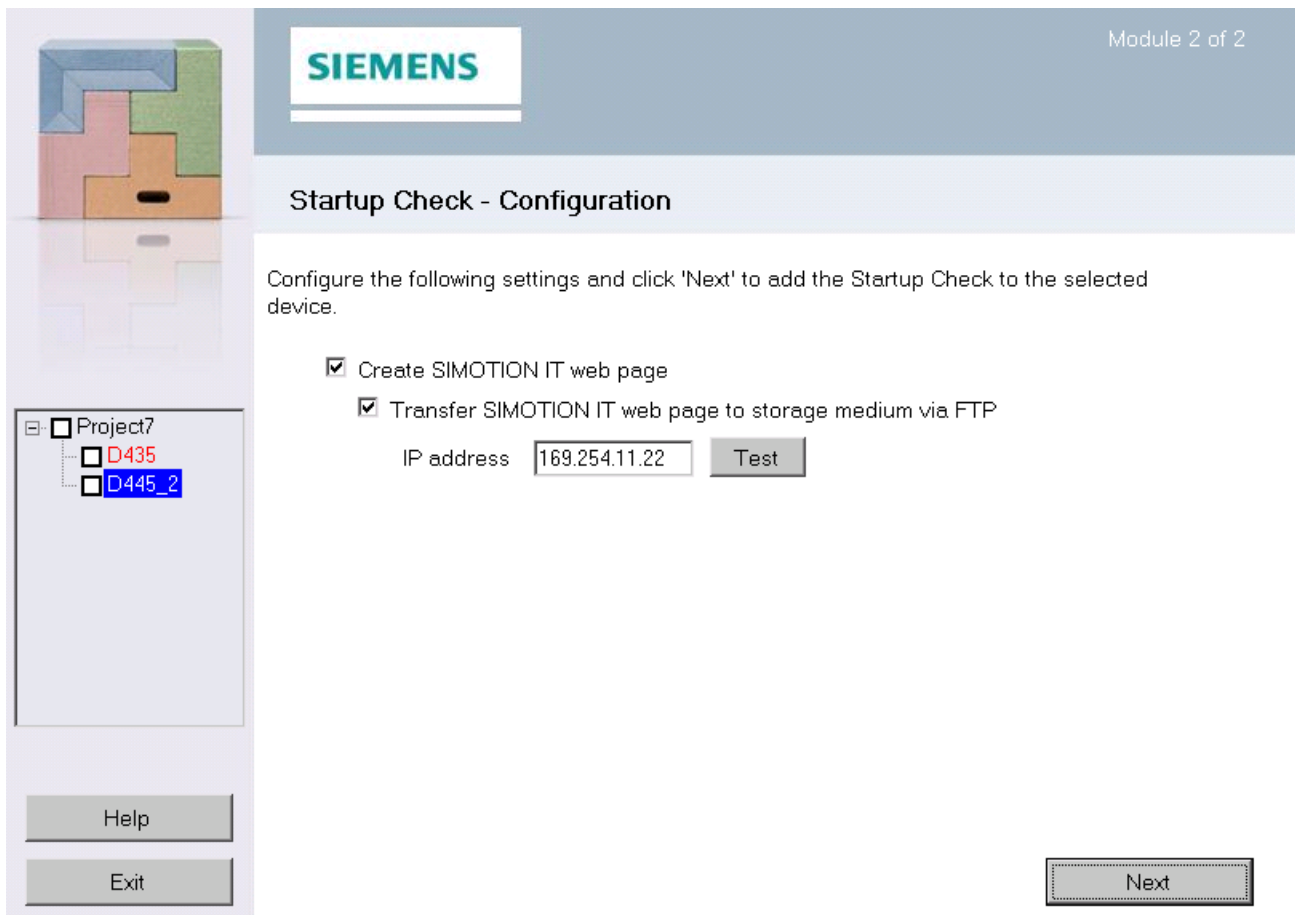


Bild 1-19 Konfiguration des Standardmoduls *Startup Check*

Die Bedeutung von **transfer SIMOTION IT web page to storage medium via FTP** siehe Schritt 6.

Es öffnet sich das Fenster *Generate the project*.

8. Wenn Sie über **Configure devices** kein weiteres Gerät mehr konfigurieren und das Projekt abschließen möchten, haben Sie folgende Möglichkeiten.

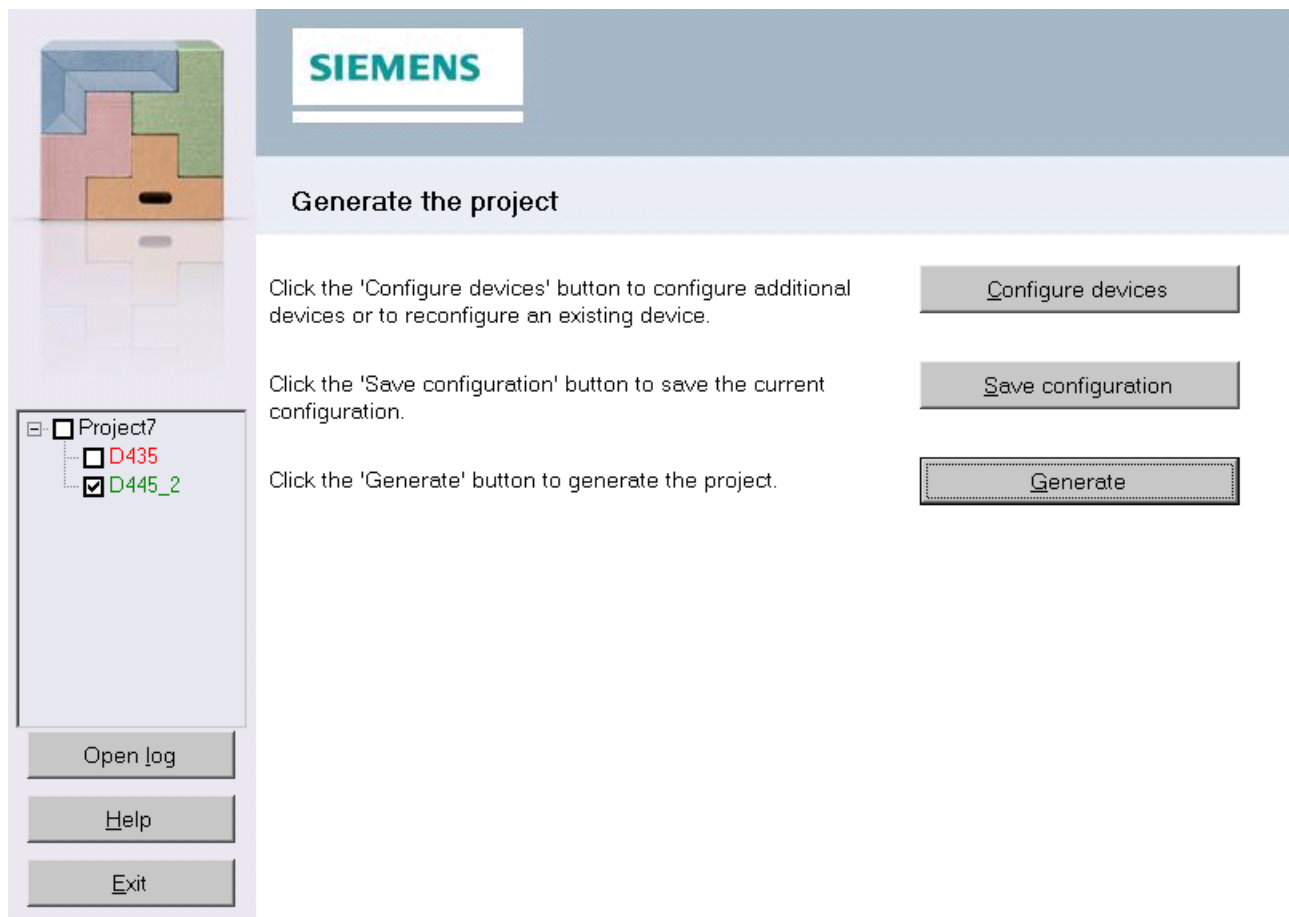


Bild 1-20 Projekt generieren

- Möchten Sie die Konfigurationen der Geräte speichern, aber erst zu einem späteren Zeitpunkt das Projekt generieren, dann klicken Sie auf die Schaltfläche **Save configuration** und geben im Explorer den Pfad zum Speicherort ein.
- Möchten Sie kein weiteres Gerät mehr konfigurieren und das Projekt generieren, dann klicken Sie auf **Generate**.

Das Projekt wird generiert. Die Dauer richtet sich nach der Art der Konfiguration und wird mit Hilfe eines Fortschrittbalkens angezeigt.

Hinweis

Bevor das Projekt bearbeitet wird, sichert der Projektgenerator das Projekt als Archiv im *Temp*-Verzeichnis des Benutzers, z. B. unter: C:\Documents and Settings\User\Local Settings\Temp\PGEN_SAVED_EXPORTS\Archive

9. Ist das Projekt fertig generiert, erscheint im Fenster die Meldung **Generation finished** und die Abfrage, ob Sie SIMOTION SCOUT öffnen möchten.

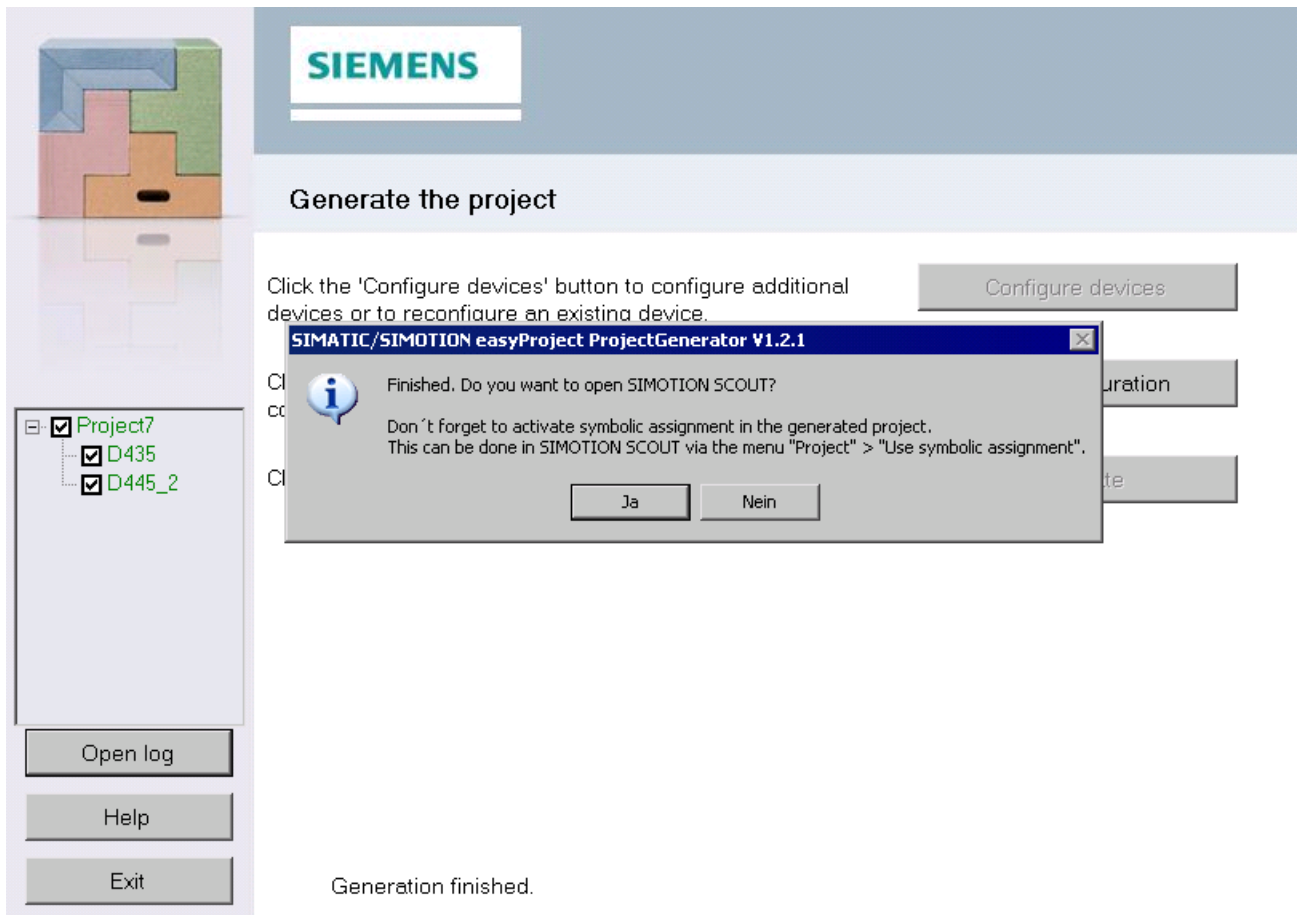


Bild 1-21 Projektgenerierung abgeschlossen

- Klicken Sie auf die Schaltfläche **Ja**, wenn Sie das Projekt in SIMOTION SCOUT öffnen möchten.

Der Projektgenerator wird geschlossen und das Projekt im SIMOTION SCOUT geöffnet. Die konfigurierten Module sind integriert und einsatzbereit.

Anwenderspezifische Funktionen können Sie im betreffenden Modul ergänzen. Details dazu finden Sie in den Dokumentationen zu den einzelnen Modulen.

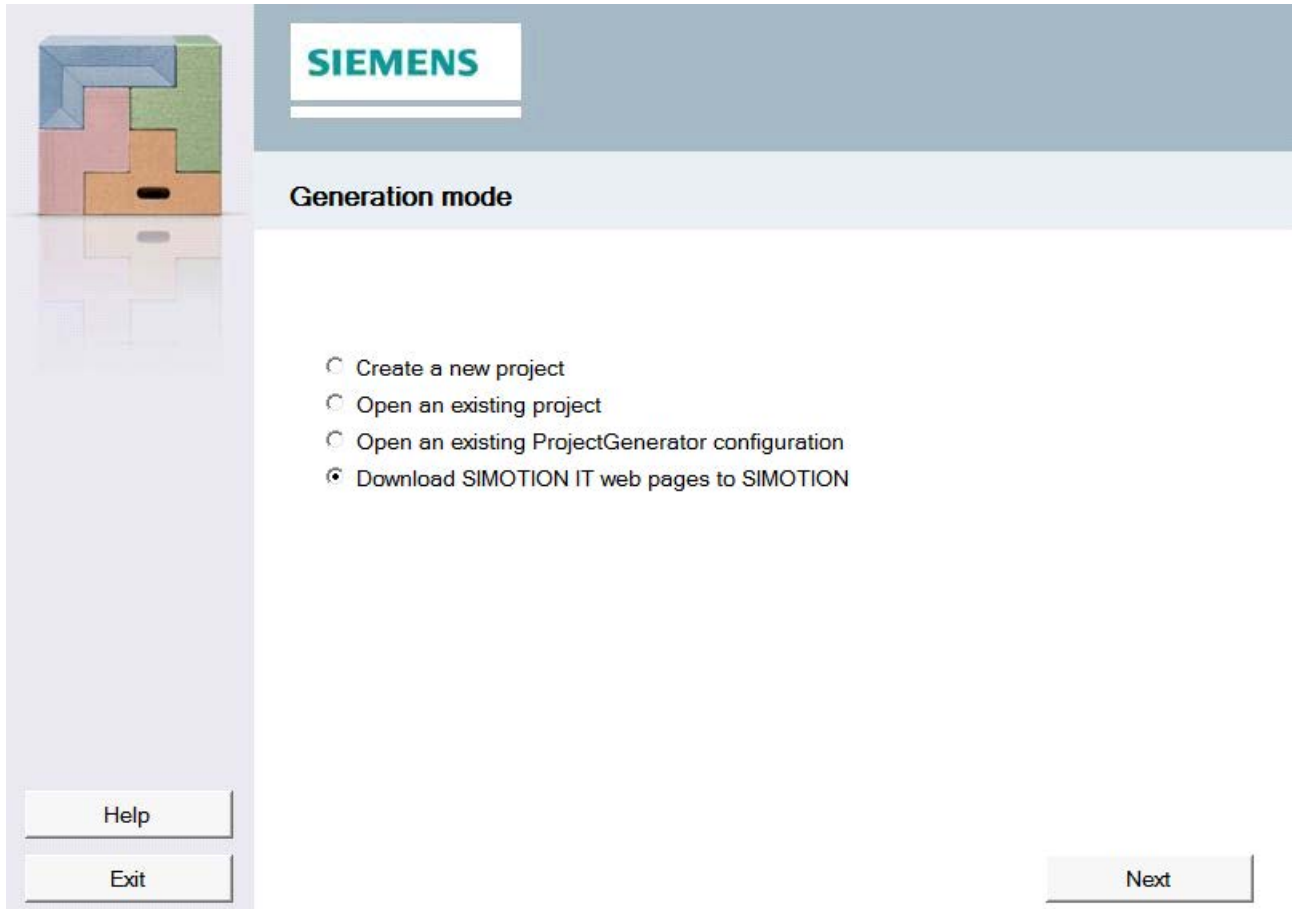
- Klicken Sie auf **Nein**, können Sie über **Save Configuration** die Anwenderangaben für diesen Projektgeneratordurchlauf speichern und den Projektgenerator über die Schaltfläche **Exit** verlassen.

1.2.4.4 Übertragen von SIMOTION IT Seiten

Seit der Version 1.3.0 des Projektgenerators gibt es, neben der bisherigen Möglichkeit SIMOTION IT Seiten während des Generierens via FTP Transfer auf eine Steuerung zu laden, auch die Möglichkeit nur die SIMOTION IT Seiten eines bestehenden Projekts auf eine Steuerung zu laden, ohne das Projekt erneut generieren zu müssen.

Diese Funktionalität wird im Folgenden beschrieben.

1. Starten Sie den Projektgenerator, bestätigen Sie den Haftungsausschluss und wählen Sie die Option **Download SIMOTION IT web pages to SIMOTION**.



2. Wenn Sie **Download SIMOTION IT web pages to SIMOTION** ausgewählt haben, können Sie den Pfad zu einem bestehenden Projekt, für das auch SIMOTION IT Seiten erzeugt wurden, auswählen. Hierbei gilt es darauf zu achten, dass nur das Verzeichnis, welches das Projekt beinhaltet, ausgewählt wird.
Des Weiteren muss die IP-Adresse der SIMOTION Steuerung angegeben werden. Zu diesem Zweck wird auch die Möglichkeit angeboten zu testen, ob es möglich ist eine Verbindung zu der Steuerung aufzubauen.

SIEMENS

Download SIMOTION IT web pages

Insert the IP address of the SIMOTION controller and select the folder of the project for which you would like to transfer the SIMOTION IT web pages.

Project path

IP address

Hinweis: Die IP-Adresse des Netzwerkadapters an Ihrem Computer und die IP-Adresse der SIMOTION Steuerung müssen in einem Subnetz liegen.

3. Nach der Angabe des Projektpfads und ggf. nach dem erfolgreichen Test der Verbindung, ist es möglich die Übertragung zu starten. Zu diesem Zweck muss der Button **Download** betätigt werden. Hier werden dann alle Dateien, welche sich unter PGEN_Data_Files\CardFiles befinden auf die Steuerung geladen.

Hinweis: Ab der Version 4.4 von SIMOTION wird beim Herunterladen der SIMOTION IT Seiten durch den Projektgenerator nach einem Benutzer und einem Kennwort gefragt. Diese Informationen müssen in der Benutzerverwaltung der SIMOTION vorhanden sein, sodass der Transfer funktionieren kann.

SIEMENS

Download SIMOTION IT web pages

Insert the IP address of the SIMOTION controller and select the folder of the project for which you would like to transfer the SIMOTION IT web pages.

Project path:

IP address:

Back
Help
Exit

Transferring ...

4. Während des Transfers wird Ihnen im unteren linken Teil der Benutzeroberfläche durch **Transferring** angezeigt, dass die Übertragung noch nicht abgeschlossen ist. Ist die Übertragung beendet, so wird dies durch **Transfer completed** angezeigt. Ist die Übertragung beendet, so haben Sie die Möglichkeit eine weitere Übertragung zu starten, über den Button **Back** zurück zum Generierungsmodus zu gelangen oder den Projektgenerator zu beenden.

SIEMENS

Download SIMOTION IT web pages

Insert the IP address of the SIMOTION controller and select the folder of the project for which you would like to transfer the SIMOTION IT web pages.

Project path

IP address

Transfer completed.

1.3 Anwenderspezifische Erweiterungen

1.3.1 Geräte und Standardmodule hinzufügen

1.3.1.1 Projektgenerator erweitern

Sie können den Projektgenerator um Geräte und Module erweitern, ohne den Quellcode des Projektgenerators ändern zu müssen.

Für die Erweiterung des Projektgenerators mit Modulen können Sie entweder die mit dem Projektgenerator ausgelieferten Standardmodule (siehe Kapitel Standardmodule hinzufügen (Seite 42)) oder auch anwenderspezifische Module verwenden.

Die Erweiterung des Projektgenerators nehmen Sie im Unterverzeichnis **SIMATIC/SIMOTION** des **Projektgenerators** vor (siehe Kapitel Lieferumfang (Seite 10)).

1.3.1.2 Geräte hinzufügen

So fügen Sie ein neues Gerät hinzu

Sie können eigene vorprojektierte Geräte (Devices) in das Verzeichnis **SIMATIC/Devices** oder **SIMOTION/Devices** mit aufnehmen, damit diese künftig in der Projektgenerator-Oberfläche zur Auswahl zur Verfügung stehen.

SIMATIC bzw. SIMOTION Gerät hinzufügen

- Legen Sie unter der entsprechenden Version einen Ordner mit dem Gerätenamen an.
 - Für SIMOTION Geräte gilt: Der Name des Ordners (z. B. SIMOTION_D435, siehe Bild unten) muss identisch mit dem Namen der exportierten XML-Datei des **Geräts** (z. B. SIMOTION_D435.xml) aus SIMOTION SCOUT sein.
 - Für SIMATIC Geräte gilt: Der Name des Ordners muss identisch mit dem Namen der exportierten CFG-Datei des **Geräts** aus STEP 7 sein.

Hinweis

Es wird empfohlen, Geräte- und Ordnername mit SIMATIC_ bzw. SIMOTION_ zu beginnen.

- Speichern Sie die exportierte CFG- bzw. XML-Datei des **Geräts** in das neu erstellte Verzeichnis.
- Speichern Sie den CFG- bzw. XML-Export der **Station** (z. B. XML_SIMOTION_D435(Station)) in das neu erstellte Verzeichnis.

Ab dem nächsten Start erkennt der Projektgenerator das neue Gerät und bietet es zum Import an.

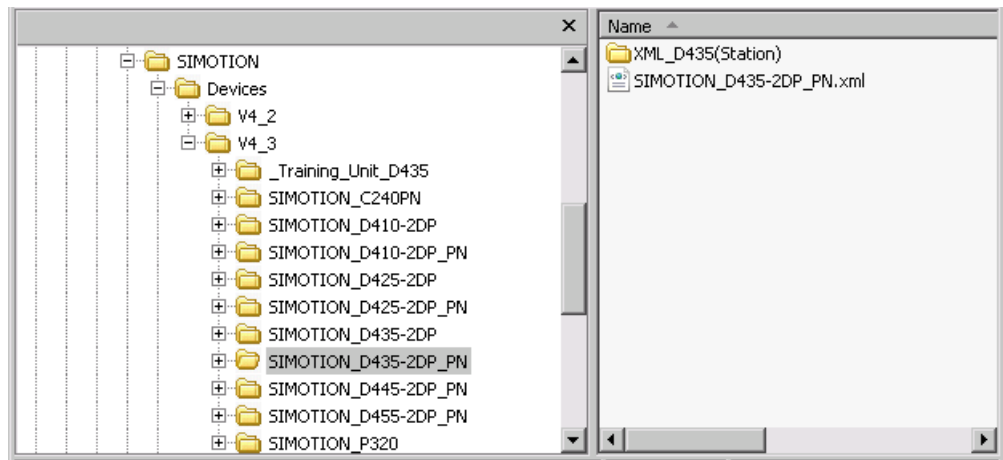


Bild 1-22 Beispiel: Geräte hinzufügen - SIMOTION D435

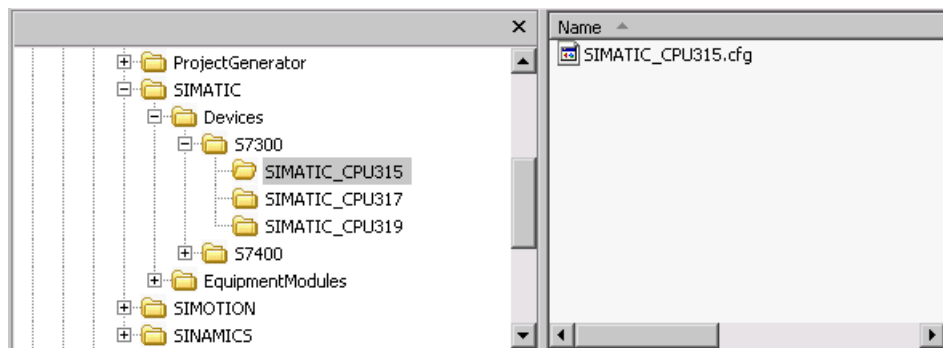


Bild 1-23 Beispiel: Geräte hinzufügen - SIMATIC Devices

1.3.1.3 Standardmodule hinzufügen

So fügen Sie ein Standardmodul hinzu

1. Legen Sie im Verzeichnis **SIMATIC/EquipmentModules** bzw. **SIMOTION/EquipmentModules** des Projektgenerators einen Ordner für das neue Standardmodul an.
2. Legen Sie in diesem Ordner eine Textdatei (.txt) an, geben Sie ihr einen entsprechenden Namen und ändern Sie das Format von *.txt* nach *.xml*.
3. Öffnen Sie die XML-Datei und legen Sie die Datenstruktur und Beschreibung für das neue Standardmodul fest.

Orientieren Sie sich dabei an den mit dem Projektgenerator ausgelieferten XML-Dateien der vorprojektierten Module.

4. Der Name des Ordners muss identisch mit dem Namen der XML-Datei sein (siehe Bild).

Orientieren Sie sich bezüglich des Inhalts des Ordners an den anderen Standardmodulen.

Der Projektgenerator erkennt beim nächsten Start das neue Modul und bietet es zum Import an.

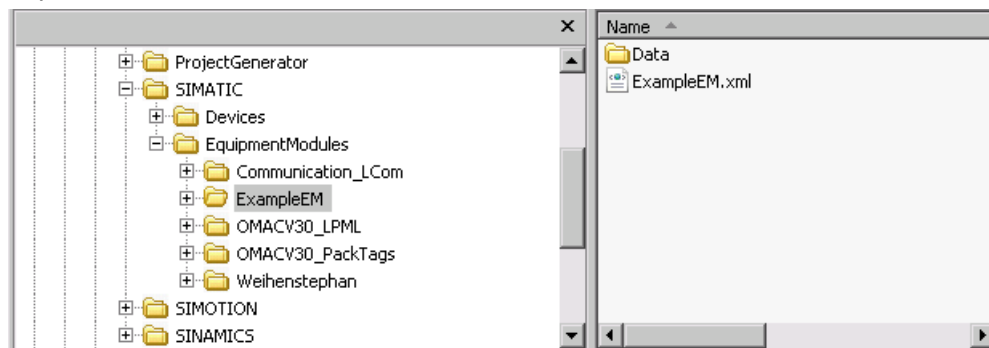


Bild 1-24 Beispiel: SIMATIC Modul hinzufügen

1.3.2 Technologieobjekte hinzufügen (nur SIMOTON)

So fügen Sie Technologieobjekte hinzu

Um ein weiteres Technologieobjekt hinzuzufügen, legen Sie im Projektgenerator-Verzeichnis **SIMOTION/TechnologyObjects** den XML-Export des neuen Technologieobjekts unter das entsprechende Versionsverzeichnis.

Der Projektgenerator erkennt beim nächsten Start das neue Technologieobjekt und bietet es zum Import an.

Jedes Standardmodul kann seine eigenen Technologieobjekte mitbringen und adressieren. Dazu legen Sie bereits an Ihre Bedürfnisse angepasste Technologieobjekte unterhalb Ihres Moduls ab und importieren diese anstatt der Technologieobjekte im Standardverzeichnis.

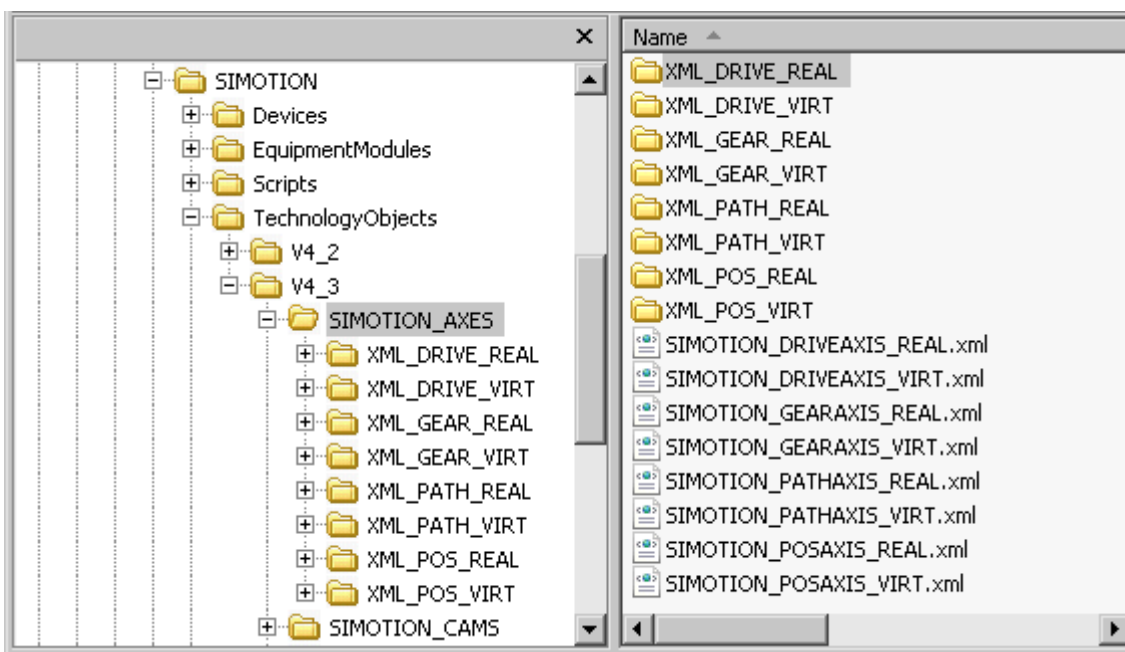


Bild 1-25 Beispiel: SIMOTION Technologieobjekte hinzufügen

Hinweis

Die Technologieobjekte **Nocken**, **Messtaster** und **Gleichlaufobjekt** können Sie nur in Verbindung mit einer Achse hinzufügen.

1.3.3 Anwenderspezifische Module erstellen und hinzufügen

Sie können über den Projektgenerator auch anwenderspezifische Module in ein Projekt einfügen, ohne sich mit der notwendigen Programmierung auseinanderzusetzen.

Sie können dabei alle Kommandos nutzen, mit denen auch die vorhandenen Standardmodule eingefügt werden. Eine genaue Beschreibung der einzelnen Kommandos finden Sie in den Listenhandbüchern *SIMATIC_Projektgenerator_Listenhandbuch* und *SIMOTION_Projektgenerator_Listenhandbuch*.

Aufbau der anwenderspezifischen Module

Hinweis

Zur Erstellung anwenderspezifischer Module empfehlen wir, dass Sie sich bezüglich Aufbau und Inhalt an den ausgelieferten Standardmodulen orientieren. Diese können als Kopiervorlage oder als Orientierungshilfe für eigene Erweiterungen verwendet werden.

Das erste Element in einem Modul ist ein *CommandList*-Tag. Hier wird unter dem Attribut *Name* der Name der Bibliothek angegeben, um beim erneuten Durchlauf des Projektgenerators erkennen zu können, ob schon ein Modul dieses Typs vorhanden ist.

Mit dem Attribut *DisplayText* wird der Text beschrieben, der bei der Auswahl des Moduls angezeigt wird.

Ist ein Modul nicht multiinstanzfähig, so kann die Anzahl über das Attribut *MaxNumberOfModules* begrenzt werden.

Die folgenden Aussagen gelten nur für den Projektgenerator für SIMOTION:

- Besitzt ein Modul keine oder keine eindeutige Bibliothek, wird auf einen festen Namen einer Unit referenziert, indem das Attribut *Mode="UnitOnly"* gesetzt wird.

Beispiel CommandList-Tag

Tabelle 1- 5 Beispiel CommandList-Tag

```
<CommandList Name="pStartupCheck"
             DisplayText="Use Startup Check"
             MaxNumberOfModules="1"
             Mode="UnitOnly"
             . . .
>/CommandList>
```

Nach dem *CommandList*-Tag folgen als *Child*-Tags die *Command*-Tags. Alle *Command*-Tags besitzen innerhalb des Moduls eine eindeutige Identifikationsnummer (ID) sowie den Namen der Funktion, die aufgerufen werden soll. Die ID ist immer eine positive Zahl. Der Einstieg des Moduls beginnt immer mit der ID 1, ansonsten kann die ID beliebig vergeben werden.

Soll nach Abschluss eines Kommandos direkt ein weiteres Kommando angeschlossen werden, kann dies durch das Attribut *NCID NextCommandID* erfolgen. Im Attribut *NCID* wird

die ID des nächsten Kommandos eingegeben. Der Kommandoname wird im Attribut *Name* eingetragen.

Beispiele Command-Tags

Tabelle 1- 6 Beispiel Command-Tag 1

```
<Command ID="1" Name="ImportUnit" NCID="2">  
    . . .  
</Command>
```

Für einige Informationen, die im Projekt und in der Projektgenerator-Datenhaltung vorhanden sind, müssen Sie eine gesonderte Logik verwenden, um diese Informationen z. B. an ein Control weiterzugeben.

Beispiel: Eine Textbox soll den aktiven Devicenamen anzeigen. Der Programmierer der XML-Beschreibungsdateien kann den Namen nicht wissen. Hierzu sind Funktionen vorhanden, die diese Informationen zurückgeben. Diese Funktionen werden mit dem Präfix *__Call_* versehen, damit der Projektgenerator erkennt, dass hier eine Systemfunktion verwendet wird. Der Projektgenerator ersetzt den Systemfunktionsaufruf anschließend durch die Rückgabeinformation. Wird z. B. dem *Text*-Tag einer Textbox der Wert *__call_GetSimotionDeviceName* zugewiesen, wird dieser Funktionsaufruf durch den aktiven Gerätenamen ersetzt, z. B. *D435*.

Einigen dieser Funktionen müssen Übergabeparameter übergeben werden. Dies erfolgt mit der Syntax von *Visual Basic .NET*. Eine Ausnahme sind hier die Anführungszeichen. Diese können in den meisten Fällen weggelassen werden, da eine automatische Konvertierung in das richtige Format erfolgt. Es kann aber auch das *'*-Zeichen anstatt des Anführungszeichens benutzt werden.

Tabelle 1- 7 Beispiel Command-Tag 2

```
<Control Action="add"  
    Type="TextBox"  
    Name="TB_DeviceName"  
    Text="__call_GetSimotionDeviceName"  
    ReadOnly="true"  
    Location="175, 300"  
    Autosize="true"  
    ToolTip="Actual device name">  
</Control>
```

1.4 Architektur des Projektgenerators

1.4.1 Allgemeines

In diesem Kapitel wird die Architektur des Projektgenerators beschrieben. Diese Informationen sind speziell für Anwender, die den Projektgenerator um eigene anwenderspezifische Module erweitern wollen.

1.4.2 Struktur des Projektgenerators

Der Projektgenerator besteht aus folgenden Komponenten:

Systemkern bestehend aus SL-Objekt und XML-Interpreter

- **SL-Objekt**
Innerhalb des Projektgenerators befindet sich das **SL-Objekt**. Das SL-Objekt beinhaltet alle Funktionen zum Lesen und Schreiben der Dateien sowie die Funktionen, die zum Erstellen und Bearbeiten eines Projekts notwendig sind.
- **XML-Interpreter**
Der **XML-Interpreter** greift zur Konfiguration eines Moduls auf ein Standardmodul bzw. ein anwenderspezifisches Modul zu. Der XML-Interpreter generiert aus den Befehlen der XML-Konfigurationsdateien die Oberfläche des Projektgenerators. Nach Abschluss der Konfiguration entscheidet der Projektgenerator, ob noch weitere Module konfiguriert werden müssen und ob die *Framework.xml* weiterbearbeitet wird. Mit dem XML-Interpreter erfolgt ein lesender Zugriff auf ein geöffnetes Projekt.

Framework

Für die Projektauswahl, die Auswahl des Geräts und die Erstellung der Übersicht über die vorhandenen Standardmodule greift der Projektgenerator auf das **Framework** zu.

CFG- bzw. XML-Konfigurationsdateien

Bei der Erstellung eines Projekts wird auf die folgenden Konfigurationsdateien zugegriffen:

- Framework greift auf die systeminternen Konfigurationsdateien zu.
- Zur Konfiguration der einzelnen Module wird auf die CFG- bzw. XML-Konfigurationsdateien der Standardmodule und der anwenderspezifischen Module zugegriffen.

Basisdaten

Zur Generierung eines Projekts wird auf die Basisdaten zurückgegriffen, die außerhalb des Projektgenerators im Verzeichnis **SIMATIC** bzw. **SIMOTION** abgelegt sind (siehe Kapitel Lieferumfang (Seite 10)). Die genaue Pfaddefinition zu den Basisdaten erfolgt in den Befehlen der XML-Konfigurationsdateien.

STEP 7-Projekt

Der lesende oder schreibende Zugriff auf das Projekt erfolgt über die STEP 7-Kommandoschnittstelle bzw. die SIMOTION SCOUT Scripting Schnittstelle.

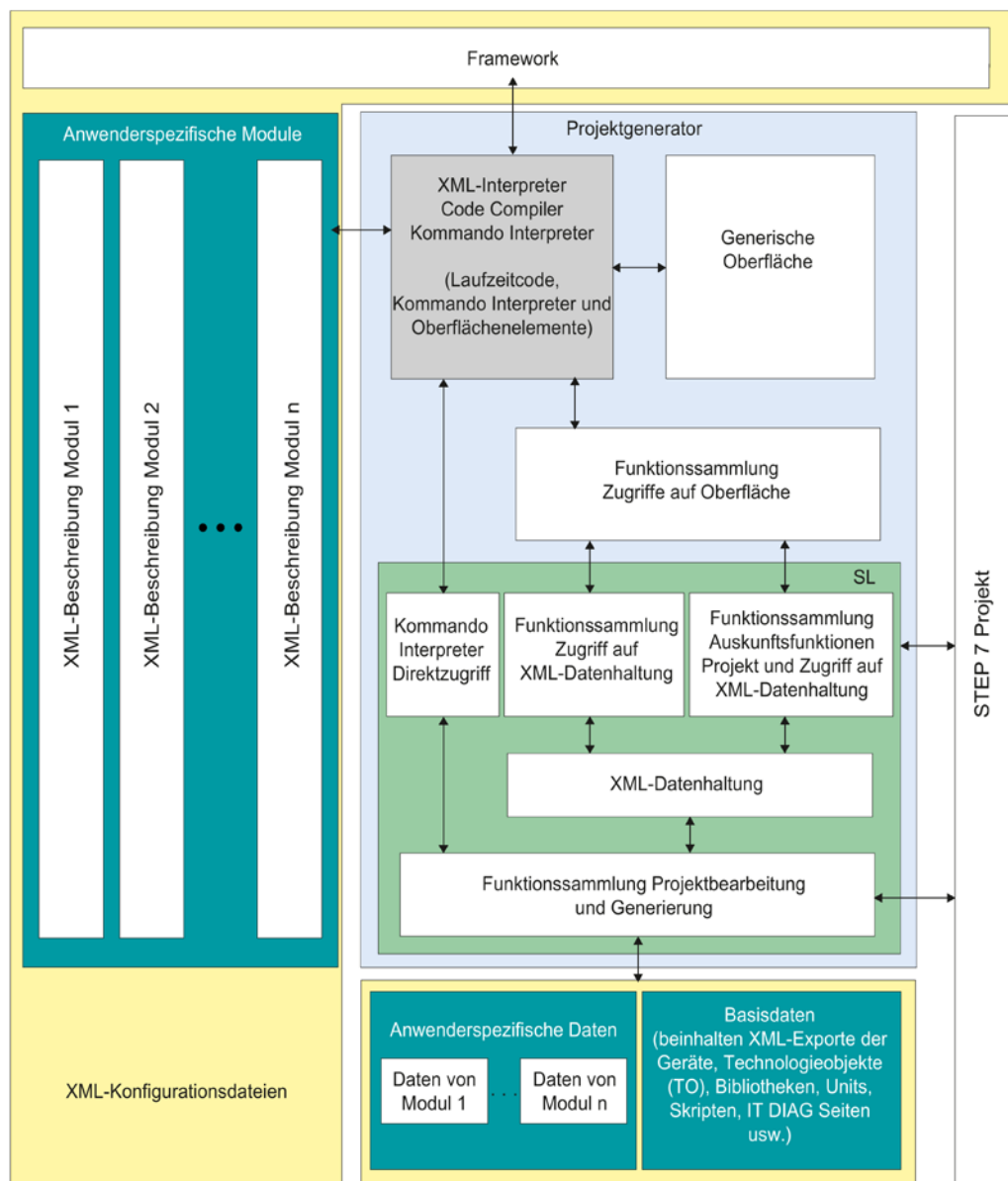


Bild 1-26 Systemarchitektur Projektgenerator

1.4.3 Verwendung des Projektgenerators im Silent Mode

Projektgenerator

Der Projektgenerator kann über den **Silent Mode** in anwenderspezifische Tools eingebettet werden. Dabei wird auf die Oberfläche des Projektgenerators verzichtet.

Im Fehlerfall werden keine Fehlermeldungen ausgegeben. Fehler können im Logfile des Projektgenerators eingesehen werden.

Externe XML-Datei

Die benötigten Informationen und Befehle werden dem Projektgenerator im **Silent Mode** über eine externe XML-Datei beim Startaufruf übergeben. Dabei werden die in der Datei enthaltenen Befehle direkt abgearbeitet und es werden keine Meldungen aufgeblendet. Sind Fehler aufgetreten, können diese im Logfile nachvollzogen werden.

Erstellung einer externen XML-Datei

Während der Erstellung und Generierung eines Projekts mit dem Projektgenerator werden alle Eingaben und Konfigurationen protokolliert und im *Temp*-Verzeichnis des Benutzers (C:\Documents and Settings\Benutzername\Local Settings\Temp) unter dem Dateinamen *PGEN_DATA_BASE.xml* abgelegt.

Sie können sich eine auf Ihre Bedürfnisse zugeschnittene XML-Datei erzeugen, indem Sie mit dem Projektgenerator einen Durchlauf mit den gewünschten Einstellungen und Konfigurationen machen und dann die daraus gewonnene XML-Datei *PGEN_DATA_BASE.xml* dem Projektgenerator beim Startaufruf mit übergeben.

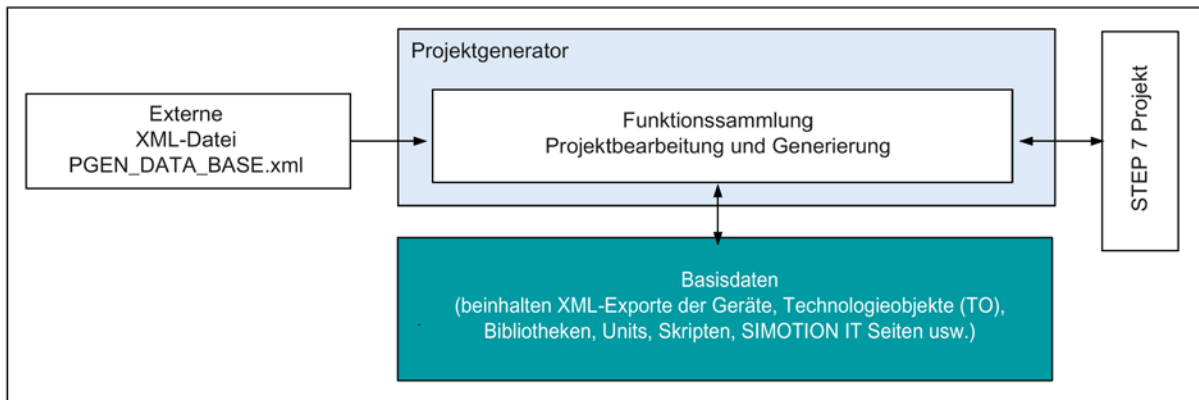


Bild 1-27 Systemarchitektur im Silent Mode

Basisdaten

Zur Generierung des Projekts wird auf die Basisdaten im Verzeichnis **SIMATIC** bzw. **SIMOTION** zurückgegriffen.

STEP 7 Projekt

Der lesende oder schreibende Zugriff auf das Projekt erfolgt über die STEP 7-Kommandoschnittstelle bzw. die SIMOTION SCOUT Scripting Schnittstelle.

System- und Fehlermeldungen

2.1 Allgemeines

Die folgenden Meldungen werden an der Oberfläche des Projektgenerators oder im System-Logfile ausgegeben.

Das System-Logfile finden Sie nach Durchlauf des Projektgenerators im Windows Ordner *temp* unter dem Dateinamen *PGenLogfile.xml*.

2.2 Systeminformationen

Die Zeichenkombinationen {0} und {1} werden mit dem entsprechenden Namen oder dem Typ gefüllt.

I_CMD_001 bis I_CMD_052

Tabelle 2- 1 I_CMD_001 bis I_CMD_052

Meldung	Bedeutung
I_CMD_001	Value of variable is: {0}
I_CMD_002	Project was not found.
I_CMD_003	Parameter value {0}
I_CMD_004	Project '{0}' ({1}) opened.
I_CMD_005	IO '{0}' on device '{1}' found.
I_CMD_006	Slave '{0}' removed.
I_CMD_007	IO '{0}' deactivated.
I_CMD_008	IO '{0}' removed.
I_CMD_009	Project renamed. New name: '{0}'
I_CMD_010	Library renamed. New name: '{0}'
I_CMD_011	IO variable '{0}' removed.
I_CMD_012	IO container '{0}' exported to '{1}'.
I_CMD_013	Library '{0}' removed.
I_CMD_014	Library '{0}' exported to '{1}'.
I_CMD_015	IO variables imported from '{0}'.
I_CMD_016	Library '{0}' imported from '{1}'.
I_CMD_017	TO '{0}' imported from '{1}'.
I_CMD_018	TO renamed. New name: '{0}'
I_CMD_019	TO '{0}' removed.
I_CMD_020	TO '{0}' exported to '{1}'.
I_CMD_021	DO renamed. New name: '{0}'

Meldung	Bedeutung
I_CMD_022	DO '{0}' removed.
I_CMD_023	DO '{0}' exported to '{1}'.
I_CMD_024	Unit renamed. New name: '{0}'
I_CMD_025	Unit '{0}' exported to '{1}'.
I_CMD_026	Unit '{0}' removed.
I_CMD_027	Program '{0}.{1}' added to task '{2}'.
I_CMD_028	Program '{0}.{1}' removed from task '{2}'.
I_CMD_029	Script '{0}' exported to '{1}'.
I_CMD_030	Script '{0}' removed.
I_CMD_031	DP-Slave '{0}' added.
I_CMD_032	Device '{0}' downloaded.
I_CMD_033	Project '{0}' closed.
I_CMD_034	Project '{0}' copied to '{1}'.
I_CMD_035	Project renamed. New name: '{0}'
I_CMD_036	Slot index '{0}' removed.
I_CMD_037	Separator from slot index '{0}' removed.
I_CMD_038	New slot '{0}' added for slave '{1}'.
I_CMD_039	Device '{0}' downloaded to folder '{1}'
I_CMD_040	IO variables exported to '{0}'.
I_CMD_041	Slot '{0}' copied to '{1}'.
I_CMD_042	Watchtable renamed. New name: '{1}'
I_CMD_043	Unit object '{0}' imported from '{1}'.
I_CMD_044	Old library {0} removed.
I_CMD_045	Slot '{0}' copied.
I_CMD_046	Slot '{0}' moved to '{1}'.
I_CMD_047	STEP 7 project closed.
I_CMD_048	Global variable '{0}' added.
I_CMD_049	Last separator removed from slot '{0}'.
I_CMD_050	STEP 7 HW Config opened.
I_CMD_051	STEP 7 HW Config closed.
I_CMD_052	Device renamed. New name: '{0}'

I_SYS_SL_001 bis I_SYS_SL005

Tabelle 2-2 I_SYS_SL_001 bis I_SYS_SL005

Meldung	Bedeutung
I_SYS_SL_001	Simulation active. Function not started.
I_SYS_SL_002	Parameter not optional at parameter column ({0}).
I_SYS_SL_003	Parameter converted from '{0}' to '{1}'.
I_SYS_SL_004	Call function {0}.
I_SYS_SL_005	Log file stored in {0}.

PG_I_001 bis PG_I_090

Tabelle 2- 3 PG_I_001 bis PG_I_090

Meldung	Bedeutung
PG_I_001	Generation started.
PG_I_002	Opening project.
PG_I_003	Generating project.
PG_I_004	Opening device.
PG_I_005	Importing device.
PG_I_006	Setting device config data.
PG_I_007	Importing TOs.
PG_I_008	Setting master slave connection.
PG_I_009	Setting config data of TO.
PG_I_010	Setting system data of TO.
PG_I_011	Importing the units.
PG_I_012	Deleting unit defines.
PG_I_013	Setting unit defines.
PG_I_014	Importing watchtables.
PG_I_015	Importing libraries.
PG_I_016	Importing unit to library.
PG_I_017	Deleting library unit defines.
PG_I_018	Setting library unit defines.
PG_I_019	Starting compilation.
PG_I_020	Closing project.
PG_I_021	Downloading to filesystem.
PG_I_022	Transferring files to filesystem.
PG_I_023	Transferring files to server.
PG_I_024	Generation finished.
PG_I_025	Finished, but errors occurred at compile time. Do you want to open SIMOTION SCOUT?
PG_I_026	Finished. Do you want to open SIMOTION SCOUT?
PG_I_027	Show the workbench.
PG_I_028	Setting property '{1}' of task '{0}' to value '{2}'.
PG_I_029	Importing project scripts.
PG_I_030	Offline unit '{0}' is newer. Do you want to replace it?
PG_I_031	Unit '{0}' already exists. Do you want to replace it?
PG_I_032	New configuration of unit '{0}' can be different. Please check the correct order of the references.
PG_I_033	Renaming task.
PG_I_034	Setting properties of the tasks.
PG_I_035	Setting name of task '{0}' to value '{1}'.
PG_I_036	Rearranging programs in the execution system.
PG_I_037	Program '{1}' of unit '{0}' in task '{2}' is rearranged to position '{3}'.
PG_I_038	Value '{0}' was outside the limits '{1}'. It will be changed to min/max value.

Meldung	Bedeutung
PG_I_039	Text length was greater than '{0}'.
PG_I_040	Do you really want to close this tool?
PG_I_041	Ping was successful.
PG_I_042	Wrong input The default value will override your input.
PG_I_043	SIMOTION SCOUT is not installed.
PG_I_044	SIMATIC STEP 7 is not installed.
PG_I_045	File not found. This tool will be closed.
PG_I_046	Registry key not found.
PG_I_047	Unit '{0}' already exists. Do you want to replace it?
PG_I_048	Offline library '{0}' is newer. Do you want to replace it?
PG_I_049	Object with the Name '{0}' already exists. Choose another name.
PG_I_050	Object cannot be deleted.
PG_I_051	Ping failed.
PG_I_052	Wrong IP-Address.
PG_I_053	Section between the labels '{0}' and '{1}' not found. (Regex Syntax)
PG_I_054	The ProjectGenerator cannot be started from a network drive.
PG_I_055	<p>Important Information</p> <p>The software and the appropriate documentation provided on this medium are made available at no charge. Customers are granted the non-exclusive and non-transferable right to use the software at no charge. This includes the right to modify the software, to copy the software either unchanged, or changed, as well as to link to customer's own software.</p> <p>The software was not subject to the standard system test that Siemens AG normally applies to software. Any liability of Siemens - irrespective of the legal grounds -, in particular due to errors in the software or the appropriate documentation, or damages arising from advice/consultation, is excluded unless mandatory by law, e.g. in cases of willful misconduct, gross negligence, personal injury or death, failure to meet guaranteed characteristics, fraudulent concealment of a defect or in case of breach of fundamental contractual obligations. The above stipulations shall not change the burden of proof to the detriment of the customer.</p> <p>These terms shall be governed by German law without recourse to its conflict of law provisions. The place of jurisdiction shall be Erlangen.</p>
PG_I_056	The ProjectGenerator cannot start modules from a network drive.
PG_I_057	The symbol with name '{0}' already exists with an different address. Do you want to replace the existing symbol?
PG_I_058	The symbol address '{0}' is already in use with a different object. Do you want to replace the existing symbol address?
PG_I_059	Importing HW Config.
PG_I_060	Importing SIMATIC libraries.
PG_I_061	Generating symbol table.
PG_I_062	Importing SIMATIC sources.
PG_I_063	Importing code to Organization Blocks.
PG_I_064	SIMATIC DB '{0}' already exists. Do you want to replace it?
PG_I_065	Importing Weihenstephan Communication SIMATIC sources.
PG_I_066	DB '{0}' already exists. Do you want to replace it?
PG_I_067	Source '{0}' compiled with errors. Do you want to open SIMATIC logfile?
PG_I_068	Source '{0}' compiled with errors. Details can be found in SIMATIC log file.

Meldung	Bedeutung
PG_I_069	Configured IP-address for next module is '{0}'
PG_I_070	Importing SINAMICS CU320-2.
PG_I_071	File '{0}' is not a valid STEP7 project.
PG_I_072	The SINAMICS station with the name '{0}' already exists on the interface '{1}'. Choose a different name.
PG_I_073	The maximal numbers of DP-Slaves exceeded on this interface.
PG_I_074	The PROFIBUS address is already used. Choose a different one.
PG_I_075	The IP-Address is already used. Choose a different one.
PG_I_076	Importing IO-Tables.
PG_I_077	Connection of TO '{{0}}' to DO '{{1}}' on device '{{2}}' established.
PG_I_078	Generating OPC export files (*.sti).
PG_I_079	Setting SINAMICS Parameters & BiCos.
PG_I_080	Importing DOs to the SINAMICS devices.
PG_I_081	Creating HW Config Slots for SINAMICS DOs.
PG_I_082	Creating IOs in the address lists.
PG_I_083	Creating global variables.
PG_I_084	Importing AlarmS messages.
PG_I_085	Creating AlarmS messages
PG_I_086	Executing scripts in project.
PG_I_087	Setting preprocessor defines.
PG_I_088	Importing global variable tables.
PG_I_089	Setting unit compile options.
PG_I_090	Deleting unit compile options.

2.3 Warnungen

Die Zeichenkombination {0} wird mit dem entsprechenden Namen bzw. Typ gefüllt.

W_CMD_001 bis W_CMD_012

Tabelle 2-4 W_CMD_001 bis W_CMD_012

Meldung	Bedeutung
W_CMD_001	Copying of an opened project not possible. Project '{0}' will be closed. Afterwards reference is automatically generated.
W_CMD_002	No project open.
W_CMD_003	Project must be offline.
W_CMD_004	Project name already existing. Cannot rename.
W_CMD_005	Slot address '{0}' not found.
W_CMD_006	DP-Subsystem '{0}' not found.
W_CMD_007	New name of imported TO is already used. Renaming is not possible.
W_CMD_008	Reopening of STEP 7 – HW Config could cause access problems.
W_CMD_011	Project '{0}' not closed.
W_CMD_012	Task '{0}' not activated.

W_SYS_SL_001 bis W_SYS_SL_009

Tabelle 2-5 W_SYS_SL_001 bis W_SYS_SL_009

Meldung	Bedeutung
W_SYS_SL_001	No optional language packet selected or selected message {0} not found.
W_SYS_SL_002	No optional language packet selected.
W_SYS_SL_003	Function will also be executed in simulation mode.
W_SYS_SL_004	Project '{0}' still open. Will be closed automatically.
W_SYS_SL_005	Warning: {0}
W_SYS_SL_007	Directory '{0}' exists.
W_SYS_SL_008	Directory '{0}' exists and will be removed.
W_SYS_SL_009	Log file extension not correct.

2.4 Fehlermeldungen

Die Zeichenkombinationen {0}, {1}, {2} werden mit dem entsprechenden Namen bzw. dem Typ gefüllt.

E_CMD_002 bis E_CMD_047

Tabelle 2- 6 E_CMD_002 bis E_CMD_047

Meldung	Bedeutung
E_CMD_002	Import from file ({0}) caused an error.
E_CMD_003	Too many project files in folder '{0}'.
E_CMD_004	Project file (.s7p) not found in folder '{0}'.
E_CMD_005	Project not open.
E_CMD_006	No write access to parameter {0}.
E_CMD_007	Project must be online.
E_CMD_008	Enum value '{0}' not defined.
E_CMD_009	Statement '{0}' not defined.
E_CMD_010	Drive object type error: Imported DO-type is '{0}'.
E_CMD_011	Optional parameter value '{0}' not defined.
E_CMD_012	Could not go to {0}.
E_CMD_013	Enum value '{0}' not defined.
E_CMD_014	State of device not changeable.
E_CMD_015	Unexpected error in STEP 7-Project.
E_CMD_016	Unexpected error with IO address for IO '{0}'.
E_CMD_017	No slave on the bus '{0}'.
E_CMD_018	IO variable '{0}' not correct.
E_CMD_019	DO '{0}' not found.
E_CMD_020	Import of Unit not successful.
E_CMD_021	Device '{0}' not found.
E_CMD_022	Object '{0}' not found.
E_CMD_023	Unit '{0}' not found.
E_CMD_024	Project could not be copied. Destination folder already exists.
E_CMD_025	Project could not be closed.
E_CMD_026	Symbol '{0}' not found.
E_CMD_027	Timeout. Expected parameter value not reached in time.
E_CMD_028	Slot address not writeable.
E_CMD_029	CU '{0}' not found.
E_CMD_030	Project could not be opened.
E_CMD_031	Ethernet port '{0}' not found.
E_CMD_032	TO '{0}' not found.
E_CMD_033	Library '{0}' not found.
E_CMD_034	Renaming of TO not possible, name '{0}' already exist.
E_CMD_035	Target slot not empty. Copying not possible.

Meldung	Bedeutung
E_CMD_036	Technological alarm '{0}' not found.
E_CMD_037	Parameter '{0}' not found.
E_CMD_038	Symbol '{0}' not found.
E_CMD_039	CU '{0}' not found.
E_CMD_040	NO VALUE.
E_CMD_041	Script '{0}' not found.
E_CMD_042	Watchsymbol '{0}' not found.
E_CMD_043	STEP 7 HW Config not open.
E_CMD_044	S7 Station not found.
E_CMD_045	S7 Subsystem not found.
E_CMD_046	Slot with address '{0}' not found.
E_CMD_047	Parameter must be greater than zero.

E_CMD_500 bis E_CMD_599

Tabelle 2- 7 E_CMD_500 bis E_CMD_599

Meldung	Bedeutung
E_CMD_500	Add control to form. Type: '{1}'; Name: '{0}'.
E_CMD_501	Start generation.
E_CMD_502	Project already open.
E_CMD_503	New project with name: '{0}' in folder: '{1}' generated.
E_CMD_504	New device with name: '{0}' and version: '{1}' and type: '{2}' is generated.
E_CMD_505	The device is already active.
E_CMD_506	The device '{0}' already open.
E_CMD_507	Open device '{0}' command is added in XML database.
E_CMD_508	Setting configuration data '{0}' at TO '{2}' to value '{1}'.
E_CMD_509	Connection between master '{0}' and following object '{1}' established.
E_CMD_510	Setting system data '{0}' at TO '{2}' to value '{1}'.
E_CMD_511	Section between '{0}' and '{1}' restored.
E_CMD_512	Constant '{0}' set to value '{1}'.
E_CMD_513	Taskinformation added.
E_CMD_514	Label '{0}' replaced with '{1}'.
E_CMD_515	Information about '{2}' added to section between '{0}' and '{1}'.
E_CMD_516	Adding program '{0}' to execution system.
E_CMD_517	The program '{0}' already exists in the execution system.
E_CMD_518	Preprocessor instruction in unit '{0}' set to '{1}'.
E_CMD_519	Preprocessor instruction in unit '{0}' in library '{2}' set to '{1}'.
E_CMD_520	Importing watchtable '{0}' from folder '{1}' .
E_CMD_521	Adding template code '{2}' between the labels '{0}' and '{1}'.
E_CMD_522	Technology package of library '{0}' set to '{1}'.
E_CMD_523	Constants for unit '{0}' saved. Number of VAR_GLOBAL CONSTANT sections found is '{1}'.

Meldung	Bedeutung
E_CMD_524	Constants in unit '{0}' restored.
E_CMD_525	Unit '{1}' imported to library '{0}'.
E_CMD_526	Project is being compiled.
E_CMD_527	Ping to IP-Address '{0}' failed. The transfer will not be executed.
E_CMD_528	Transfer from local folder '{1}' to FTP-Server '{0}' successfully finished.
E_CMD_529	'{1}' ({0}) read.
E_CMD_530	Value incremented to '{0}'.
E_CMD_531	Generation successfully finished.
E_CMD_532	Entry '{1}' deleted in object '{0}'
E_CMD_533	Object '{0}' (type '{1}') deleted in XML database.
E_CMD_534	Entry '{1}' created in object '{0}'
E_CMD_535	Writing constant '{0}' in unit '{2}' to value '{1}' in XML database.
E_CMD_536	Writing constant '{0}' in library '{2}' and library unit '{3}' to value '{1}' in XML database.
E_CMD_537	Writing master '{0}' <-> slave '{1}' connection in XML database.
E_CMD_538	Writing new TO '{0}' of type '{1}' in XML database.
E_CMD_539	Setting TO '{0}' config data '{1}' to value '{2}'.
E_CMD_540	Setting TO '{0}' system data '{1}' to value '{2}'.
E_CMD_541	Writing label '{0}' in unit '{2}' to value '{1}' in XML database.
E_CMD_542	Writing constant '{0}' in library '{2}' and library unit '{3}' to value '{1}' XML database.
E_CMD_543	Writing task information for unit '{0}' in XML database.
E_CMD_544	Writing task information for library '{1}' and unit '{0}' in XML database.
E_CMD_545	Writing device and slave information for unit '{0}' in XML database.
E_CMD_546	Creating backup data of unit '{0}' in XML database.
E_CMD_547	Creating backup data of library '{1}' and unit '{0}' in XML database.
E_CMD_548	Assigning program '{0}' to task '{1}' in XML database.
E_CMD_549	Open Browser.
E_CMD_550	Selected path: '{0}'
E_CMD_551	Reading next Equipment Module.
E_CMD_552	Creating backup of constant in unit '{1}' in library '{0}' in XML database.
E_CMD_553	Setting define '{0}' in unit '{1}' in XML database.
E_CMD_554	Setting define '{0}' in unit '{1}' in library '{2}' in XML database.
E_CMD_555	Setting TP '{1}' in library '{0}' in XML database.
E_CMD_556	Setting FTP transfer from local path '{0}' to IP address '{1}' in XML database.
E_CMD_557	Setting import of unit '{0}' in library '{1}' in XML database.
E_CMD_558	Setting import of watchtable '{0}' from path '{1}' in XML database.
E_CMD_559	Configuration data '{1}' for device '{0}' is set to value '{2}'.
E_CMD_560	Configuration data '{1}' with value '{2}' is set for device '{0}' in XML database.
E_CMD_561	Deleting preprocessor define '{1}' in unit '{0}'.
E_CMD_562	Deleting preprocessor define '{1}' in unit '{0}' in library '{2}'.
E_CMD_563	Restoring defines in unit '{1}' in library '{0}'.
E_CMD_564	Deleting define '{0}' in unit '{1}' is set in XML database.
E_CMD_565	Deleting define '{0}' in unit '{1}' in library '{2}' is set in XML database.

Meldung	Bedeutung
E_CMD_566	Restoring defines in unit '{0}'.
E_CMD_567	Restoring defines in unit '{0}' in library '{1}'.
E_CMD_568	Creating backup of constants in unit '{0}' in XML database.
E_CMD_569	Transfer command for the files in '{0}' to '{1}' is set in XML database.
E_CMD_570	Instruction 'USEPACKAGE' activated in unit '{0}'.
E_CMD_571	File '{0}' not found.
E_CMD_572	No code found to set a constant.
E_CMD_573	No device active.
E_CMD_574	Directory '{0}' not found.
E_CMD_575	No device GUID found.
E_CMD_576	No TP GUID found.
E_CMD_577	GUID of device '{0}' not found.
E_CMD_578	GUID of TP '{0}' not found.
E_CMD_579	Text of the sender was empty.
E_CMD_580	There was no pattern given.
E_CMD_581	Create directory on the server '{0}' not possible.
E_CMD_582	Create file on the server '{0}' not possible.
E_CMD_583	Unit not found.
E_CMD_584	Wrong file extension.
E_CMD_585	Wrong input parameter.
E_CMD_586	CU not found.
E_CMD_587	Object with name '{0}' already exists. Choose another name.
E_CMD_588	Set auto-define TP in unit '{0}' in XML database.
E_CMD_589	Set auto-define TP in unit '{0}' in library '{1}' in XML database.
E_CMD_590	'{0}' is not a valid IP address.
E_CMD_591	Path does not exist.
E_CMD_592	Set auto-define TP in unit '{0}' in XML database.
E_CMD_593	Set auto-define TP in unit '{0}' in library '{1}' in XML database.
E_CMD_594	Property '{0}' is not available in the Task '{1}'.
E_CMD_595	Value '{1}' not allowed for property '{0}'.
E_CMD_596	Property '{0}' not found.
E_CMD_597	Could not set the property '{1}' of task '{0}' to value '{2}'.
E_CMD_598	Property '{0}' not found.
E_CMD_599	Task '{0}' not found.

E_CMD_600 bis E_CMD_664

Tabelle 2- 8 E_CMD_600 bis E_CMD_664

Meldung	Bedeutung
E_CMD_600	The object with the name '{0}' already exists. Choose another Name.
E_CMD_601	The object with the name '{0}' already exists. Do you want to replace it?
E_CMD_602	The object with the name '{0}' can't be replaced, because it's a real TO.
E_CMD_603	The following object with the index '{1}' of the techobject '{0}' was renamed to '{2}'.
E_CMD_604	The following object with the index '{1}' of the techobject '{0}' was renamed to '{2}'. Index was not found.
E_CMD_605	The following object of the techobject '{0}' cannot be renamed due to wrong index '{1}'.
E_CMD_606	Techobject '{0}' has no following objects to rename.
E_CMD_607	Writing rename following object '{0}' from the object '{1}' in XML database.
E_CMD_608	Data type error in line '{0}'
E_CMD_609	Initial value wrong in line '{0}'
E_CMD_610	The number of the identifiers STRUCT and END_STRUCT is not the same.
E_CMD_611	The length of the cell, row = '{0}' column= '{1}' is wrong.
E_CMD_612	Project '{1}' in path '{0}' already exists. Choose another path or name.
E_CMD_613	Unit with the name '{0}' already exists. Choose another name.
E_CMD_614	IP-Address of device '{0}' changed to address '{1}' on module '{2}'
E_CMD_615	Writing label '{0}' in SIMATIC source '{2}' to value '{1}' in XML database.
E_CMD_616	Setting symbolic block name '{0}' of block number '{1}' in XML database.
E_CMD_617	Adding networks in path '{1}' to Organization Block '{0}' at position '{2}'.
E_CMD_618	Writing constant '{0}' in source '{2}' to value '{1}' in XML database.
E_CMD_619	Importing IL (AWL) source '{0}' from folder '{1}'.
E_CMD_620	Port '{0}' not found. Port renamed to '{1}'.
E_CMD_621	Setting TO '{0}' to DO '{1}' on device '{2}' in XML database.
E_CMD_622	Removing TO - DO connection on TO '{0}' from the XML database.
E_CMD_623	Creating SINAMICS device '{0}' of type '{1}' with address '{2}'.
E_CMD_624	The device with the name '{0}' already exists. Proceeding with existing device.
E_CMD_625	Importing DO '{0}' and changing the power module to order no '{1}'.
E_CMD_626	Importing DO '{0}' and connecting it to the power unit of DO '{1}'.
E_CMD_627	Importing DO '{0}'.
E_CMD_628	Creating HW Config slots for device '{0}'.
E_CMD_629	Set BiCo connection from SrcName '{0}', SrcNumber '{1}' and SrcIndex '{2}' to target Name '{3}', Number '{4}' and Index '{5}'.
E_CMD_630	Setting parameter '{1}' on DO '{0}' to value '{2}'.
E_CMD_631	Writing new DO '{0}' to device '{1}' in XML database.
E_CMD_632	SINAMICS device '{1}' not found. Can't insert new DO '{0}'.
E_CMD_633	Writing new SINAMICS device '{0}' at CPU Interface '{1}' in XML database.
E_CMD_634	The SINAMICS device '{0}' at CPU Interface '{1}' already exists in XML database.
E_CMD_635	The IO variable '{0}' already exists in the project.
E_CMD_636	Writing new IO variable '{0}' with address '{1}' in XML database.

2.4 Fehlermeldungen

Meldung	Bedeutung
E_CMD_637	The global variable '{0}' already exists in the project.
E_CMD_638	Writing new global variable '{0}' with data type '{1}' in XML database.
E_CMD_639	The AlarmS message '{0}' already exists in the project.
E_CMD_640	Writing new AlarmS message '{0}' in XML database.
E_CMD_641	Writing AlarmS messages in path '{0}' in XML database.
E_CMD_642	Can't convert MessageClass number '{0}' into an Integer. Proceeding with value 0.
E_CMD_643	Can't convert AlarmID number '{0}' into an Integer. Proceeding with value -1.
E_CMD_644	The AlarmS message '{0}' will be removed.
E_CMD_645	Creating AlarmS message '{0}'.
E_CMD_646	Creating global variable '{0}'.
E_CMD_647	Creating IO variable '{0}'.
E_CMD_648	The DO '{0}' has no free X2 slot.
E_CMD_649	The X2 slot is already free at the DO '{0}'.
E_CMD_650	Deleting X2 slot at the DO '{0}'.
E_CMD_651	DO '{0}' on device '{1}' not found.
E_CMD_652	Setting parameter '{0}' to value '{1}'.
E_CMD_653	Connecting BiCo '{0}' to '{1}'.
E_CMD_654	Subsystem '{0}' found. Proceeding with this subsystem.
E_CMD_655	'{0}' subsystem '{1}' created.
E_CMD_656	Writing '{0}' device with name '{1}' in XML database.
E_CMD_657	Overwriting '{0}' device with name '{1}' in XML database.
E_CMD_658	The device '{0}' is a CPU and can't be used for a SINAMICS import.
E_CMD_659	Writing subsystem '{0}' property '{1}' with value '{3}' in XML database.
E_CMD_660	Subsystem '{0}' not found.
E_CMD_661	Setting property '{1}' to value '{2}' at subsystem '{0}'.
E_CMD_662	Sinamics device '{1}' at subsystem '{0}' not found.
E_CMD_663	Setting property '{1}' to value '{2}' in device '{3}' at subsystem '{0}'.
E_CMD_664	No CPU in the imported station found. Please check the .cfg file.

E_SYS_SL_001 bis E_SYS_SL_509

Tabelle 2- 9 E_SYS_SL_001 bis E_SYS_SL_509

Meldung	Bedeutung
E_SYS_SL_001	Exception: {0}
E_SYS_SL_002	Message '{0}' not found in external language file.
E_SYS_SL_006	Parameter '{0}' does not exist for function '{1}'
E_SYS_SL_007	Parameter column '{0}' not defined.
E_SYS_SL_008	Function attribute: critical error.
E_SYS_SL_009	Command '{0}' not found in Standard Library.
E_SYS_SL_010	{0} parameter conversion error: parameter '{1}' (Exception: {2})
E_SYS_SL_011	Parameter conversion error: '{0}' expected.
E_SYS_SL_012	No parameter in parameter column {0}.
E_SYS_SL_013	Internal error: XML node not found.
E_SYS_SL_014	Message '{0}' not found.
E_SYS_SL_015	Resource '{0}' not found.
E_SYS_SL_016	Standard Library function could not be started.
E_SYS_SL_017	XML file save error. Log file could not be written.
E_SYS_SL_018	Taskname '{0}' not defined.
E_SYS_SL_019	Element '{0}' of the enum list '{1}' is not defined.
E_SYS_SL_020	Directory '{0}' not found.
E_SYS_SL_021	STEP 7 project file (*.s7p) not found.
E_SYS_SL_500	Conversion Error "Router Active".
E_SYS_SL_501	IP address of router is not valid.
E_SYS_SL_502	IP address is not valid.
E_SYS_SL_503	SIMATIC station type not found or not implemented.
E_SYS_SL_504	Symbolic-Name '{0}' of block not found.
E_SYS_SL_505	Program container was not found.
E_SYS_SL_506	Searching for program container '{0}'. No project found.
E_SYS_SL_507	Variable '{0}' already exists in source.
E_SYS_SL_508	Variable '{0}' with type '{1}' generated.
E_SYS_SL_509	Selected Weihenstephan PDACONF.XML is incorrect ('{0}').

Tipps und Hilfestellung

3.1 Sonderzeichen in XML

Da einige Zeichen innerhalb von XML eine Bedeutung haben, müssen diese mit sogenannten Entitäten umschrieben werden. Dies gilt beispielsweise auch im Kommentarinhalt.

Wie stelle ich Sonderzeichen in XML dar?

Verwenden Sie in XML anstelle von Sonderzeichen die zugehörigen Entitäten.

Sonderzeichen und Umschreibung

Sonderzeichen	Entität
&	&
<	<
>	>
"	"
'	Chr(39)

Hinweis

Um Umlaute und andere Sonderzeichen in der XML Datei nutzen zu können, muss diese im UTF-8 Format kodiert werden.

3.2 Kommentare im Quellcode

Im Visual Basic-Code können aus Kompatibilitätsgründen Kommentare nicht mit dem einfachen Anführungszeichen (') eingefügt werden.

Wie füge ich Kommentare im Quellcode ein?

Um Ihrem Quellcode Kommentare hinzuzufügen, verwenden Sie die REM-Anweisung.

Beispiel

Tabelle 3- 1 Codebeispiel

REM Kommentar

3.3 MessageBox ausgeben

Das Und-Zeichen (&) kann als Verkettungsoperator nicht direkt benutzt werden. Programmieren Sie es entweder als Sonderzeichen oder verwenden Sie stattdessen das Plus-Zeichen (+).

Wie gebe ich eine MessageBox aus?

Um eine MessageBox aus dem Anwendercode auszugeben, kann die Visual Basic Funktion *Show* der Klasse *MessageBox* benutzt werden.

Codebeispiel

```
MessageBox.Show('Message ' + tmpText1)
```

3.4 Verwendbare Parameter, Events und Objekte

Aufgrund der Vielzahl möglicher Parameter und Events können in dieser Dokumentation nicht alle beschrieben werden.

Welche Parameter, Events und Objekte sind möglich?

Weiterführende Informationen zu den Parametern finden Sie im Internet in der Microsoft .NET Framework-Klassenbibliothek (<http://msdn.microsoft.com/de-de/library/ff361664.aspx>).

3.5 Textdateien verwenden

Wie arbeite ich mit Textdateien?

Es kann immer wieder nötig sein, Projektgenerator-fremde Dateien zu lesen bzw. zu schreiben, z. B. als Schnittstelle zu anderen Programmen.

Beispiel

Das nachfolgende Beispiel zeigt, wie im Eventcode eines Buttons eine Datei geschrieben und anschließend wieder gelesen wird.

Tabelle 3-2 Codebeispiel

```
<Control Action="add"
  Type="Button"
  Name="BT_WriteRead "
  Text="Write & Read "
  Location="650, 531"
  Size="130, 30"
  Enabled="true"
  ToolTip="Write & read a file">
<Events>
  <VisibleChanged code="@BT_Next@.Focus()"/>
  <Click code="
    Dim tmpWindowsPath As String
    Dim axNameArray(4) As String
    REM get windows temp path
    tmpWindowsPath = IO.Path.GetTempPath()

    REM create temp path for ProjectGenerator. projgen always use tmpProjectGenerator
    IO.Directory.CreateDirectory(tmpWindowsPath + 'tmpProjectGenerator\')

    REM copy text to array
    For i As Integer = 0 To Microsoft.VisualBasic.UBound(axNameArray)
      axNameArray(i) = 'Line' + i.ToString()
    Next

    REM Save some information for next steps
    IO.File.WriteAllLines(tmpWindowsPath + 'tmpProjectGenerator\' + 'axNameArray.txt',
axNameArray, System.Text.Encoding.Default)

    REM read and fill automatically
    Dim tmpProjGenPath As String
    Dim readAxNamesArray() As String

    REM get windows temp path
    tmpProjGenPath = IO.Path.GetTempPath() + 'tmpProjectGenerator\'
```

3.5 Textdateien verwenden

```
    If IO.File.Exists(tmpProjGenPath + 'axNameArray.txt') Then REM file exists?
        REM read file to the array
        readAxNamesArray = IO.File.ReadAllLines(tmpProjGenPath + 'axNameArray.txt', Sys-
tem.Text.Encoding.Default)

        REM show all read lines
        For Each Line As String In readAxNamesArray REM for all lines
            Microsoft.VisualBasic.Interaction.MsgBox(Line, , 'Show read lines')
        Next

    Else REM file does not exist
        Microsoft.VisualBasic.Interaction.MsgBox('File not found. Path = ' + tmpProjGen-
Path + 'axNameArray.txt',64, 'File not found')
    End If"/>
</Events>
</Control>
```

3.6 Quelle mehrfach verwenden

Wie verwende ich im Ablaufsystem eine Quelle mehrfach?

Um eine Quelle mehrfach im Ablaufsystem verwenden zu können, müssen alle Programmnamen sowie alle im Interface befindlichen globalen Variablen einen eindeutigen Namen besitzen. Hier bietet sich der Name der Quelle an. Dieser muss in jede der nötigen Variablen und Programmnamen eingesetzt werden.

Um diese Aufgabe vom Projektgenerator erledigen zu lassen, erstellen Sie Labels (z. B. <UnitName>) im Export der Unit.

Beispiel

Tabelle 3-3 Codebeispiel (Auszug aus dem Modul LCom)

```
//SIEMENS AG
//(c)Copyright 2008 All Rights Reserved
//-----
//file name:      pCom.st
//library:        uses LCom
//version:        SIMOTION / SCOUT 4.1.1.6
//restrictions:
//functionality:  communicate with other controllers with tcp
//-----
//change log table:
//version  date      expert in charge      changes applied
//01.01.00  05.2011    TM      multi instance corrected for scripting
//=====
INTERFACE
//----- Import -----
  USELIB LCom;
//----- Device Global Type Definitions -----
  TYPE
    //types for example data
    //-----
    s<sgUnitName>UserDataSendType : STRUCT; //type for example send data
    r32Spare1 : REAL;
    r32Spare2 : REAL;
    b32Spare1 : DWORD;
    b32Spare2 : DWORD;
    b8Spare1  : BYTE;
    b8Spare2  : BYTE;
  END_STRUCT;
  ...
END_TYPE
//----- Device Global Variables -----
  VAR_GLOBAL
```

3.6 Quelle mehrfach verwenden

```
gbo<sgUnitName>Enable          : BOOL := TRUE; //connect if cpu running
gbo<sgUnitName>Communicate     : BOOL := TRUE; //communicate if cpu running
gul6<sgUnitName>SendDataLength : UINT := 500; //byte length of send data
gs<sgUnitName>UserDataSend     : s<sgUnitName>UserDataSendType;
...
END_VAR
//----- Export -----
PROGRAM <sgProgramName>;
//-----
END_INTERFACE

IMPLEMENTATION
//-----
PROGRAM <sgProgramName>
...
END_IMPLEMENTATION
```

3.7 Oberfläche neu laden

Wie lade ich die Oberfläche neu?

Bei der Erstellung von Anwendermodulen ist es sehr hilfreich, wenn die Oberfläche neu geladen wird, ohne das Modul verlassen zu müssen. Hierzu kann nachfolgendes Konstrukt verwendet werden. Dieses ermöglicht Änderungen im XML-Code direkt durch einen Knopfdruck auf der Oberfläche sichtbar zu machen.

Beispiel

Im Beispiel wird der Button „Reload“ zur Oberfläche hinzugefügt. Wenn man auf den Button klickt, wird die Oberfläche abgebaut und automatisch neu aufgebaut.

Tabelle 3- 4 Codebeispiel

```
<Command ID="1" Name="ChangeForm" >
  <Control Action="add"
    Type="Button"
    Name="BT_Reload"
    Text="Reload"
    Location="450, 531"
    Size="130, 30"
    Enabled="true"
    Visible="true"
    ToolTip="Configuration of the Axis FB">
    <Events>
      <Click code="MyApp.NextCommand(100)"/>
    </Events>
  </Command>
<Command ID="100" Name="DestroyForm" NCID="1"/>
```

3.8 Generisch auf Oberflächenelemente zugreifen

Wie kann ich generisch auf Oberflächenelemente zugreifen?

Bei sehr komplexen Seiten ist es notwendig, generisch auf die Oberflächenelemente zugreifen zu können. Alle Elemente der aktuellen Oberfläche finden sich in der Collection „MyApp.Controls('Panel').Controls“.

Beispiel

Das Beispiel schaltet alle Oberflächenelemente unsichtbar, deren Name mit „CB_“ beginnt.

Tabelle 3-5 Codebeispiel

```
For Each tmpControl As Object In MyApp.Controls('Panel').Controls
    If tmpControl.Name.ToUpper.StartsWith('CB_') Then
        tmpControl.Visible = False
    End If
Next
```


3.9 Generisch die Oberfläche verändern

Wie verändere ich die Oberfläche generisch mit Anwender-XML?

Bei komplexen Modulen kann es notwendig sein, die Oberfläche generisch zu verändern, um schneller an das Ziel eines lauffähigen Moduls zu gelangen.

Beispiel

Das folgende Beispiel zeigt, wie der Inhalt der Oberfläche mithilfe einer XML-Datei (fclnitSyncDataList.xml) generisch angepasst werden kann. Ausgangspunkt ist die Maske im Bild 3.1. Die Maske beinhaltet zwanzig unsichtbare Gruppen von Controls (Visible = False). Im Bild sind zur Verdeutlichung vier dieser Gruppen zu sehen. Eine Gruppe besteht aus folgenden Controls:

- Ein Label für die Beschreibung des Parameters (LBL_FC1Para1_X). Dieses Label wird immer angezeigt und beinhaltet den Namen des Parameters.
- Eine Textbox (TB_FC1ParaX), eine Combobox (CmB_FC1ParaX) und eine Checkbox (CB_FC1ParaX). In der XML-Datei wird bestimmt, welcher dieser drei Controls angezeigt werden soll.
- Ein Label für eine eventuelle Einheit (LBL_FC1ParaX). Im Bild sind es die Labels mit dem Text „Unit“. Dieses Label wird nur dann angezeigt wenn es sich bei dem zweiten Control um eine Textbox handelt.

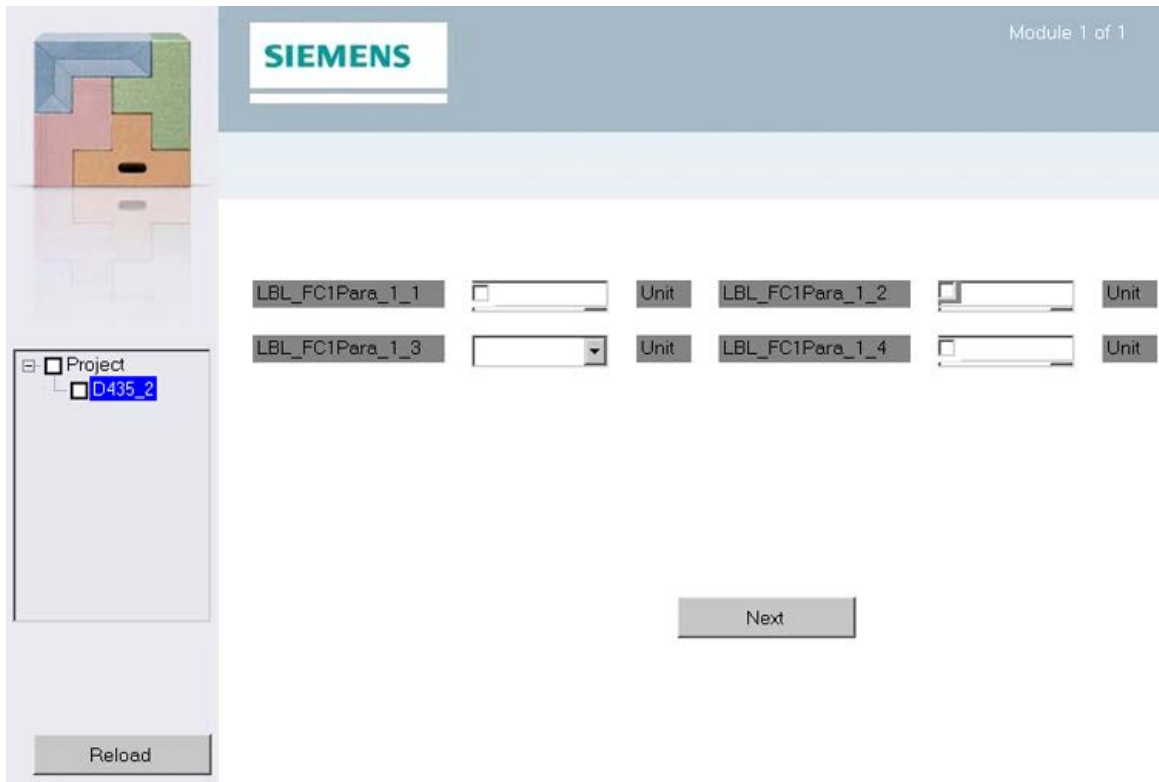


Bild 3-1 Beispiel: Oberfläche mit allen Elementen

3.9 Generisch die Oberfläche verändern

Die folgende Tabelle zeigt einen Codeabschnitt aus der XML-Datei des Equipment Module. In diesem Abschnitt wird das Modul definiert und die Oberfläche mit den Controls von Gruppe 1 neu gezeichnet.

Tabelle 3- 6 Codebeispiel Oberfläche erstellen

```
<CommandList Name="DocExample"
    DisplayText="DocExample"
    MaxNumberOfModules="1"
    ModuleInfoFile=""
    ToolTip="" >
<Command ID="1" Name="DestroyForm" NCID="2" />
<Command ID="2" Name="ResizeForm" NCID="3" X="860" Y="600" />
<Command ID="3" Name="ChangeForm" NCID="" >
    <Control Action="add"
        Type="Label"
        Name="LBL_FC1Para_1_1"
        Text="LBL_FC1Para_1_1"
        Visible ="True"
        Location="180, 200"
        AutoSize="False"
        Size="140,20"
        BackColor="__call_SetColor(Gray)" >
</Control>
<Control Action="add"
    Type="CheckBox"
    Name="CB_FC1Para1"
    Text=""
    Location="340,202"
    Size="17, 17"
    AutoSize="False"
    Visible ="True"
</Control>
<Control Action="add"
    Type="TextBox"
    Name="TB_FC1Para1"
    Text=""
    Location="340, 200"
    Size="100, 20"
    Autosize="false"
    Visible ="True">
```

In der XML Datei (fclnitSyncData-List.xml) wird jede Control Gruppe durch ein XML-Knoten (Item) konfiguriert. Da nur zwanzig Control Gruppen vorhanden sind, darf die Anzahl der Items in der XML-Datei diese Zahl nicht überschreiten. Je nach ausgewähltem Control müssen bestimmte Parameter vorgegeben werden. Im folgenden Codeabschnitt wird der vom Quellcode erwartete Aufbau der XML Datei gezeigt. Um bestimmte Control Gruppen zu überspringen und so Einfluss auf die Darstellung zu haben, wird ein Dummy Control „None“ definiert.

Tabelle 3-7 Codebeispiel Aufbau fclnitSyncDataList.xml

```
<GenericControls>
  <!-- TextBox selection -->
  <Item Control="TextBox"
    ParameterName ="Label text (LBL_FC1Paral_X)"
    Parameter="Label tooltip (LBL_FC1Paral_X)"
    InfoTooltip="TextBox tooltip"
    DefaultValue="TextBox text"
    RotaryUnit="Label unit text"/>

  <!-- CheckBox selection -->
  <Item Control="CheckBox"
    ParameterName ="Label text (LBL_FC1Paral_X)"
    Parameter="Label tooltip (LBL_FC1Paral_X)"
    InfoTooltip="CheckBox tooltip"
    DefaultValue="Checked setting: True or False"/>

  <!-- ComboBox selection -->
  <Item Control="ComboBox"
    ParameterName ="Label text (LBL_FC1Paral_X)"
    Parameter="Label tooltip (LBL_FC1Paral_X)"
    InfoTooltip="ComboBox tooltip"
    DefaultValue="Text of item to be selected initially"
    MaxItems="Number of items, provide at least this number of items"
    Item1="Item1 text"
    Item2="Item2 text"
    <!-- add more ComboBox items if needed --> />

  <!--skip control-->
  <Item Control="None"/>

</GenericControls>
```

In der nachfolgenden Tabelle ist ein konkretes Beispiel mit drei Controls. Die zweite Control Gruppe wird nicht angezeigt.

Tabelle 3-8 Codebeispiel fclnitSyncDataList.xml

```
<GenericControls>
```

3.9 Generisch die Oberfläche verändern

```

<!-- CheckBox to configure the new axis as a virtual axis -->
<Item Control="CheckBox"
    ParameterName ="Add virtual axis "
    Parameter=""
    InfoTooltip="Check to add a virtual axis"
    DefaultValue="False"/>

<!-- skip second group -->
<Item Control="None"/>

<!-- TextBox to enter the name of the new axis -->
<Item Control="TextBox"

    ParameterName ="Axis name "
    Parameter=""
    InfoTooltip="Name of the new axis"
    DefaultValue="NewAxis"
    RotaryUnit="" />

<!-- Combobox to select axis type -->
<Item Control="ComboBox"
    ParameterName ="Type of axis "
    Parameter=""
    InfoTooltip="Select the type of the new axis"
    DefaultValue="PosAxis"
    MaxItems="4"
    Item1="DriveAxis"
    Item2="PosAxis"
    Item3="GearAxis"
    Item4="PathAxis" />

</GenericControls>

```

Im Quellcode in der nachfolgenden Tabelle werden die Konfigurationsdaten der Oberfläche aus der XML-Datei eingelesen, die ausgewählten Controls angezeigt und mit den eingelesenen Daten initialisiert.

Tabelle 3- 9 Codebeispiel fcInitSyncDataList.xml

```

REM =====
REM Read fcInitSyncDataList and insert Labels, Textboxes, CheckBoxes and ComboBoxes
REM =====

REM Read the xml file and save the items in the Arraylist fcInitSyncDataList
Dim fcInitSyncDataList As System.Collections.Arraylist =
My-
App.myIsl.readxmlfile('SIMOTION\EquipmentModules\V4_3\DocExample\Data\Lists\fcInitSyncData
List.xml')

REM Check the number of the xml items. Exit the module (Command0) if any er-

```

```

ror
If fcInitSyncDataList.count = 0 Then          REM No items in XML file or file is mis-
sing
    Microsoft.VisualBasic.Interaction.MsgBox('List with parameters
SIMOTION\EquipmentModules\V4_3\UserModuleName\Data\Lists\fcInitSyncDataList.XML
could not be found or is empty.' + Microsoft.VisualBasic.vbCrLf + 'Please check
fcInitSyncDataList.xml.',64, 'fcInitSyncDataList is missing or corrupt')
    MyApp.NextCommand(0)
Else If fcInitSyncDataList.count > 20 Then    REM More than 20 items in XML file
    Microsoft.VisualBasic.Interaction.MsgBox('Parameter list contains more parameters
than allowed.' + Microsoft.VisualBasic.vbCrLf + 'Please check
fcInitSyncDataList.xml',64,'fcInitSyncDataList: too many parameters')
    MyApp.NextCommand(0)
Else                                          REM File and count ok

REM Loop through the items in the XML file
For i As Integer = 0 To fcInitSyncDataList.count - 1
    REM If the item is None continue with the next item
    If fcInitSyncDataList(i)('CONTROL') = 'None' Then
        Continue For
    End If

    REM Show the label LBL_FC1Para_1_X and set the text as in the
    REM attribute ParameterName in the XML file
    With MyApp.Controls('Panel').Controls('LBL_FC1Para_1_' + Cstr(i+1))
        .Visible = True
        .Text = fcInitSyncDataList(i)('PARAMETERNAME')
    End With

    REM Set the tooltip text of the label LBL_FC1Para_1_X as in the
    REM attribute 'PARAMETER' in the XML file
    MyApp.Tooltip.SetToolTip(MyApp.Controls('Panel').Controls('LBL_FC1Para_1_' +
Cstr(i+1)), fcInitSyncDataList(i)('PARAMETER'))

    REM =====Textbox selected =====
    If fcInitSyncDataList(i)('CONTROL') = 'TextBox' Then

        REM Show the textbox 'TB_FC1ParaX' and get the text from the variable
        REM with same name. If the variable does not exist use the attribute
        REM DefaultValue in the XML File
        With MyApp.Controls('Panel').Controls('TB_FC1Para' + Cstr(i+1))
            .Visible = True
            .Text = MyApp.myISL.getValueOfTemporaryVariable('TB_FC1Para' +
Cstr(i+1), 'Local', fcInitSyncDataList(i)('DEFAULTVALUE'))
        End With

        REM Set the tooltip of the textbox as in the attribute InfoTooltip
        MyApp.Tooltip.SetToolTip(MyApp.Controls('Panel').Controls('TB_FC1Para' +
Cstr(i+1)), fcInitSyncDataList(i)('INFOTOOLTIP'))

```

```

    REM Set the text of the unitlabel as in the attribute RotaryUnit
    MyApp.Controls('Panel').Controls('LBL_FC1Para' + Cstr(i+1)).Text =
    fcInitSyncDataList(i)('ROTARYUNIT')

    REM Make the unitlabel visible
    MyApp.Controls('Panel').Controls('LBL_FC1Para' + Cstr(i+1)).Visible = True

REM ===== Checkbox selected =====
Else If fcInitSyncDataList(i)('CONTROL') = 'CheckBox' Then

    REM Show the checkbox 'CB_FC1ParaX' and set the tooltip
    MyApp.Controls('Panel').Controls('CB_FC1Para' + Cstr(i+1)).Visible = True
    MyApp.ToolTip.SetToolTip(MyApp.Controls('Panel').Controls('CB_FC1Para' +
        Cstr(i+1)), fcInitSyncDataList(i)('INFOTOOLTIP'))

    REM Get the checked property from the variable with same name.
    REM If the variable does not exist
    REM get the value from the attribute DefaultValue in the XML
    REM file and save it in a variable with
    REM the same name as the checkbox
    If MyApp.myISL.getValueOfTemporaryVariable('CB_FC1Para' +
        Cstr(i+1),'Local','noVariableFound') = 'noVariableFound' Then
        CType(MyApp.Controls('Panel').Controls('CB_FC1Para' +
            Cstr(i+1)),System.Windows.Forms.Checkbox).Checked =
            fcInitSyncDataList(i)('DEFAULTVALUE')
        MyApp.myISL.setTemporaryVariable('CB_FC1Para' +
            Cstr(i+1),'Local',fcInitSyncDataList(i)('DEFAULTVALUE'))
    Else REM Variable exists, load saved value
        CType(MyApp.Controls('Panel').Controls('CB_FC1Para' +
            Cstr(i+1)),System.Windows.Forms.Checkbox).Checked =
            MyApp.myISL.getValueOfTemporaryVariable('CB_FC1Para' +
                Cstr(i+1),'Local',fcInitSyncDataList(i)('DEFAULTVALUE'))
    End If

REM ===== Combobox selected =====
Else If fcInitSyncDataList(i)('CONTROL') = 'ComboBox' Then
    REM Show the combobox 'CmB_FC1ParaX' and set the tooltip
    MyApp.Controls('Panel').Controls('CmB_FC1Para' + Cstr(i+1)).Visible = True
    MyApp.ToolTip.SetToolTip(MyApp.Controls('Panel').Controls('CmB_FC1Para' +
        Cstr(i+1)),
        fcInitSyncDataList(i)('INFOTOOLTIP'))

    REM Add items to the combobox
    For j As Integer = 1 To fcInitSyncDataList(i)('MAXITEMS')
        CType(MyApp.Controls('Panel').Controls('CmB_FC1Para' +
            Cstr(i+1)),System.Windows.Forms.ComboBox).Items.Add(fcInitSyncDataList(i)
            ('ITEM' + Cstr(j)))
    Next

```

```
REM Get the item to be selected from a variable or from the XML file
If MyApp.myISL.getValueOfTemporaryVariable('CmB_FC1Para' +
Cstr(i+1),'Local','noVariableFound') = 'noVariableFound'
Then
    MyApp.Controls('Panel').Controls('CmB_FC1Para' + Cstr(i+1)).Text =
    fcInitSyncDataList(i)('DEFAULTVALUE')
    MyApp.myISL.setTemporaryVariable('CmB_FC1Para' +
    Cstr(i+1),'Local',fcInitSyncDataList(i)('DEFAULTVALUE'))
Else
    MyApp.Controls('Panel').Controls('CmB_FC1Para' + Cstr(i+1)).Text =
    MyApp.myISL.getValueOfTemporaryVariable('CmB_FC1Para' +
    Cstr(i+1),'Local',fcInitSyncDataList(i)('DEFAULTVALUE'))
End If
End If
Next
End If
```

3.10 Performante Oberflächenelemente

Wie verbessere ich die Performanz der Oberfläche?

Um die Performanz der Oberfläche zu verbessern, können Sie die folgenden Hinweise berücksichtigen.

Verwendung von Transparenz auf der Oberfläche

Wenn Sie die Hintergrundfarbe der Oberfläche auf transparent stellen, ist diese weniger performant.

Verwenden Sie stattdessen eine Standardfarbe.

Verwendung von Autosize

Wenn Sie die Größe der Controls der Oberfläche von *.NET* automatisch bestimmen lassen, ist diese weniger performant.

Übergeben Sie stattdessen eine feste Größe über das Element *SIZE*.

Verwendung von DestroyForm und ChangeForm

Gehen Sie sparsam mit der Verwendung dieser Funktionen um. Oft können bessere Ergebnisse erzielt werden, wenn die komplette Oberfläche in einem Schritt aufgebaut, und aus dem Quellcode durch ein- und ausblenden von Controls angepasst wird.

Kontakt

A.1 Ansprechpartner

Siemens AG

Digital Factory

Factory Automation

Production Machines

DF FA PMA APC

Frauenauracher Straße 80

D - 91056 Erlangen

Fax.: +49 9131 98 1297

<mailto:tech.team.motioncontrol@siemens.com>

A.2 Internetadressen

Weitere Informationen zu den verschiedenen Themen finden Sie auf folgenden Internetseiten.

Siehe auch

SIMOTION (www.siemens.com/simotion)

SINAMICS (www.siemens.com/sinamics)

MotionControl/Applikationszentrum (www.siemens.com/motioncontrol/apc)

Verpackung (www.siemens.com/packaging)