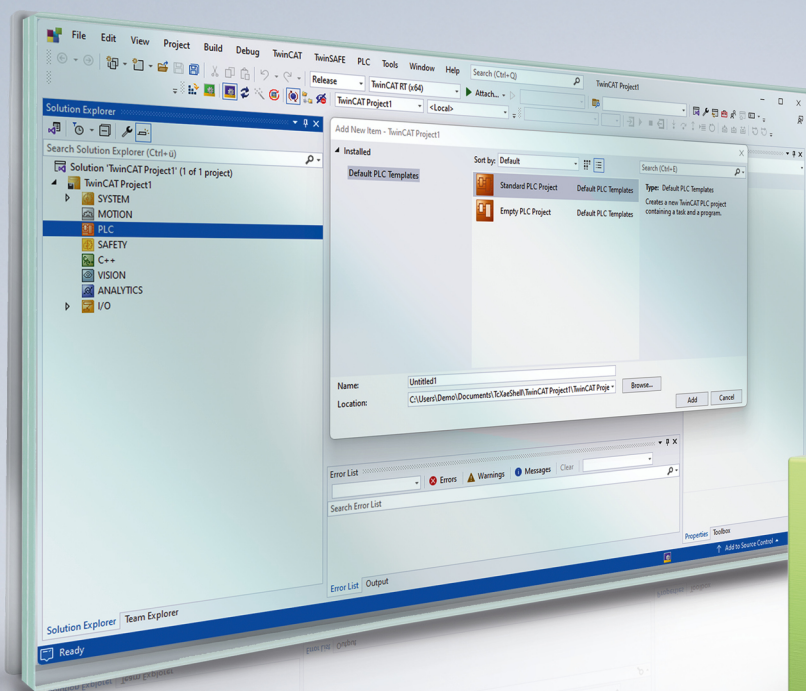


Handbuch | DE

TE1420

TwinCAT 3 Target for FMI



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht.....	8
3	Installation	9
4	Lizenzen	11
5	Quickstart	12
6	Einrichten der Treibersignierung	18
7	Parametrierung der Modulgenerierung.....	21
7.1	Erstellung versionierter Treiber	21
7.2	Erstellen von TMX-Archiven.....	21
7.3	Bündelung mehrerer Modelle in einem TwinCAT-Treiber	22
8	Ausführung von Modulen in TwinCAT	24
8.1	Ausführung als TcCOM Module	24
8.2	Ausführung innerhalb der SPS.....	24
8.3	Exception Handling - Ausnahmebehandlung	24
9	Referenz Benutzeroberfläche	27
9.1	Reiter 'General'.....	28
9.2	Reiter 'Build'	29
9.3	Reiter 'PLC Library'	31
9.4	Reiter 'NewFmuModule1'	31
9.4.1	Reiter 'General'	32
9.4.2	Reiter 'PLC FB'.....	33
9.4.3	Reiter 'TcCOM'	34
10	Kommandozeilen-Interface	38
11	Einbindung weiterer Tools	39
12	FAQ.....	41
13	Beispiele	42
14	Support und Service	43

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Kennzeichnungen führen.



EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland.

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: <https://www.beckhoff.com/trademarks>.

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.

Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

2 Übersicht

Das Functional Mock-up Interface (FMI) ist ein freier Standard zum Austausch bzw. Koppeln von Simulationsmodellen, welche mit verschiedenen Simulationswerkzeugen erstellt wurden. Dies ist häufig der Fall, wenn die Modelle von verschiedenen Zulieferern bereitgestellt werden oder domainspezifische Simulationswerkzeuge zum Einsatz kommen, um die verschiedenen Aspekte einer Maschine oder Anlage zu simulieren (z.B. thermische Simulation, regelungstechnische Simulation).

Ein Simulationswerkzeug, das FMI unterstützt, erlaubt es, ein Modell als Functional Mock-up Unit (FMU) zu exportieren bzw. zu importieren. Dies kann als Model Exchange oder Co-Simulation erfolgen. Bei Ersterem wird das reine Simulationsmodell exportiert. Bei letzterem werden auch die zugehörigen Solver mit exportiert, so dass die Berechnungen des Modells im „Ziel-Simulations-Werkzeug“ analog zum Ausgangswerkzeug geschehen.

Beim Exportieren einer FMU bieten einige Simulationswerkzeuge die Auswahl, das Verhalten des Modells in einer DLL zu kapseln oder / und das Verhalten des Modells als Source-Code zu exportieren. Letztere Option ist zwingend für eine FMU, die in TwinCAT importiert werden soll. Es können also nur FMUs importiert werden, deren Verhalten in der FMU als Source Code vorliegt.

3 Installation

Systemvoraussetzung

Es gelten für das Target for FMI dieselben Anforderungen wie für TwinCAT 3 C/C++. Die detaillierte Beschreibung der TwinCAT 3 C/C++ Anforderungen können Sie im Handbuch TwinCAT 3 C++ Kapitel 4 [Installation](#) nachgucken.

Im Folgenden werden diese nur stichpunktartig und nicht ausführlich aufgegriffen.

Auf dem Engineering-PC

- Microsoft Visual Studio 2017, 2019 oder 2022 Professional, Premium, Ultimate oder Community Edition
 - Bei der Installation von Visual Studio die Checkbox Visual C++ selektieren.
- TwinCAT 3 XAE

Auf dem Laufzeit-PC

- IPC oder Embedded CX PC mit Microsoft Betriebssystem basierend auf „Windows NT Kernel“ (Win 10 und entsprechende embedded Versionen)
- TwinCAT 3 XAR
 - TwinCAT 3.1 unterstützt 32 Bit und 64 Bit Betriebssysteme. Ist das Target ein x64-System, müssen die erstellten Treiber signiert werden. Siehe auch „[x64: Treibersignierung](#)“ im Handbuch TC3 C++

TwinCAT Package Manager: Installation (TwinCAT 3.1 Build 4026)

Eine ausführliche Anleitung zur Installation von Produkten finden Sie im Kapitel [Workloads installieren](#) in der [Installationsanleitung TwinCAT 3.1 Build 4026](#).

Installieren Sie den folgenden Workload, um das Produkt nutzen zu können:

TE1420 | TwinCAT 3 Target for FMI

Setup: Installation (TwinCAT 3.1 Build 4024)

Wenn weder eine der unterstützten Visual Studio Versionen noch ein TwinCAT 3 Setup vorhanden sind, gehen Sie bei der Installation wie folgt vor:

1. Installieren Sie eine der unterstützten Visual Studio Versionen.
2. Laden Sie das TwinCAT 3 Setup auf der Beckhoff Website im Bereich Download herunter. Siehe auch: [Installation bis TwinCAT 3.1 Build 4024](#)
3. Starten Sie das TwinCAT 3 Setup.
4. Folgen Sie den Anweisungen des Installationsassistenten, um TwinCAT 3 zu installieren.
⇒ TwinCAT 3 ist installiert.
5. Starten Sie dann das TE1420-TargetForFMI Setup zur Installation des Target for FMI.
6. Folgen Sie den Anweisungen des Installationsassistenten.
⇒ Target for FMI ist installiert.

Sollten eine Visual Studio sowie eine TwinCAT-Installation bereits vorhanden sein, die Visual Studio Version jedoch nicht den oben genannten Anforderungen entsprechen (z.B. Visual Studio Shell oder Visual Studio ohne Visual C++), gehen Sie wie folgt vor:

1. Installieren Sie eine geeignete Visual Studio Version.
2. Führen Sie das TwinCAT 3 Setup aus.
⇒ TwinCAT 3 wird in die neue oder veränderte Visual Studio Version integriert.

**Treiber-Signierung für Targets mit x64-Betriebssystem**

Falls Sie als Laufzeit-PC ein x64-Betriebssystem nutzen möchten, ist eine Signierung der Treiber notwendig. Details dazu finden Sie im Handbuch TC3 C++ unter [Treibersignierung](#).

4 Lizenzen

Um die gesamte Funktionalität des TE1420 TwinCAT Target for FMI nutzen zu können, sind zwei Lizenzen erforderlich. Zum einen benötigen Sie die Engineering-Lizenz TE1420 zur Erstellung von TwinCAT-Objekten aus einer FMU und zum anderen eine Laufzeitlizenz zur Ausführung dieser Objekte in der TwinCAT-Laufzeit.

Engineering-Lizenz

Die Lizenz **TE1420 Target for FMI** wird auf dem Engineering-System für das Erstellen von TcCOM- und SPS-Bausteinen aus FMUs benötigt. Die Lizenzierung erfolgt also ausschließlich auf dem Engineering-System.



Für dieses Produkt ist keine 7-Tage-Testlizenz verfügbar.

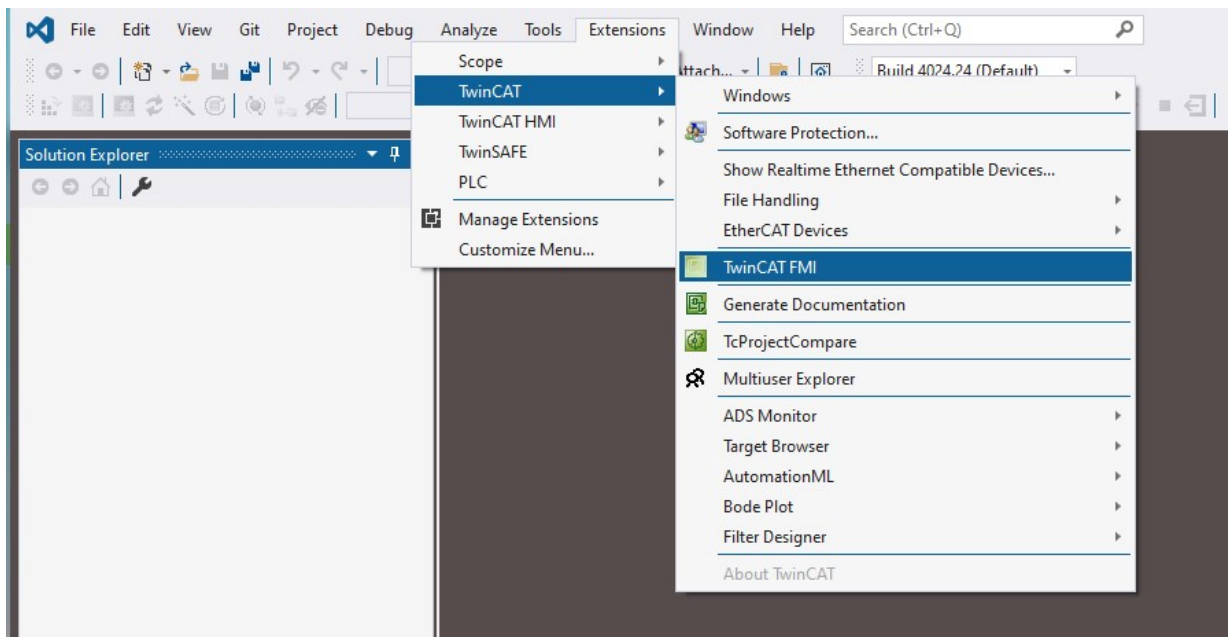
Laufzeitlizenz

Sie benötigen die Lizenz TF1420, um eine TwinCAT-Konfiguration mit einem aus einer FMU generierten TwinCAT-Objekt zu starten. Ohne aktivierte Lizenz können Sie das Modul und damit auch das TwinCAT-System nicht starten. Die Lizenz TF1420 ist eine Erweiterung der C++-Laufzeitlizenzen und benötigt daher eine TC1300 bzw. eine TC1210 mit inkludierter SPS-Lizenz als Grundvoraussetzung.

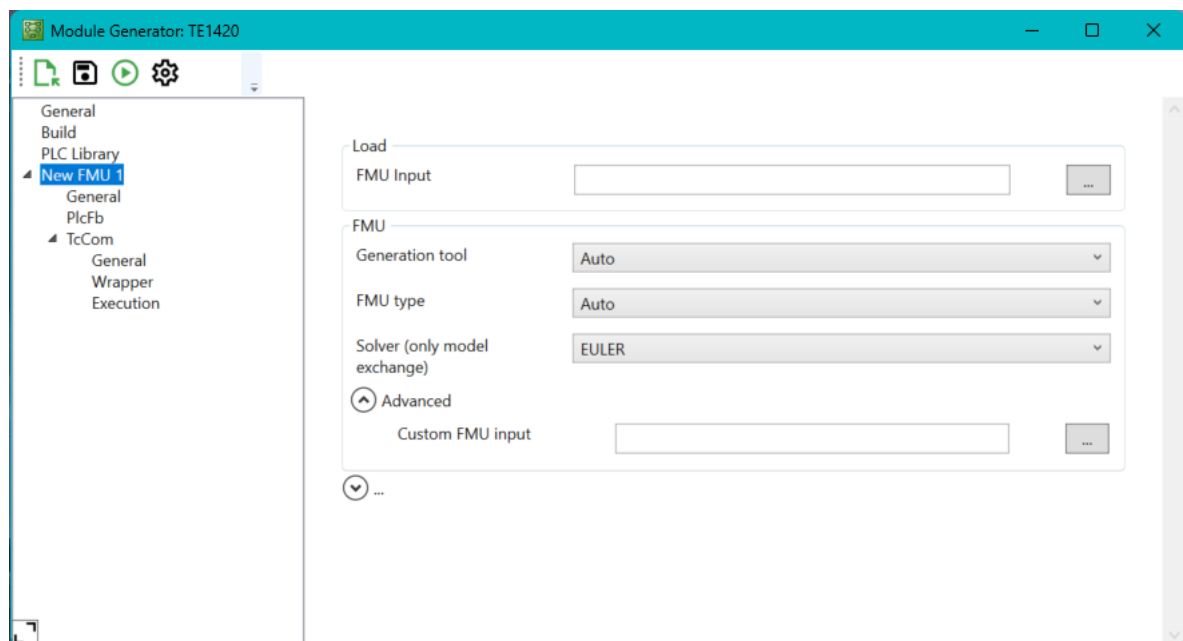
5 Quickstart

Erzeugen eines TcCOM-Moduls aus einer FMU

1. Öffnen Sie die Vollversion eines Visual Studios mit integriertem TwinCAT.
2. Wählen Sie die Option **TwinCAT FMI** im Menü **Extensions -> TwinCAT**.



⇒ Das Fenster **FMI-Interface Tool** öffnet sich.



3. Verwenden Sie den Button ... unter **FMU Input**, um zu der FMU zu navigieren, welche in ein TcCOM-Modul konvertiert werden soll. Sollte die Ansicht nicht wie in der vorangegangenen Abbildung dargestellt öffnet, wählen Sie zuerst den Knoten „New FMU 1“ in der Baumansicht aus.
4. Wählen Sie im Drop-down-Menü **GenerationTool** das Tool aus, aus welchem die FMU exportiert wurde. Ab der Version 2.7.2 des TE1420 wird das Generierungstool automatisch aus der FMU ausgelesen. Unter FMU-Typ steht dann die Auswahl **Auto**.
5. Wählen Sie im Auswahlmeneü **Solver** den Solver aus, der für die Berechnungen des Simulationsmodells der FMU verwendet werden soll.
Wurde das Modell als Co-Simulation exportiert, dann ist der Solver bereits Teil der FMU. In diesem Fall verwenden Sie hier die Auswahl **COSIMULATION**.

6. Nutzen Sie den Button **Start**, um die Generierung des TcCOM-Modells zu starten.

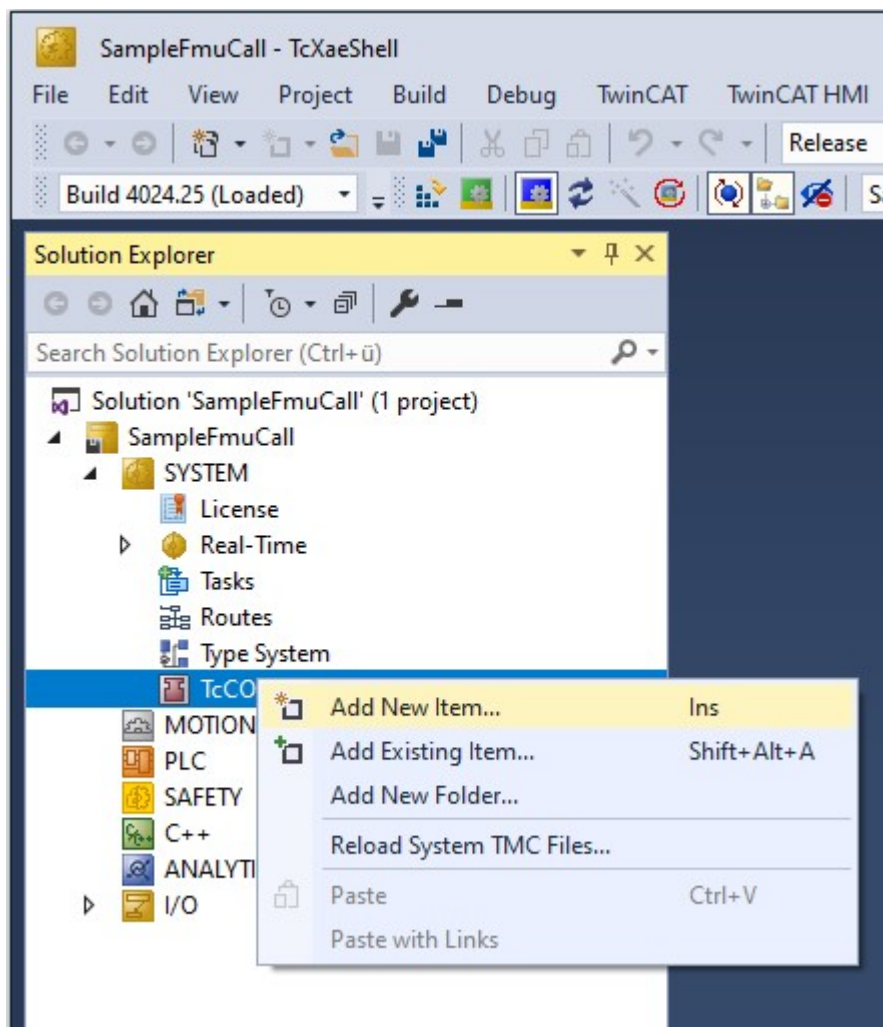
⇒ Bei erfolgreicher Generierung erscheint im Ausgabefenster des TE1420 der Hinweis „Publish procedure completed successfully...“

Hintergrund zur Auswahl eines Generation Tools

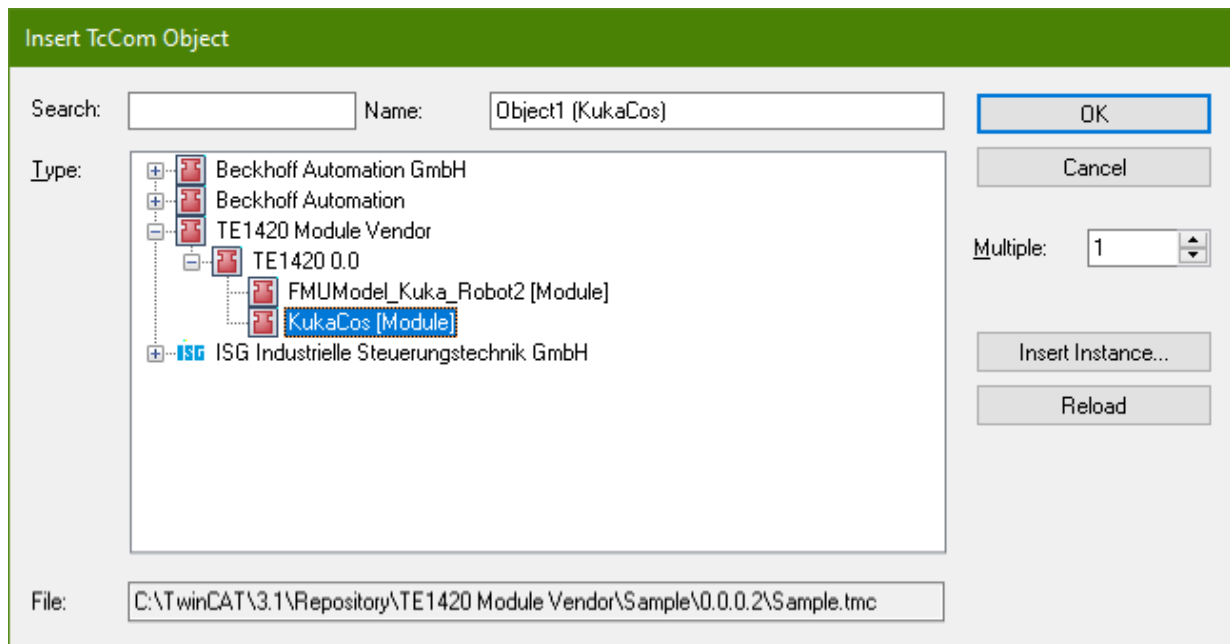
Da das Exportieren des Quellcodes eines Simulationsmodells von den verschiedenen Tools unterschiedlich erfolgt, sind je nach Tool andere Einstellungen im Visual Studio C++-Projekt notwendig, welches bei der Konvertierung automatisch erzeugt und verwendet wird. Für die vom TE1420 bereits unterstützten Tools, erfolgt das Setzen dieser Einstellungen automatisch anhand der Auswahl des Generation Tools. Weitere Tools können auch selbst angebunden werden (siehe Kapitel [Einbindung weiterer Tools](#) [► 39]).

Einbinden des TcCOM-Moduls in ein TwinCAT-Projekt

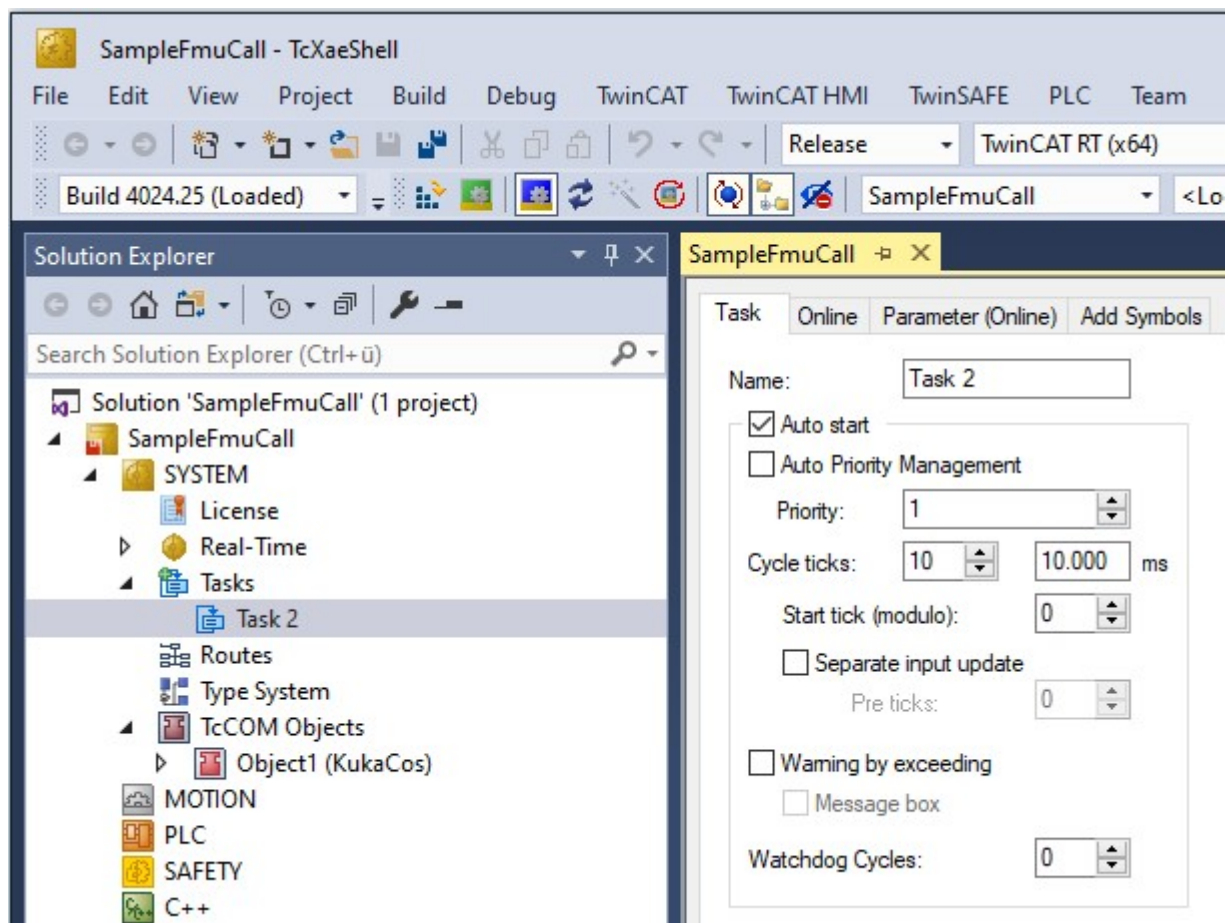
1. Öffnen Sie TwinCAT (TwinCAT XAE oder TwinCAT in einer Visual-Studio-Umgebung).
2. Instanzieren Sie ein neues TcCOM-Objekt.



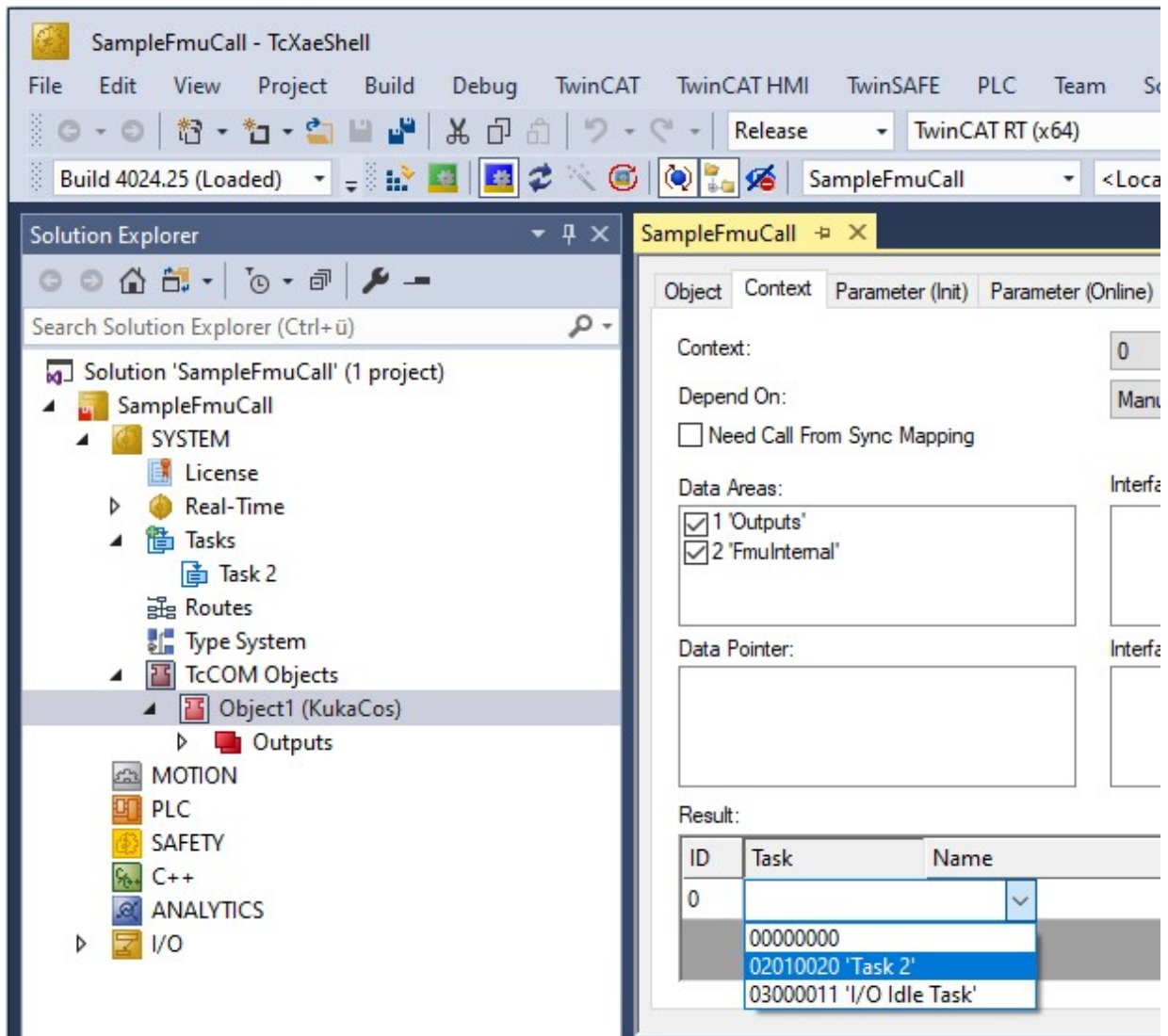
3. Wählen Sie das gewünschte Objekt aus.



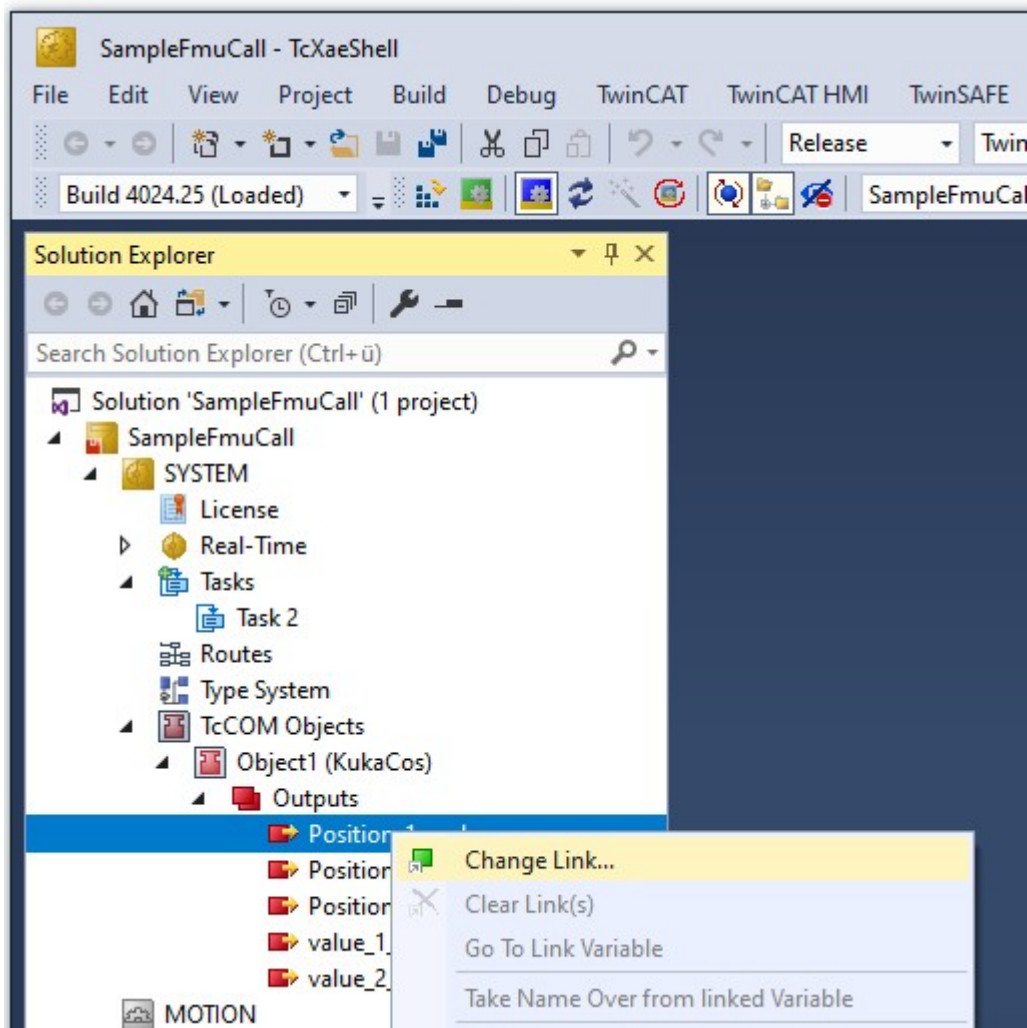
4. Erstellen Sie eine zyklische Task.



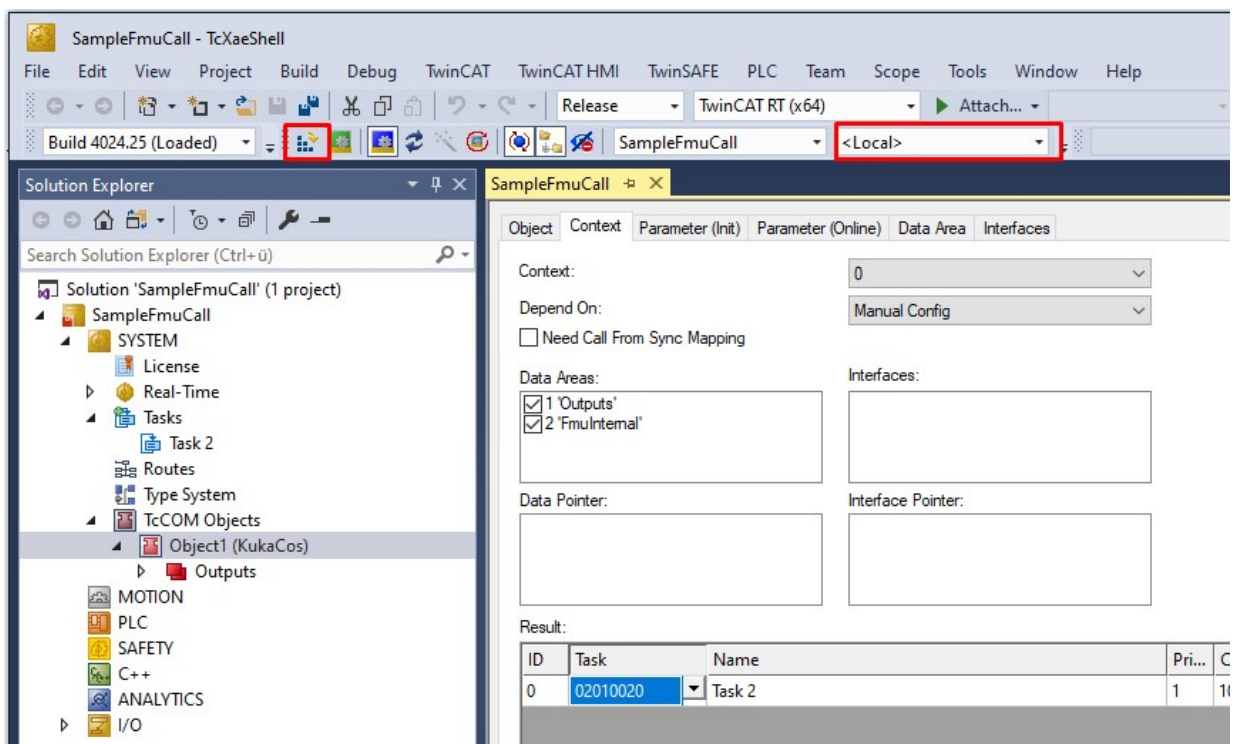
5. Weisen Sie die erstellte Task ihrer TcCOM-Instanz zu.



6. Verknüpfen Sie die Ein- und Ausgänge des Moduls wie erforderlich.



7. Wählen Sie das Zielsystem und aktivieren Sie die Konfiguration.



⇒ Das TcCOM-Modul wurde nun in Ihr TwinCAT-Projekt eingebunden.



FMI Tool Standalone verwenden

Neben der im Visual Studio integrierten Version kann das FMI Interface Tool auch Standalone verwendet werden. Ein installiertes Visual Studio ist auch bei der Verwendung des standalone FMI Interface Tools notwendig.

- Starten Sie die ausführbare Datei FMUConfigurationApp.exe im Verzeichnis *C:\Program Files (x86)\Beckhoff\TwinCAT\Functions\TE1420-TargetForFmi\GUI*, um das FMI Interface Tool Standalone zu verwenden.
 - Fahren Sie danach wie oben beschrieben mit Schritt 3 fort.
-

6 Einrichten der Treibersignierung

OEM-Zertifikat Level 2 erstellen

Die mithilfe des Target for FMI generierten TwinCAT-Objekte basieren, ebenso wie TwinCAT-C++-Objekte, auf einem tmx-Treiber (TwinCAT Module Executable). Die Treiber müssen mit einem OEM-Zertifikat Level 2 signiert werden, damit sie in der TwinCAT-Laufzeit auf dem Laufzeit-PC geladen werden können.

Unter folgenden Links finden Sie eine ausführliche Dokumentation zur Erstellung eines OEM-Zertifikats zur Treibersignierung:

- [Allgemeine Dokumentation zu OEM-Zertifikaten](#)
- [Anwendungsbezogene Dokumentation zu tmx-Treibersignierung](#)

Das Wichtigste in Kürze:

- Sie können selbst ein Zertifikat erstellen. Gehen Sie dazu in Visual Studio auf: **Menu bar > TwinCAT > Software Protection...**
- Sie benötigen ein OEM-Zertifikat Crypto Version 2 (Option: *Sign TwinCAT C++ executables (*.tmx)*).
- Alle Treiber (für 32bit und für 64bit Betriebssysteme) müssen signiert werden.
- Treiber können auch ohne Signierung erstellt und nachträglich signiert werden.
- Für Testzwecke in der Entwicklungsphase genügt ein Zertifikat, das nicht gegensigniert ist.
- Gegensignierte Zertifikate können Sie kostenfrei bei Beckhoff bestellen (TC0008).

Nutzen eines OEM-Zertifikats Level 2 zur Treibersignierung

Es sind drei Varianten zur Signierung von tmx-Treibern denkbar.

1. Sie können auf einem Engineering-PC ein Default-Zertifikat setzen, welches immer für TwinCAT C++ und Target for FMI genutzt wird, sofern Sie kein anderes Zertifikat explizit angeben.
2. Sie können für jeden Build-Vorgang ein Zertifikat explizit benennen.
3. Sie können ohne Zertifikat bauen und nachträglich mit dem TcSignTool signieren.

Für **Variante 1** nutzen Sie eine Windows-Umgebungsvariable. Legen Sie eine neue Umgebungsvariable unter **User > Variables** an, mit:

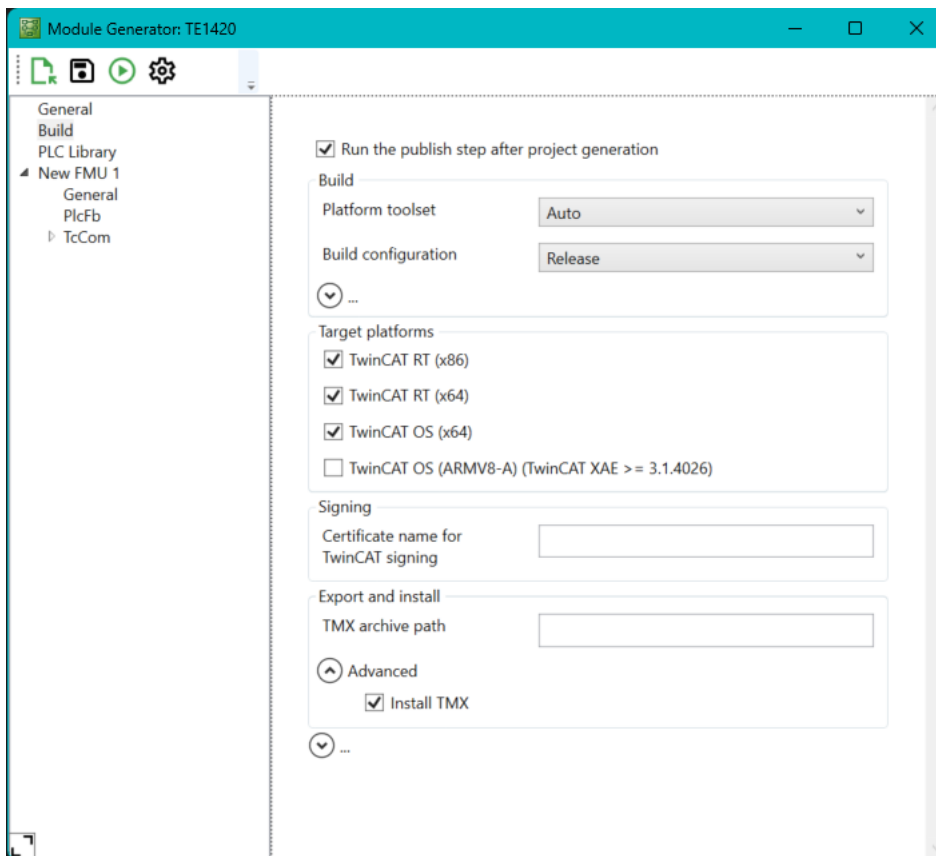
Variable: *TcSignTwinCatCertName*

Value: Name des gewünschten Zertifikates

(Verfügbare Zertifikate liegen unter *TwinCAT\3.1\CustomConfig\Certificates*.)

Für **Variante 2** müssen Sie keine weiteren Einstellungen vorab treffen. Sie können vor jedem Build-Vorgang ein Zertifikat Ihrer Wahl für genau diesen Build-Vorgang definieren.

TwinCAT FMI > Build > Certificate name for TwinCAT signing



Für **Variante 3** können Sie das TcSignTool verwenden. Das TcSignTool ist ein Kommandozeilenprogramm, welches sich im Pfad `C:\TwinCAT\3.x\sdk\Bin\` befindet. Mit `tcsigntool /?` bzw. `tcsigntool sign /?` bekommen Sie Hilfe, wie Sie das Tool konkret benutzen können.

```
TcSignTool sign /f "C:\TwinCAT\3.1\CustomConfig\Certificates\MyCertificate.tccert" /p MyPassword
"C:\TwinCAT\3.1\Repository\TE140x Module Vendor\ModulName\0.0.0.1\TwinCAT RT (x64)\MyDriver.tmx"
```

Für **Variante 1 bis 3** müssen Sie zusätzlich zur Angabe des Zertifikats mit dem TcSignTool das zugehörige Passwort im System hinterlegen. Das Passwort soll aus Sicherheitsgründen nicht im Quellcode eingetragen werden. Mit dem TcSignTool können Sie zu ihren Zertifikaten gehörige Passwörter verschlüsselt in der Registry des Windows-Betriebssystems ablegen.

Die Ablage des Passworts wird mit folgenden Parametern durchgeführt:

```
tcsigntool grant /f "C:\TwinCAT\3.1\CustomConfig\Certificates\MyCertificate.tccert" /p MyPassword
```

Mit folgenden Parametern wird das Passwort gelöscht:

```
tcsigntool grant /f "C:\TwinCAT\3.1\CustomConfig\Certificates\MyCertificate.tccert" /r
```

Die Ablage des unverschlüsselten Passworts erfolgt unter:

```
HKEY_CURRENT_USER\SOFTWARE\Beckhoff\TcSignTool\
```

Verhalten der TwinCAT-Laufzeit

Wird ein mittels des Target for FMI erstelltes TwinCAT-Objekt mit signiertem Treiber in einer TwinCAT-Solution genutzt und mit **Activate Configuration** auf ein Zielsystem geladen, beachten Sie Folgendes:

Jede TwinCAT-Laufzeit (XAR) hat eine eigene White-List mit vertrauenswürdigen Zertifikaten. Steht das Zertifikat, welches zur Signierung genutzt wurde, nicht auf dieser White-List, wird der Treiber nicht geladen. Im TwinCAT Engineering (XAE) wird eine entsprechende Fehlermeldung ausgegeben.

```
7/2/2021 12:31:12 PM 466 ms | 'Port_851' (851): Could not link external function
<extref>FB_BASESTATISTICITERATIVE_GUID_960333F6_F2EA_1737_9AD1_D861CDB4708A__MAIN</extref>
7/2/2021 12:31:12 PM 466 ms | 'Port_851' (851): Could not link external function
<extref>FB_BASESTATISTICITERATIVE_GUID_960333F6_F2EA_1737_9AD1_D861CDB4708A__FB_INIT</extref>
7/2/2021 12:31:12 PM 288 ms | 'TCOM Server' (10): OEM 'MyTestCert'-[REDACTED] certificate currently not trusted. Import 'C:\TwinCAT\3.1\Target
\OemCertificates\[REDACTED]_b86f70ff-06d7-7c09-b29a-da6a4a26d400.reg' to add OEM to trusted list
7/2/2021 12:31:12 PM 466 ms | 'Port_851' (851): Could not link external function
<extref>FB_BASESTATISTICITERATIVE_GUID_960333F6_F2EA_1737_9AD1_D861CDB4708A__FB_EXIT</extref>
7/2/2021 12:31:12 PM 288 ms | 'TCOM Server' (10): Loading 'C:\TwinCAT\3.1\Boot\Repository\TE140x Module Vendor\Tc3_BaseStatistics\0.0.0.1
\Tc3_BaseStatistics.tmx' failed
```

Die Fehlermeldung enthält die Anweisung, ein Registry File, welches auf dem Zielsystem automatisch erstellt wurde, auf dem Zielsystem als Administrator auszuführen. Dieser Vorgang fügt das genutzte Zertifikat der White-List hinzu.

● Registry File ist nur vom OEM-Zertifikat abhängig

i Das Registry File kann ebenso auf weiteren Zielsystemen genutzt werden. Es enthält nur Informationen über das genutzte OEM-Zertifikat und ist nicht zielsystemabhängig.

Wenn Sie ein OEM-Zertifikat, das nicht gegensigniert ist, zur Signierung nutzen, müssen Sie zudem ihr Zielsystem in den Test-Modus versetzen. Führen Sie dazu den folgenden Befehl als Administrator auf dem Zielsystem aus:

```
bcdedit /set testsigning yes
```

Wenn Sie ein gegensigniertes OEM-Zertifikat nutzen, ist dieser Schritt nicht notwendig.

7 Parametrierung der Modulgenerierung

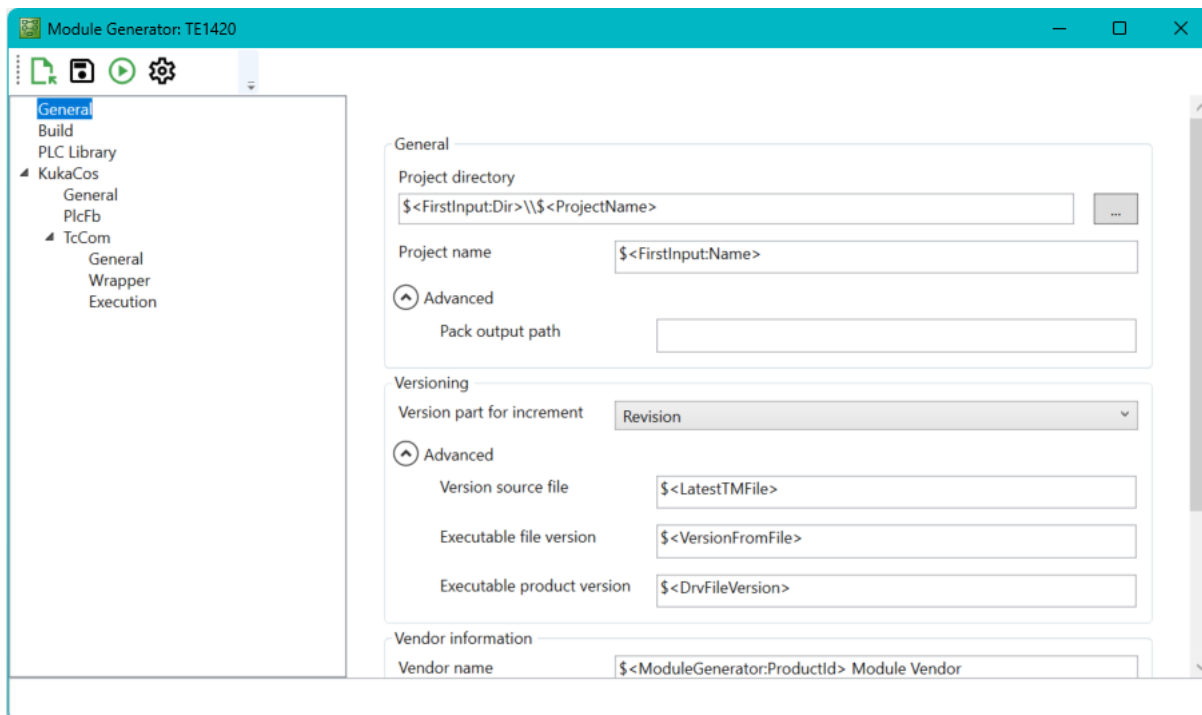
Für das Importieren von FMUs stehen eine Reihe von Einstellungen zur Verfügung, welche die Generierung eines TwinCAT-Laufzeitobjekts (TcCOM-Modul / SPS-Funktionsbaustein) beeinflussen. Auf diese soll in diesem Kapitel eingegangen werden.

7.1 Erstellung versionierter Treiber

Jedes aus einer FMU in TwinCAT 3 importierte Laufzeitobjekt enthält automatisch Versionsinformationen. Damit ist es möglich, mehrere Versionen desselben Modells in TwinCAT-Laufzeitobjekte zu überführen. Diese können dann in TwinCAT versionsselektiv instanziiert werden.

Einstellung der Versionierung

Unter dem Reiter **General** können Sie bei den Einträgen **Version source file** und **Version part for increment** einstellen, wie die Versionierung der Laufzeitmodule erfolgen soll.



Die Basisversion auf Basis derer ein Versionsupdate erstellt werden soll, wird über **Version source file** angegeben. Im Standardfall ist dort `$<LatestTMFile>` angegeben. Dadurch wird auf dem lokalen Engineering-PC nach der letzten verfügbaren Version des Modells gesucht und diese dann als Basis für den Versionsinkrement verwendet. Die Versionsnummer besteht aus vier Digits, z. B. 1.0.3.2 oder 2.12.123.14. Jede Stelle kann separat nach dem Schema `<Major>.<Minor>.<Build>.<Revision>` hochgezählt werden. Wird *None* gewählt, erfolgt kein Versionsupdate und die letzte Version auf dem Engineering-PC wird überschrieben.

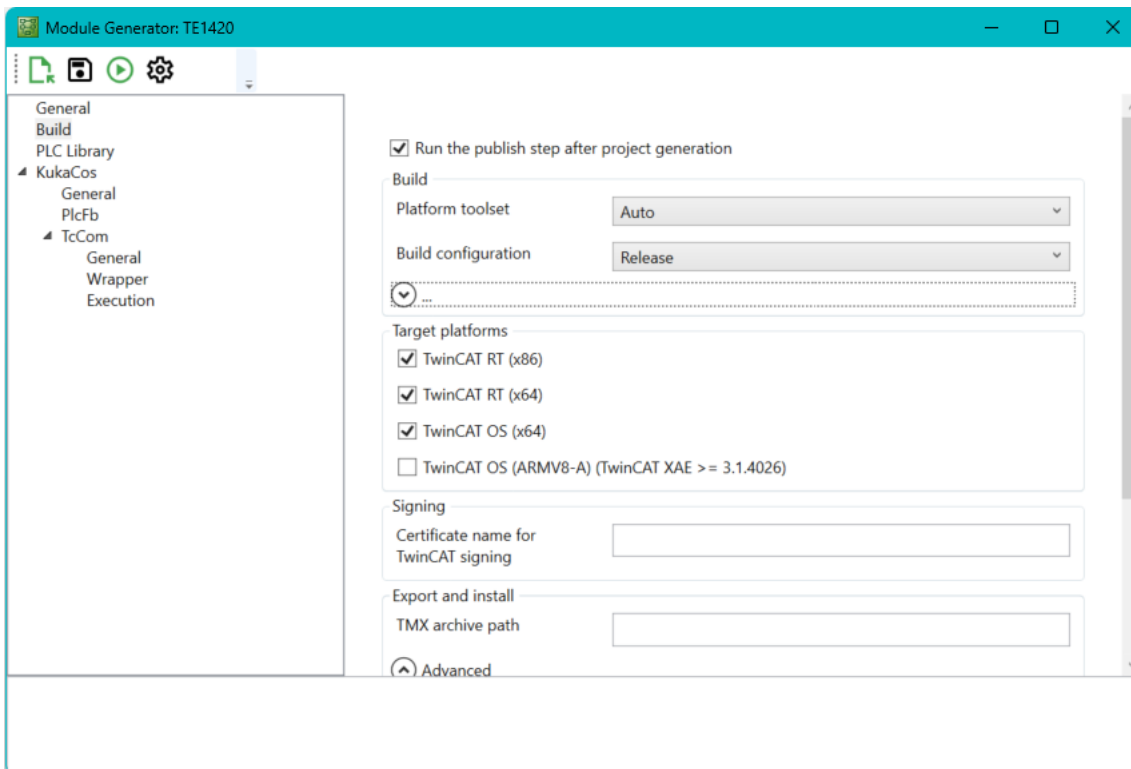
Über **DrvFileVersion** können Sie eine Version fest vorgeben. Geben Sie dafür die Zielversion im Eingabefeld an.

7.2 Erstellen von TMX-Archiven

Um die erstellten TwinCAT-Objekte in einem Projekt verwenden zu können, müssen diese auf dem Engineering System in den entsprechenden Repositorien/Ordnern vorliegen. Das entsprechende Kopieren in diese Ordner kann händisch erfolgen, was aber fehleranfällig ist. Die Lösung hierfür ist die Erstellung eines TMX-Archives. Dies ist ein selbstentpackendes Exe-Archiv, welches die Dateien in die entsprechenden Verzeichnisse kopiert, so dass diese durch die TwinCAT Engineering Umgebung gefunden werden können.

Vorgehen:

1. Öffnen Sie den Reiter **Build**.



2. Tragen Sie im Feld **TMX Archive** den Pfad und den Namen des TMX-Archives ein, welches mit dem nächsten Build erstellt werden soll.

⇒ Im angegebenen Verzeichnis finden Sie ein selbst entpackendes Archiv, welches die tmx-Dateien enthält.

Sie können bei der Pfad- und Namensangabe auch Platzhalter verwenden:

z.B.: C:\temp\[Date]-[Time]-[LibName][LibVersion].exe

Damit wird z.B. ein TMX-Archiv *2022-06-27-150403-Sample0.0.0.7.exe* erzeugt, welches dann an eine beliebige Stelle auf einem Engineering System kopiert und ausgeführt werden kann.

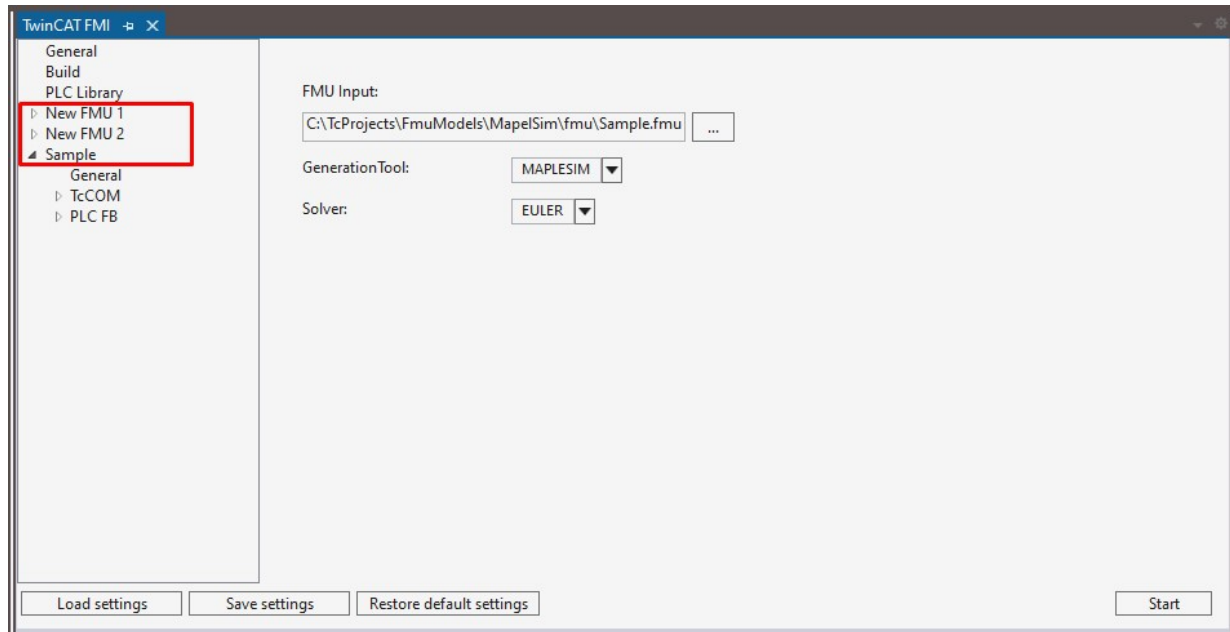
7.3 Bündelung mehrerer Modelle in einem TwinCAT-Treiber

Mithilfe des TE1420 ist es möglich, mehrere Modelle in einem TMX-Treiber bzw. in einer SPS-Bibliothek zu bündeln. Diese Funktion kann zum Beispiel verwendet werden, um eine Sammlung von Modellen (z.B. automatisierungstechnischen Komponenten) in einer Bibliothek zur Verfügung zu stellen.

Gehen Sie hierzu wie folgt vor:

- ✓ Sie haben bereits ein FMI-Projekt mit einer konfigurierten FMU erstellt. Siehe Schritt 2 im Kapitel [Quickstart \[► 12\]](#).
1. Rechtsklicken Sie auf einen freien Bereich im FMI-Projektbau.
 2. Wählen Sie die Option **Add FMU**.

3. Konfigurieren Sie diese FMU wie benötigt und wiederholen Sie diese Schritte so lange, bis alle FMUs, die sie benötigen, im FMI-Projekt enthalten und konfiguriert sind.



4. Betätigen Sie die Taste **Start**.

⇒ Die gewünschten FMUs werden gebündelt.



Beim Export von FMUs verwenden einige Export-Tools intern statische Variablen. Aus diesem Grund können für solche Tools nicht mehrere FMUs aus demselben Tool in einen Treiber oder eine Bibliothek gepackt werden. Eine Übersicht der Tools, die die Bündelung nicht unterstützen, finden Sie im Kapitel [FAQ](#) [► 41].

8 Ausführung von Modulen in TwinCAT

Im folgenden Kapitel wird beschrieben, auf welche Arten Module in TwinCAT ausgeführt werden können.

8.1 Ausführung als TcCOM Module

Die Ausführung eines mithilfe des Target for FMI erzeugten TcCOM-Moduls erfolgt so wie im Kapitel [Quickstart \[► 12\]](#) beschrieben.

8.2 Ausführung innerhalb der SPS

Mithilfe des Target for FMI ist es nicht nur möglich, ein TcCOM-Modul zu erzeugen, das direkt im TwinCAT-Projekt instanziiert werden kann. Sie können zudem eine SPS-Bibliothek erzeugen, welche es erlaubt, das über das Target for FMI importierte Verhalten eines Modells auch innerhalb der SPS aufzurufen.

Der Aufruf des importierten Modells in der SPS funktioniert wie folgt:

1. Erstellen Sie ein SPS-Projekt.
2. Fügen Sie die generierte Bibliothek dem SPS-Projekt hinzu.

8.3 Exception Handling - Ausnahmebehandlung

Bei der Abarbeitung eines mithilfe des TE1420 generierten TcCOM-Moduls in TwinCAT kann es während der Laufzeit zu Floating Point Exceptions kommen. Dieses kann zum Beispiel passieren, wenn einer Funktion ein unerwarteter Wert (z. B. Anfangswert) übergeben wird. Die Behandlung von solchen Exceptions wird im folgenden Kapitel beschrieben.

Was ist eine Floating Point Exception?

Eine Floating Point Exception tritt dann auf, wenn eine arithmetisch nicht exakt ausführbare Rechenoperation in der Floating Point Unit der CPU beauftragt wird. IEEE 754 definiert diese Fälle: *inexact*, *underflow*, *overflow*, *divide-by-zero*, *invalid-operation*. Tritt einer dieser Fälle auf, wird ein Status Flag gesetzt, welches auf die nicht exakt ausführbare Rechenoperation hinweist. Es wird des Weiteren definiert, dass jede Rechenoperation ein Ergebnis liefern muss, und zwar ein solches, das in der Mehrzahl der Fälle dazu führt, dass man die Exception ignorieren kann.

Beispielsweise ergibt eine Division durch Null $+\infty$ oder $-\infty$. Wird im weiteren Code ein Wert durch ∞ geteilt, ergibt dies Null, sodass keine Folgeprobleme zu erwarten sind. Wird ∞ allerdings multipliziert, oder werden andere Rechenoperationen mit ∞ ausgeführt, sind dies *invalid operations*, deren Ergebnis als Not-a-Number (NaN) dargestellt wird.

Wie reagiert die TwinCAT-Laufzeit bei Exceptions?



TwinCAT C++ Debugger nicht aktiv

Folgende Ausführungen gelten nur für den Fall, dass der C++ Debugger **nicht** auf dem TwinCAT-Laufzeitsystem aktiviert ist. Bei aktiviertem C++ Debugger werden Exceptions vom Debugger abgefangen und können behandelt werden, siehe Debugging.

Standardverhalten

Default-Einstellung in TwinCAT ist, dass bei „divide-by-zero“ und „invalid-operation“ die Ausführung des Programms gestoppt wird und TwinCAT eine Fehlermeldung ausgibt.

Task-Einstellung: Floating Point Exceptions

Diese Default-Einstellung lässt sich auf Ebene einer jeden TwinCAT Task verändern. Wird die Checkbox „Floating Point Exception“ deaktiviert, führt eine Exception **nicht** zu einem TwinCAT-Stopp und es wird **keine** Fehlermeldung ausgegeben. Diese Einstellung ist dann gültig für alle Objekte, die durch diese Task aufgerufen werden. In der Folge müssen Sie in der Applikation darauf achten, dass NaN- und ∞ -Werte entsprechend im Programmcode behandelt werden.

Prüfen auf NaN und Inf

Wird beispielsweise ein NaN per Mapping an ein TwinCAT-Objekt weitergegeben, welches Floating Point Exceptions aktiviert hat, führt eine Rechenoperation mit NaN in diesem Objekt zu einer Exception und in der Folge zu einem TwinCAT-Stopp. Daher muss direkt nach dem Mapping auf NaN bzw. inf geprüft werden. In der SPS stehen dazu entsprechende Funktionen in der Bibliothek Tc2_Uilities zur Verfügung, bspw. LrealIsNaN.

Try-Catch-Anweisung

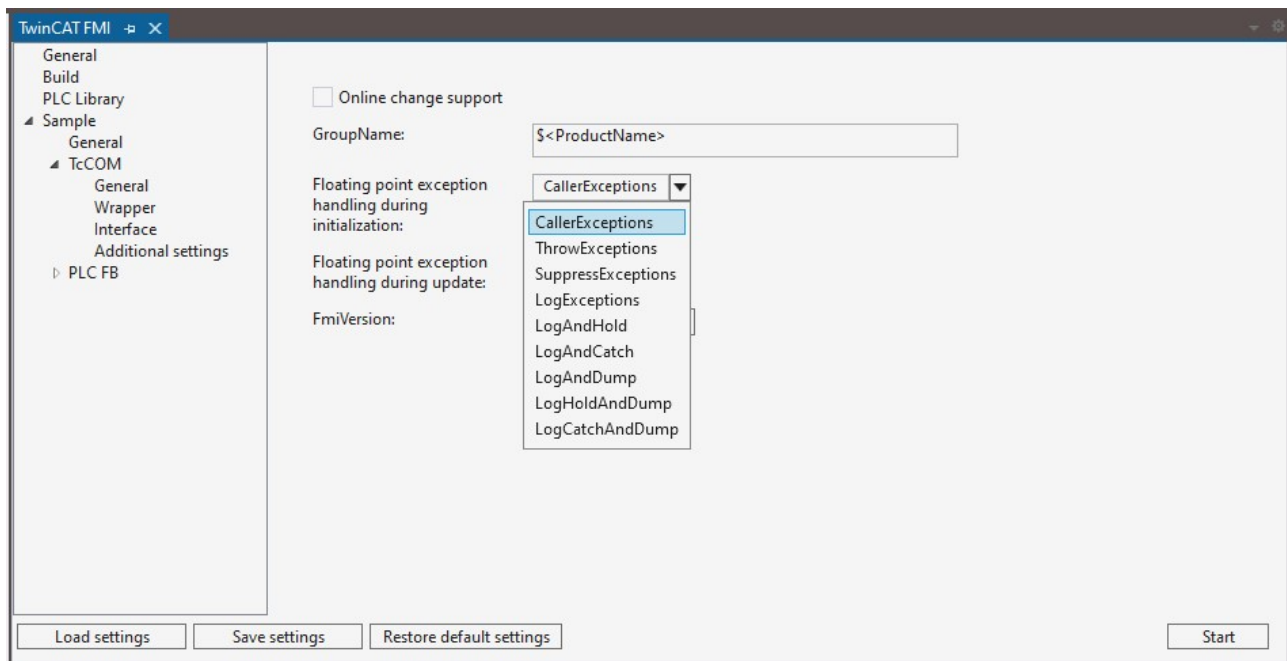
Eine weitere Möglichkeit zum Umgang mit Exceptions ist die Einbettung in eine Try-Catch-Anweisung. In der SPS stehen dazu die Anweisungen TRY, CATCH, FINALLY, ENDTRY zur Verfügung. Sind auf der aufrufenden Task Floating Point Exceptions aktiviert und tritt innerhalb des Try-Catch eine Exception auf, so wird diese im Catch-Zweig gefangen und kann behandelt werden. Entsprechend werden bei dieser Vorgehensweise keine Variablen auf inf oder NaN gesetzt. Wichtig ist aber zu bemerken, dass der Code in Try-Zweig nur bis zu der Stelle der Exception durchlaufen wird und dann ein Sprung zum Catch-Zweig erfolgt. Im Applikationscode ist entsprechend zu beachten, dass interne Zustände im Try-Zweig ggf. nicht konsistent sind.

Dump Files

Ab TwinCAT 3.1.4024.22 (XAR) können zur Laufzeit Dump Files im Falle von Exceptions im TcCOM-Objekt erstellt werden.

Definition des Objektverhaltens bei auftretenden Exceptions

Das Verhalten eines TcCOM-Objekts bei auftretenden Exceptions können Sie unter dem Reiter **TcCOM>General** definieren. Sie müssen dabei das Verhalten getrennt für die Initialisierungsphase des TcCOM und für die Laufzeitphase (update phase) definieren.



Es stehen insgesamt 9 unterschiedliche Einstellungen zur Verfügung:

- **CallerExceptions** (default): Exceptions werden so ausgelöst, wie in der aufrufenden Task konfiguriert.
- **ThrowExceptions**: Exceptions im TwinCAT-Objekt werden in jedem Fall ausgelöst, unabhängig davon wie die Task konfiguriert ist.
 - Eine Exception verursacht eine TwinCAT-Fehlermeldung und einen TwinCAT-Stopp.
- **SuppressExceptions**: Exceptions werden nicht ausgelöst, unabhängig davon wie die Task konfiguriert ist.
 - Eine Exception verursacht keinen TwinCAT-Stopp.
 - Ausgänge oder interne Zustände können NaN oder inf sein.

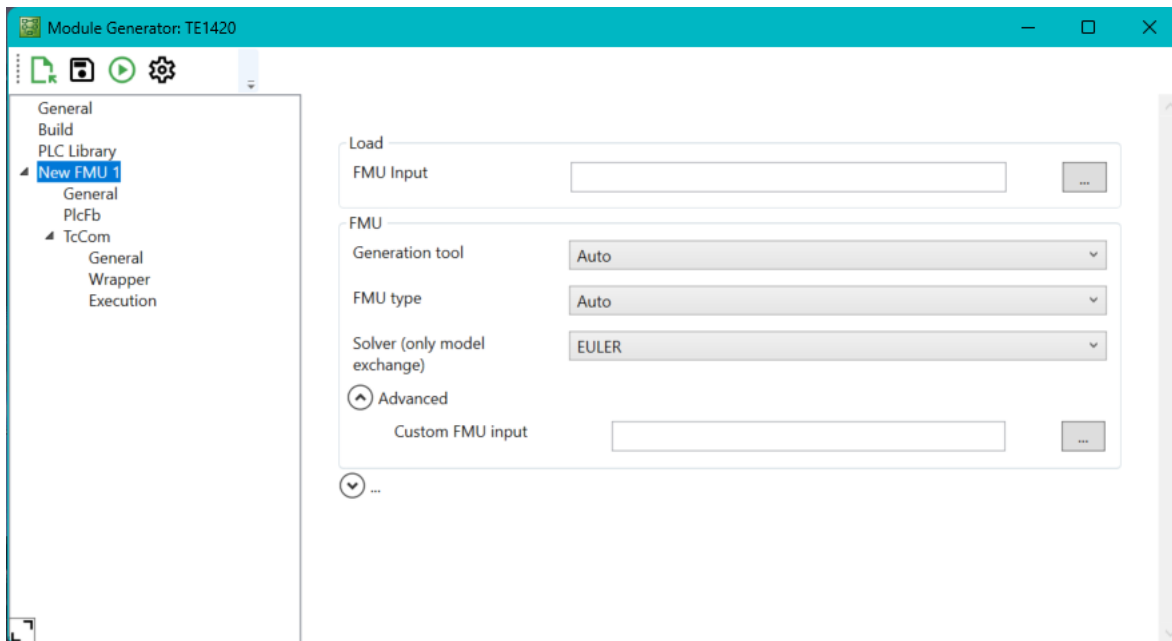
- **LogExceptions:** Exceptions werden ausgelöst, führen aber nicht zu einem TwinCAT-Stopp.
 - Eine Exception verursacht keinen TwinCAT-Stopp.
 - Ausgänge oder interne Zustände können NaN oder inf sein.
 - Der Ausgang `ExecutionInfo` wird mit Informationen über eine Exception im aktuellen Zyklus gefüllt. Treten mehrere Exceptions in einem Zyklus auf, wird nur die erste Exception am Ausgang angezeigt. Beim erneuten Aufruf des TwinCAT-Objekts werden die Informationen zurückgesetzt.
- **LogAndHold:** Exceptions werden ausgelöst. Die Ausführung des TwinCAT-Objekts wird gestoppt.
 - Eine Exception verursacht keinen TwinCAT-Stopp.
 - Ausgänge oder interne Zustände können NaN oder inf sein.
 - Der Ausgang `ExecutionInfo` wird mit Informationen über eine Exception im aktuellen Zyklus gefüllt. Treten mehrere Exceptions in einem Zyklus auf, wird nur die erste Exception am Ausgang angezeigt. Beim erneuten Aufruf des TwinCAT-Objekts werden die Informationen zurückgesetzt.
 - Die Ausführung des TwinCAT-Objekts wird nach Auftreten einer Exception gestoppt. TwinCAT selbst bleibt im Run-Modus.
Ausführung wieder starten: ReleaseObjectStop.
- **LogAndCatch:** Exceptions werden mit try-catch im TwinCAT-Objekt gefangen. Die Ausführung des TwinCAT-Objekts wird gestoppt.
 - Eine Exception verursacht keinen TwinCAT-Stopp.
 - Ausgänge oder interne Zustände können **keine** NaN oder inf enthalten.
 - Der Ausgang `ExecutionInfo` wird mit Informationen über eine Exception im aktuellen Zyklus gefüllt.
 - Die Ausführung des Codes endet an der Stelle der Exception. Von dort wird in den Catch-Zweig gesprungen, d. h. interne Zustände können inkonsistent sein.
 - Die Ausführung des TwinCAT-Objekts wird nach Auftreten einer Exception gestoppt. TwinCAT selbst bleibt im Run-Modus. Ausführung wieder starten: ReleaseObjectStop.
- **LogAndDump, LogHoldAndDump und LogCatchAndDump**
 - Verhalten bei LogExceptions
 - Zusätzlich wird ein Dump File auf dem Laufzeitsystem im TwinCAT-Ordner *Boot* abgelegt. Weiteres zu Dump Files finden Sie im Kapitel Exception Handling in der Dokumentation TE1400 Target for Simulink.

9 Referenz Benutzeroberfläche

Mithilfe des TE1420 Target for FMI können aus FMUs TwinCAT-Laufzeitobjekte (TcCOM-Module bzw. SPS-Funktionsbausteine) generiert werden. Diese werden in einem TwinCAT-Treiber (*.tmx) bzw. in einer SPS-Bibliothek verpackt. Aus einem TwinCAT-FMI-Projekt wird immer genau ein TMX-Treiber und/ oder eine SPS-Bibliothek erzeugt.

Die Benutzeroberfläche gliedert sich daher in einen allgemeinen Bereich, in dem Einstellungen gesetzt werden können, die für alle Laufzeitobjekte in diesem TwinCAT-Treiber/ dieser SPS-Bibliothek gelten und in einen objektspezifischen Teil, welcher die Einstellungen für das jeweilige Laufzeit-Objekt enthält.

Das TE1420 kann auch komplett über die Kommandozeile bedient werden. (Siehe Kapitel [Kommandozeilen-Interface](#) [► 38].) Aus diesem Grund ist zu Verständnisszwecken der jeweilige Kommandozeilenparameter, der einer Einstellung entspricht, jeweils mit in der letzten Spalte der Tabellen aufgeführt.



Menüleiste

Icon	Beschreibung
	Lädt eine FMU oder Konfiguration.
	Speichert die aktuelle Konfiguration.
	Generiert die aktuelle Konfiguration.
	Setzt des Konfigurationslevel.
	Zeigt das Ausgabefenster.
	Versteckt das Ausgabefenster.

Kontextmenü im Projektbaum

Klicken Sie mit der rechten Maustaste auf eine FMU:

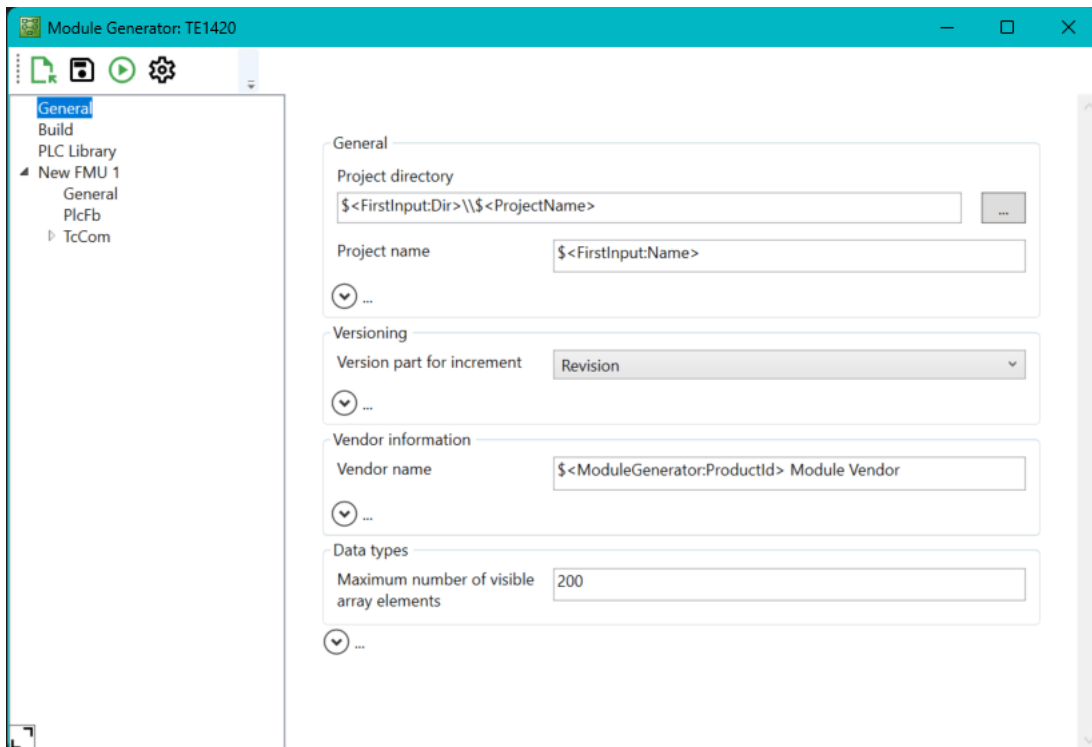
Einstellung	Bedeutung
Remove	Entfernt die FMU aus dem TwinCAT-FMI-Projekt.
Restore defaults	Setzt die Einstellungen des FMU-Knotens auf die Default-Werte. Danach kann die FMU erneut ausgewählt werden.

Klicken Sie mit der rechten Maustaste auf einen freien Bereich im Projektbaum:

Einstellung	Bedeutung
Restore defaults	Stellt für alle FMUs im TwinCAT-FMI-Projekt die Default-Werte wieder her.
Add new FMU	Fügt dem TwinCAT-FMI-Projekt eine weitere FMU hinzu.

9.1 Reiter 'General'

Im Reiter **General** können Sie allgemeine Einstellungen vernehmen. Diese sind standardmäßig mit Platzhaltern versehen und werden mit den Default-Werten belegt. Geänderte Einstellungen werden in einer Xml-Datei persistiert und bleiben auch bei neuen Projekten erhalten. Über die Kontextmenüoption **Restore defaults** können diese wieder zurückgesetzt werden.



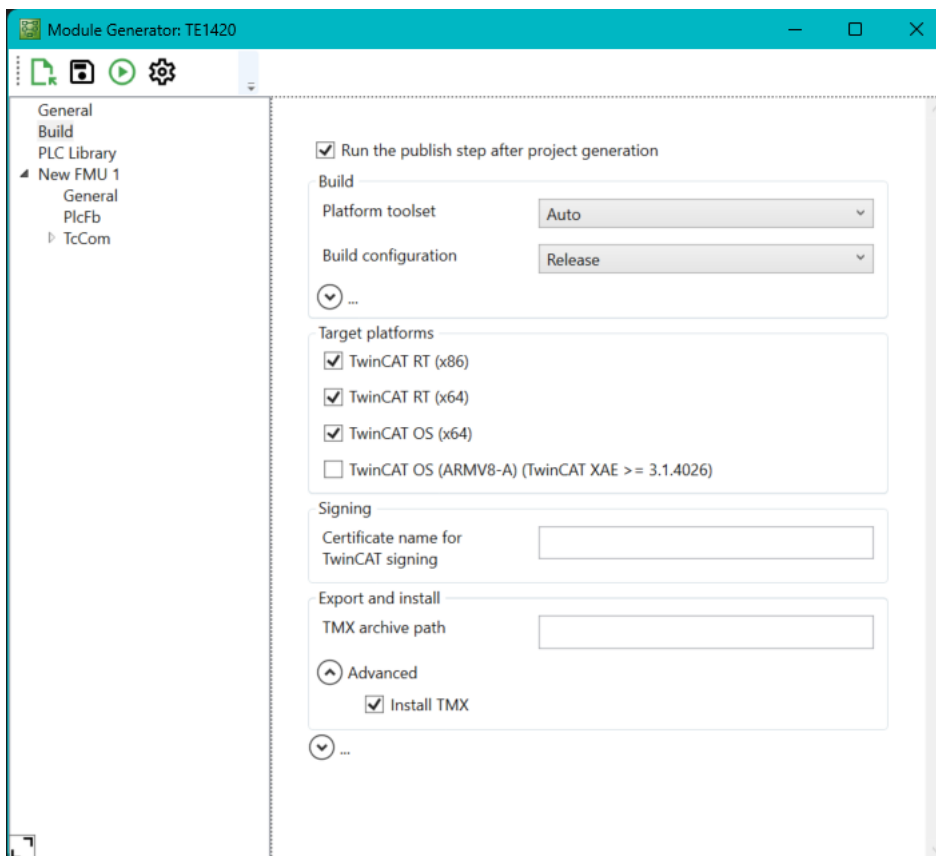
Einstellung	Bedeutung	Kommandozeilenparameter
Project directory	Projekt-Ordner, unter welchem das TcCOM-Modul generiert werden soll.	--Project.ProjectDir
Project Name	Name des Projekts	--Project.ProjectName
Version part for increment	Teil der Versionsnummer, die bei jedem neuen Erzeugen inkrementiert werden soll.	--Project.IncrementVersion Default: Revision
Vendor Name	Herstellernamen, der Name, unter welchem das generierte TcCOM-Modul im Modul-Auswahldialog einsortiert wird.	--Project.VendorName
Maximum number of visible array elements	Maximale Anzahl der sichtbaren Array-Elemente	--Project.MaxVisibleArrayElements Default: 200

Zusätzlich gibt es noch die folgenden erweiterten Parameter, die standardmäßig zugeklappt sind und normalerweise nicht angepasst werden müssen. Hier sind zumeist bereits Templates hinterlegt, welche die Parameterwerte automatisch anpassen.

Einstellung	Bedeutung
Pack output path	Pfad, an welchen das generierte C++Projekt hinterlegt wird.
Version source file	Versionsnummer der Source Code-Datei
Executable file version	Versionsnummer des generierten Moduls oder der Bibliothek
Executable product version	Produktversion des generierten Moduls oder der Bibliothek
Class factory name	Name der TcCOM Classfactory bzw. des Treibers
Product name	Produktname
Copyright notice	Copyright Informationen
Executable discription	Beschreibungstext des Moduls
Additional Options	Möglichkeit zum Überschreiben interner Codegenerator-Parameter
ID of OEM	ID des OEM, wird bei der Verwendung von OEM-Lizenzen benötigt.
IDs of OEM licenses	Liste der OEM-Lizenzen, die beim Aufstarten des Moduls abgefragt werden.

9.2 Reiter 'Build'

Unter dem Reiter **Build** konfigurieren Sie die Build-Einstellungen.



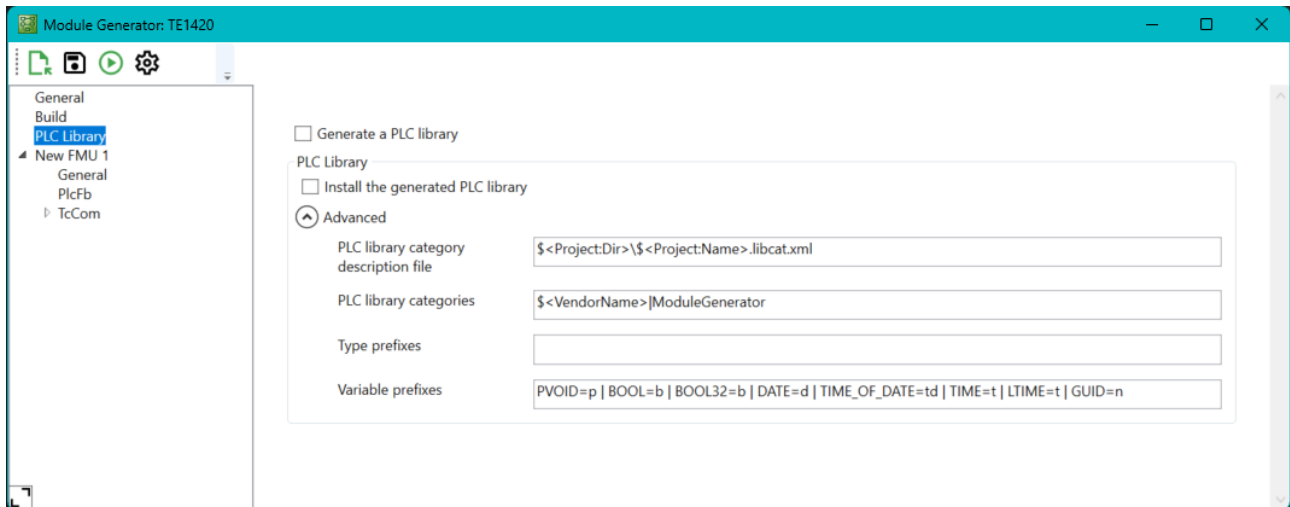
Einstellung	Bedeutung	Kommandozeilenparameter
Run the publish step after project generation	Auswahl, ob nur der Code generiert werden soll oder der tmx-Treiber auch für die ausgewählten Plattformen gebaut werden soll.	--Project.Publish Default: „true“
Platform Toolset	Auswahl des zum Bauen verwendeten Visual Studio Toolsets.	--Project.PublishPlatformtoolset Default: Auto
Build configuration	Auswahl der Build-Konfiguration (Debug/Release).	--Project.PublishConfiguration Default: „Release“
TwinCAT RT(x86)	Erzeugen eines tmx-Treibers für die TwinCAT RT(x86) Plattform.	--Project.PublishTcRTx86 Default: „true“
TwinCAT RT(x64)	Erzeugen eines tmx-Treibers für die TwinCAT RT(x64) Plattform.	--Project.PublishTcRTx64 Default: „true“
TwinCAT OS(x64)	Erzeugen eines tmx-Treibers für die TwinCAT OS(x64) Plattform.	--Project.PublishTcOSx64 Default: „true“
TwinCAT OS(ARMV8-A)	Erzeugen eines tmx-Treibers für die TwinCAT OS(ARMV8-A) Plattform.	--Project.PublishTcOSARMv8A Default: „false“
Certificate name for TwinCAT signing	TwinCAT-Zertifikatname, welcher für die Signierung des erzeugten tmx-Treibers verwendet werden soll.	--Project.SignTwinCatCertName Default: „“
tmx Archive	Name eines tmx-Archives, welches für die Weitergabe der tmx-Dateien verwendet werden kann.	--Project.TmxArchive Default: „“

Zusätzlich gibt es noch die folgenden erweiterten Parameter, die standardmäßig zugeklappt sind und normalerweise nicht angepasst werden müssen.

Einstellung	Bedeutung
Specify C++ language standard version	Auswahl des verwendeten C++ Standards
Code generation and build verbosity	Detailgrad mit der Codegenerierungs- und Baumeldungen im Ausgabefenster geloggt werden.
Always rebuild all source files on publish	Auswählen, damit immer alle Source-Code-Dateien neu gebaut werden.
Build parallel to publish	Wenn mehrere Build-Plattformen selektiert sind, sind die Kompilervorgänge unabhängig voneinander und werden nacheinander in Auftrag gegeben. Durch Aktivieren dieser Option werden alle Build-Prozesse parallel auf Ihrem Engineering System durchgeführt. Die Parallelisierung erfolgt durch das Betriebssystem. Um unnötige Wartezeiten zu vermeiden, sollten Sie immer nur für solche Plattformen Binaries erstellen, die Sie auch wirklich einsetzen.
Install TMX	Installiert das generierte TMX ins Repository.
Deployment project	Absoluter Pfad zum TwinCAT-3-Projekt (*.tsproj), in welchem das Modul automatisch geupdated werden soll.
Activate and restart deployment project	Aktivieren und restarten des Projektes auf dem im Projekt hinterlegten Zielsystem.

9.3 Reiter 'PLC Library'

Unter dem Reiter **PLC Library** stellen Sie ein, ob das FMI-Projekt auch als SPS-Bibliothek erzeugt bzw. auch installiert werden soll.



Einstellung	Bedeutung	Kommandozeilenparameter
Generate a PLC library	Einstellung, ob eine SPS-Bibliothek erstellt werden soll.	--Project.GeneratePlcLibrary Default: „false“
Install the generated PLC library	Einstellung, ob die generierte SPS-Bibliothek auch installiert werden soll.	--Project.InstallPlcLibrary Default: „false“

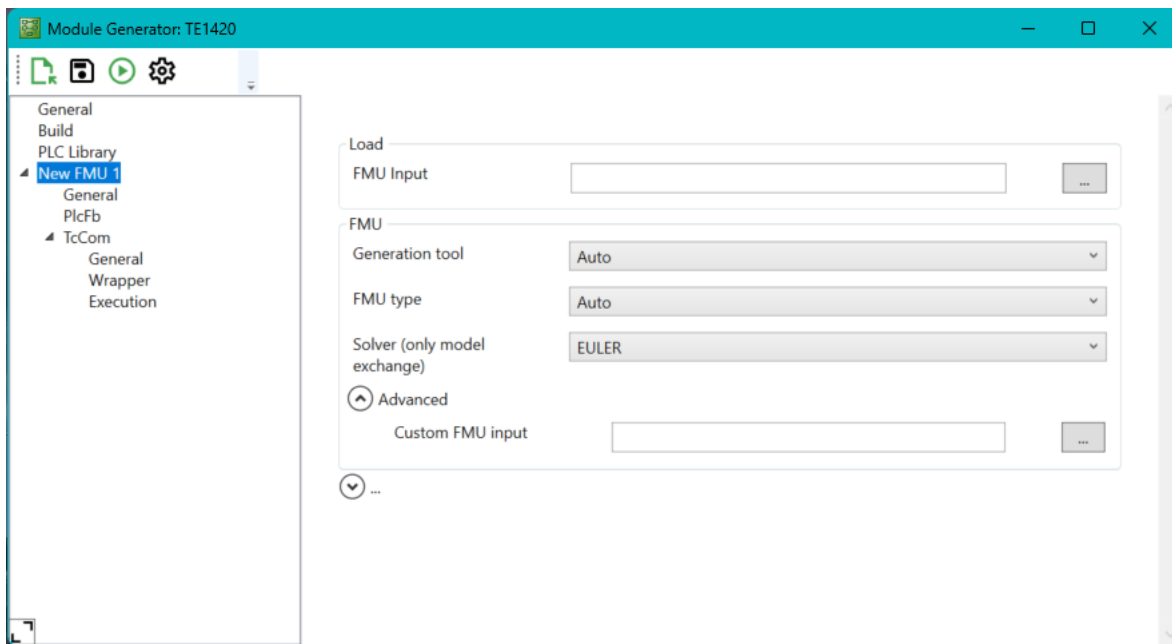
Zusätzlich gibt es noch die folgenden erweiterten Parameter, die standardmäßig zugeklappt sind und normalerweise nicht angepasst werden müssen.

Einstellung	Bedeutung
PLC library category description file	Pfad zur Beschreibungsdatei der SPS-Bibliotheks- Kategorien
PLC library categories	SPS-Bibliothekskategorien der zu generierenden Bibliothek
Type prefixes	Definition von Typ-Prefixen
Variable prefixes	Definition von Variablen-Prefixen

9.4 Reiter 'NewFmuModule1'

In einem FMI-Projekt können eine oder mehrere FMUs in einem tmx-Treiber bzw. in einer SPS-Bibliothek zusammengefasst werden.

Um zu einem Projekt eine neue FMU hinzuzufügen, klicken Sie per Rechtsklick auf einen freien Bereich im Projektbaum. Im sich öffnenden Kontextmenü haben Sie die Möglichkeit, eine FMU aus dem Projekt zu entfernen, den Default-Zustand wiederherzustellen oder eine weitere FMU hinzuzufügen. Im Default-Zustand bzw. wenn eine FMU hinzugefügt wurde, sieht die Oberfläche wie im folgenden Bild aus:

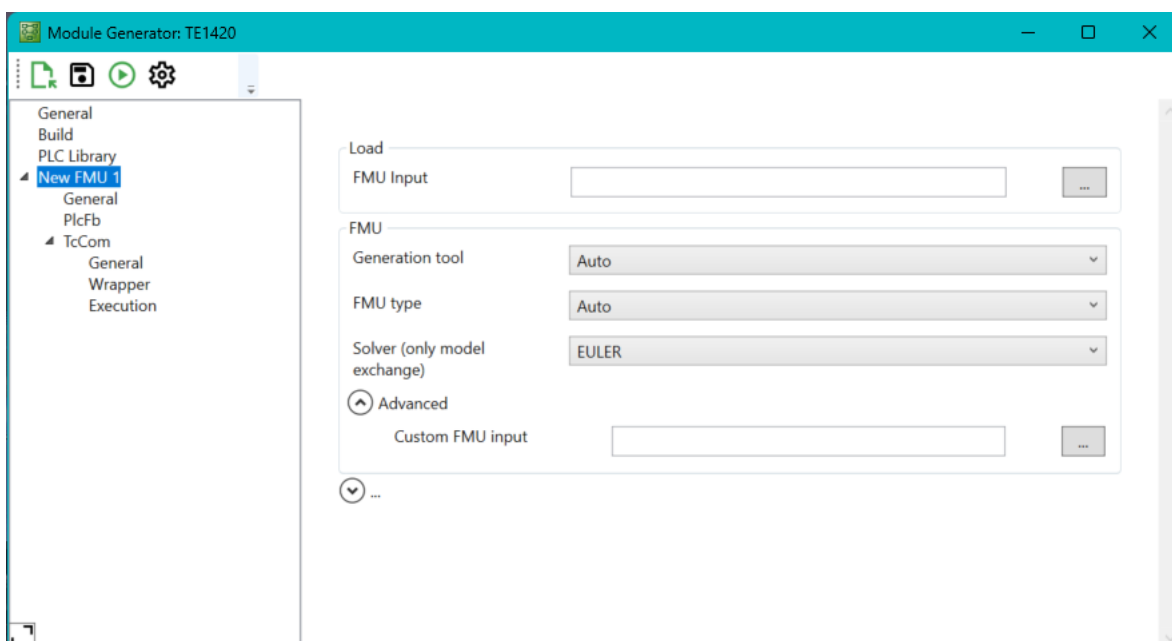


Die vorgenommenen Einstellungen bzw. das zuletzt exportierte FMI-Projekt wird persistiert, so dass bei einem erneuten Exportieren, z.B. nach Änderungen im Modell, diese nicht neu vorgenommen werden müssen.

9.4.1 Reiter 'General'

Unter dem Reiter **General** nehmen Sie allgemeine Einstellungen für die jeweilige FMU vor. Dies ist vor allem die Auswahl der zu importierenden FMU. Die weiteren Parameter wie zum Beispiel das Erstellungstool bzw. der FMU-Typ werden für anhand der Modeldescription.xml-Datei in der FMU automatisch erkannt. Notwendig sind diese Einstellungen, weil die exportierenden Tools nicht immer alle benötigten Informationen in eine FMU mit exportieren bzw. aufgrund von Tool-spezifischen Eigenheiten. Die von der jeweiligen Version unterstützten exportierenden Tools erkennen Sie anhand des Drop-Down-Felds *Generation-Tool*. Sollte das von Ihnen verwendete Tool (welches reine Source Code FMUs erzeugt) hier nicht aufgeführt sein, können Sie, wie unter [Einbindung weiterer Tools](#) [► 39] beschrieben, die Informationen zum Bauprozess in einer Beschreibungsdatei hinterlegen und das Tool somit in das TE1420 einbinden.

Für Model-Exchange-FMUs ist eine weitere notwendige Einstellung der Solver der für die Lösung der Differentialgleichungen verwendet werden soll.



Einstellung	Bedeutung	Kommandozeilenparameter
FMU Input	Pfad zur FMU die importiert werden soll.	InputFile
Generation tool	Auswahl des Tools, welches die FMU erzeugt hat. Bei der Auswahl Auto , erfolgt die Auswahl automatisch anhand der Model-Description.	--Class.GenerationTool Default: „Auto“
FMU Type:	Auswahl des FMU-Typs (Model-Exchange oder Co-Simulation). Bei der Auswahl Auto , erfolgt die Auswahl automatisch anhand der Model-Description.	--Class.FmuVersionType Default: „Auto“
Solver:	Auswahl des Solvers, der verwendet werden soll.	--Class.Solver Default: „Euler“

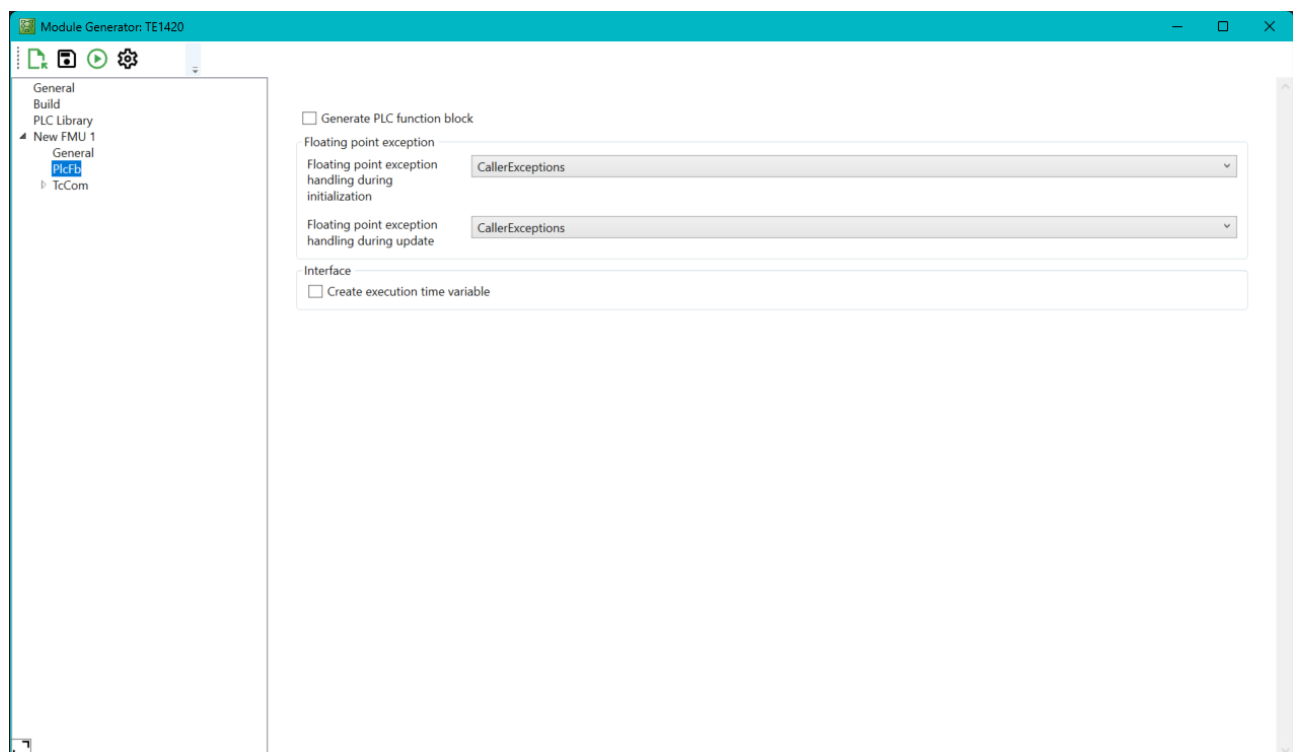
Wird eine FMU per Model-Exchange exportiert, enthält diese nur die entsprechenden Differential-Gleichungen, welche das Modell beschreiben. Nicht enthalten in der FMU sind dann die Solver, um diese Gleichungen zu lösen. Im Auswahlfeld **Solver** stehen Solver aus der internen TwinCAT-Solver-Bibliothek zur Verfügung, welche Sie für die Lösung der Gleichungen auswählen können. Wurde eine FMU per Co-Simulation exportiert, enthält diese bereits den Solver. Im Auswahlfeld **Solver** muss dann **Co-Simulation** ausgewählt werden.

Zusätzlich gibt es noch die folgenden erweiterten Parameter, die standardmäßig zugeklappt sind und normalerweise nicht angepasst werden müssen.

Einstellung	Bedeutung
Custom FMU Import	Pfad zur Beschreibungsdatei für die Einbindung zusätzlicher Tools (siehe Kapitel Einbindung weiterer Tools [► 39]).
Binary type size for input and output (only FMI 3)	Größe des Datentypen binary
String type size	Größe des Datentypen string

9.4.2 Reiter 'PLC FB'

Unter dem Reiter **PLC FB** können weitere Einstellungen vorgenommen werden, welche die Bausteine in der generierten SPS-Bibliothek betreffen. Siehe [Reiter 'PLC Library'](#) [► 31].



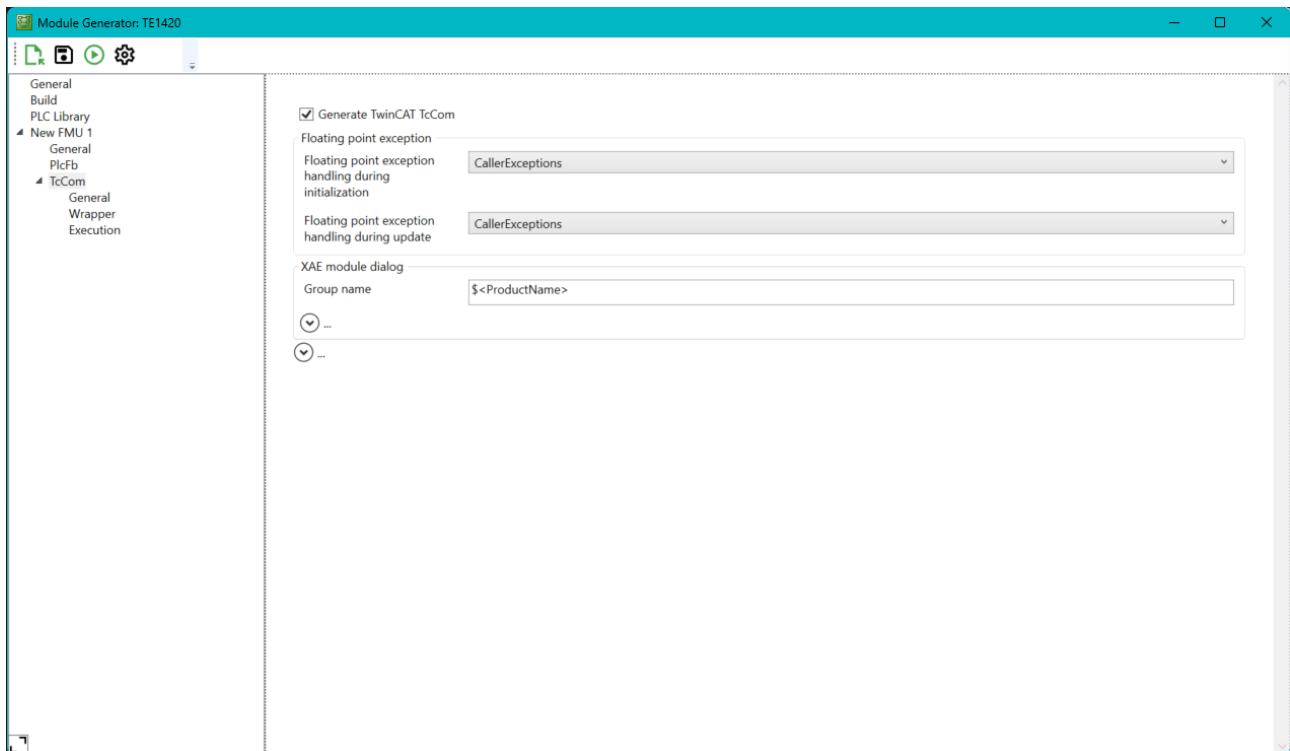
Einstellung	Bedeutung	Kommandozeilenparameter
Generate TwinCAT PLC Function Block	Aktiviert, dass diese FMU (als Funktionsbaustein) Bestandteil der zu generierenden SPS-Bibliothek wird.	--Class.PlcFb.Generate
Floating point exception handling during initialization	Behandlung von Floating Point Exceptions während der Initialisierung des TcCOM-Moduls.	--Class.PlcFb.InitExceptionHandling
Floating point exception handling during update	Behandlung von Floating Point Exceptions während des Aufrufs des TcCOM-Moduls.	--Class.PlcFb.UpdateExceptionHandling
Create execution time variable	Erzeugt eine Variable, welche die gemessene Ausführungszeit anzeigt.	--Class.PlcFb.MonitorExecutionTime

9.4.3 Reiter 'TcCOM'

Unter Reiter **TcCOM** nehmen Sie weitere Einstellungen vor, welche das zu generierende TcCOM-Modul betreffen. Die Grundeinstellungen sind so gewählt, dass in den meisten Fällen auch ohne weitere Einstellungen in diesem Bereich ein ausführbares TcCOM-Modul erzeugt wird.

9.4.3.1 Reiter 'General'

Unter dem Reiter können Sie die allgemeinen Einstellungen des zu generierenden TcCOM-Moduls verändern. Dies betrifft unter anderem den Gruppennamen des Moduls und die Behandlung von Floating Point Exceptions.



Einstellung	Bedeutung	Kommandozeilenparameter
Generate TwinCAT TcCom	Aktiviert, dass für diese FMU ein TcCOM-Modul erzeugt wird.	--Class.TcCom.GenerateTcCom Default: „true“
Floating point exception handling during initialization	Behandlung von Floating Point Exceptions während der Initialisierung des TcCOM-Moduls.	--Class.TcCom.InitExceptionHandling
Floating point exception handling during update	Behandlung von Floating Point Exceptions während des Aufrufs des TcCOM-Moduls.	--Class.TcCom.UpdateExceptionHandling
GroupName	Gruppenname, unter dem das generierte TcCOM-Modul im Auswahldialog TcCOM Object einfügen in der TwinCAT-3-Entwicklungsumgebung einsortiert wird.	--Class.TcCom.GroupName

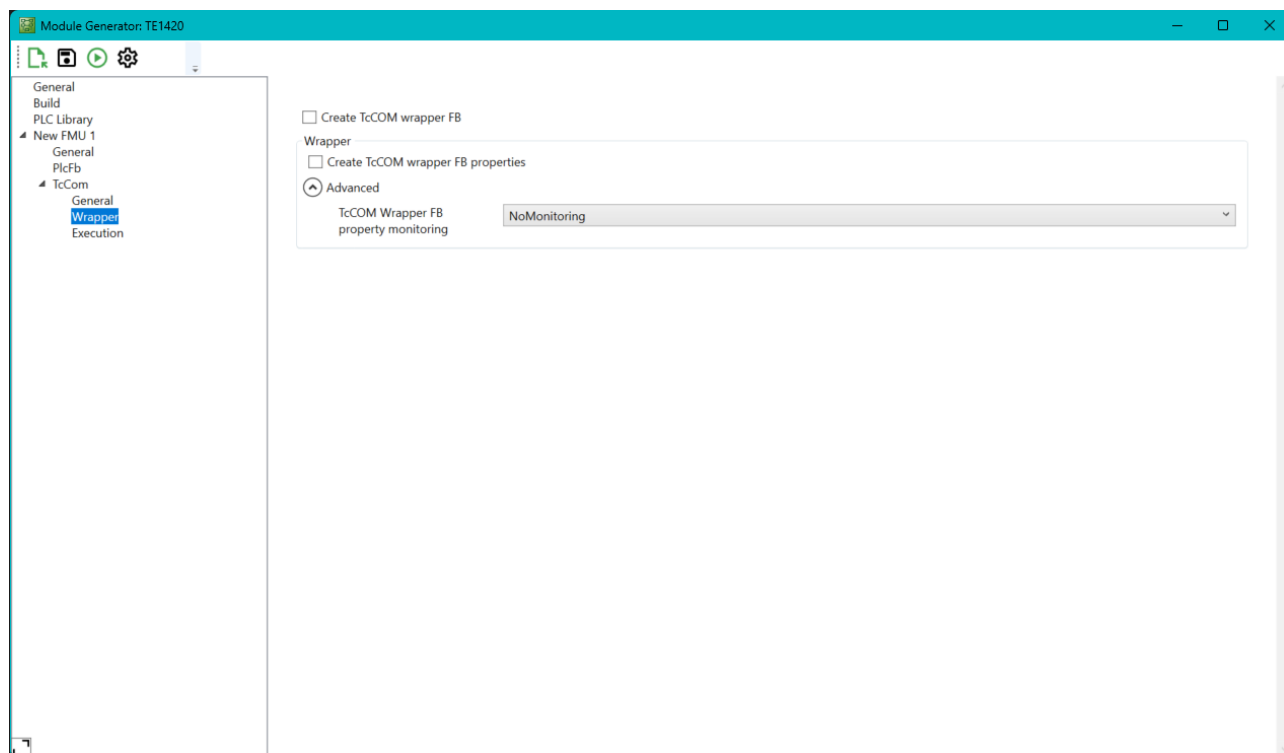
Zusätzlich gibt es noch die folgenden erweiterten Parameter, die standardmäßig zugeklappt sind und normalerweise nicht angepasst werden müssen. Hier sind zumeist bereits Templates hinterlegt, welche die Parameterwerte automatisch anpassen.

Einstellung	Bedeutung
Group Display Name	Gruppenname, unter welchem das TcCOM-Modul im Insert TcCom-Object-Dialog einsortiert wird.
Group icon	Icon der Gruppe unter welcher das TcCOM-Modul einsortiert wird.
Module icon	Icon, welches für das TcCOM-Modul hinterlegt wird. Sichtbar im Insert TcCom-Object-Dialog angezeigt wird
Online Change Support	Schaltet die Online-Change Unterstützung für die generierten TcCOM-Modul ein.
TMC properties	Ermöglicht es zusätzliche Properties in der TMC-Datei zu hinterlegen. Name1=Value1 Name2=Value2 ...
Use stop time	Definiert, ob nach der im Simulationstool definierten und in der Model Description hinterlegten Stop-Time die interne Ausführung des Codes gestoppt wird. Ist dieses Feld nicht aktiv, läuft die Simulation weiter, bis das TcCOM-Modul bzw. die Task gestoppt wird.
Init state transition	Auswahl in welchem State die Initialisierungen im generierten TcCOM-Modul erfolgen. Dies hat Einfluss auf die zur Verfügung stehende Stack-Größe
Create online parameters for initial values	Generiert für die Initialwerte Online-Parameter.
IDs of OEM licenses	Hier können die License IDs für OEM-Lizenzen hinterlegt werden, welche durch das generierte Modul beim Aufstarten abgefragt werden sollen.

9.4.3.2 Reiter 'Wrapper'

Unter diesem Reiter können Sie einstellen, ob zu einem TcCOM-Modul auch ein Wrapper-Funktionsbaustein erzeugt werden soll. Dieser wird im Format PlcOpen XML erzeugt und kann händisch in ein SPS-Projekt importiert werden.

Anders als bei den Einstellungen im Reiter 'PLC Library' [► 31], mit welchen eine SPS-Bibliothek erzeugt wird, in welcher die einzelnen Modelle als Funktionsbausteine repräsentiert werden, erzeugen die Einstellungen im Reiter **TcCOM->Wrapper** einen Funktionsbaustein, der den Aufruf des TcCOM-Moduls „verpackt“. Das eigentliche TcCOM-Modul „lebt“ in diesem Fall außerhalb der SPS und wird nur durch diese aufgerufen bzw. beeinflusst.



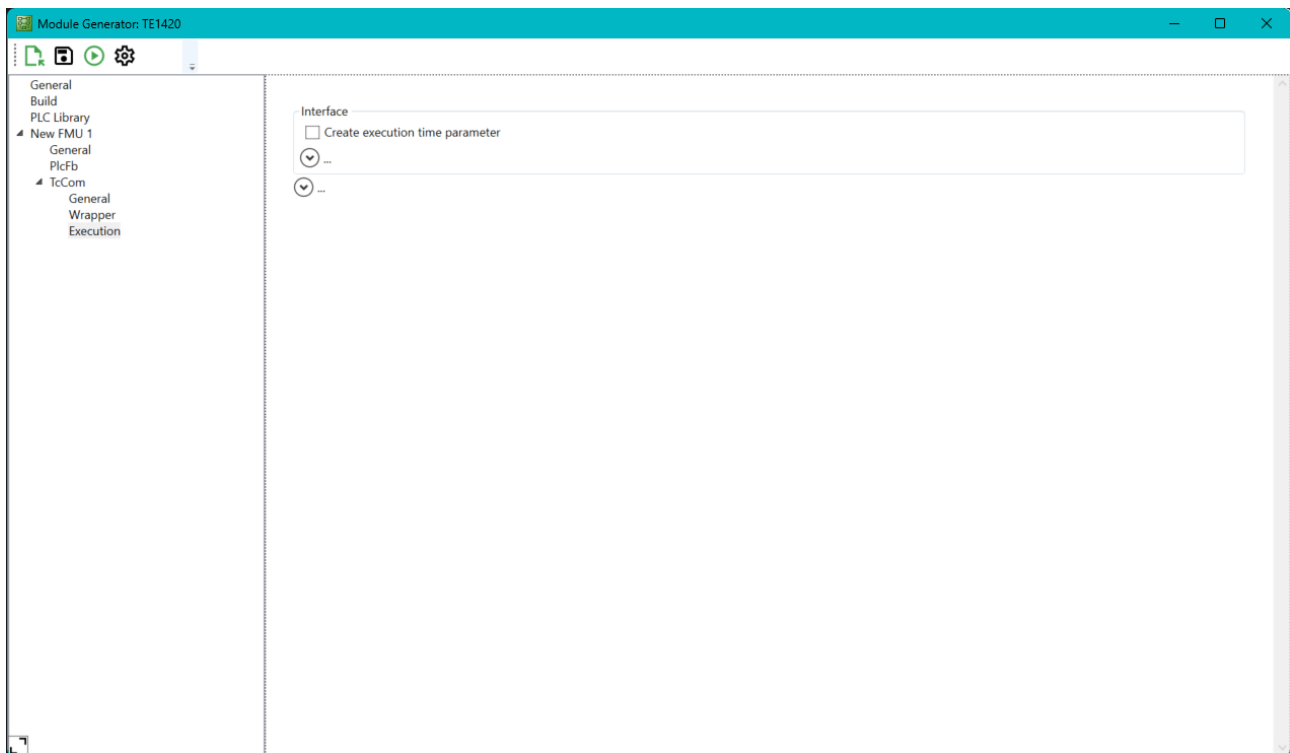
Einstellung	Bedeutung	Kommandozeilenparameter
TcCom Wrapper FB	Erzeugt einen Wrapper FB.	--Class.TcCom.TcComWrapperFb
TcCom Wrapper FB properties	Erzeugt einen Wrapper FB, bei dem die Variablen als Properties zugreifbar sind.	-- Class.TcCom.TcComWrapperFbProperties

Zusätzlich gibt es noch die folgenden erweiterten Parameter, die standardmäßig zugeklappt sind und normalerweise nicht angepasst werden müssen.

Einstellung	Bedeutung	Kommandozeilenparameter
TcCom Wrapper FB property monitoring	Erzeugt eine Variable zum Monitoren der Ausführungszeit.	-- Class.TcCom.MonitorExecutionTime

9.4.3.3 Reiter 'Execution'

Unter diesem Reiter können verschiedene Parameter die Ausführung des generierten TcCom-Modules betreffend, getätigt werden.



Einstellung	Bedeutung
Create execution time parameter	Erzeugt einen Module Parameter im Parameter (Online)-Tab, über welchen die Ausführungszeit ausgelesen werden kann.

Zusätzlich gibt es noch die folgenden erweiterten Parameter, die standardmäßig zugeklappt sind und normalerweise nicht angepasst werden müssen.

Einstellung	Bedeutung
Create ‚ExcutionInfo‘ output	Erzeugt eine variable „ExcutionInfo“ als Ausgang, welche u.a. die Ausführungszeit als Bestandteil der Struktur enthält.
Default module caller	Definiert, worüber das Modul aufgerufen wird (von einem anderen Modul oder einer Task).
Verify caller	Definiert, ob der Caller durch das aufrufende Objekt verifiziert wird.
Default step size adaptation mode	Definiert, welche Zeit für die Schrittweite der Simulationsberechnungen verwendet wird
Default execution sequence	Definiert die Ausführungsreihenfolge der Abarbeitung (z. B. IO am Taskbeginn).
Execute model code after startup	Definiert, ob die Ausführung direkt nach dem Aufstarten begonnen wird (analog zur Autostart-Funktion der SPS).

10 Kommandozeilen-Interface



Die Kommandozeilenschnittstelle steht ab der Version 2.7.2.0 des TE1420 | Target for FMI zur Verfügung.

Das TE1420 Target for FMI bietet eine Kommandozeilen-Schnittstelle, mit deren Hilfe es möglich ist, automatisiert FMUs in TcCOM-Module zu überführen. Diese Schnittstelle wird im Folgenden beschrieben.

Die Verwendung der Kommandozeilen-Schnittstelle erfolgt über die Anwendung TcModuleGenerator.exe, welche sich standardmäßig in den Ordner „C:\Program Files (x86)\Beckhoff\TwinCAT\Functions\TE1420-TargetForFmi\Win32“ installiert.

Der Aufruf per Kommandozeile erfolgt nach dem folgenden Schema:

```
TcModuleGenerator generate [FmuPath1 FmuPath2 ...] [options]
```

Neben einem detaillierten Aufruf, welcher alle Parameter im Aufruf enthält, ist es ebenfalls möglich eine Settings-Datei im Aufruf zu übergeben. In dieser sind die Parameter als XML hinterlegt. Der Aufruf erfolgt dann wie folgt:

```
TcModuleGenerator.exe generate C:\Users\<UserName>\AppData\Local\Temp\configTE14xx.xml
```

Die Settings-Datei kann direkt über das UserInterface erzeugt werden (siehe Kapitel [Referenz Benutzeroberfläche](#) [► 27]; Speichern der Konfiguration).

Die Aufrufparameter der Kommandozeilenschnittstelle können anhand der Hilfe der Schnittstelle selbst über den Befehl

```
TcModuleGenerator.exe --help
```

beziehungsweise

```
TcModuleGenerator.exe generate --help
```

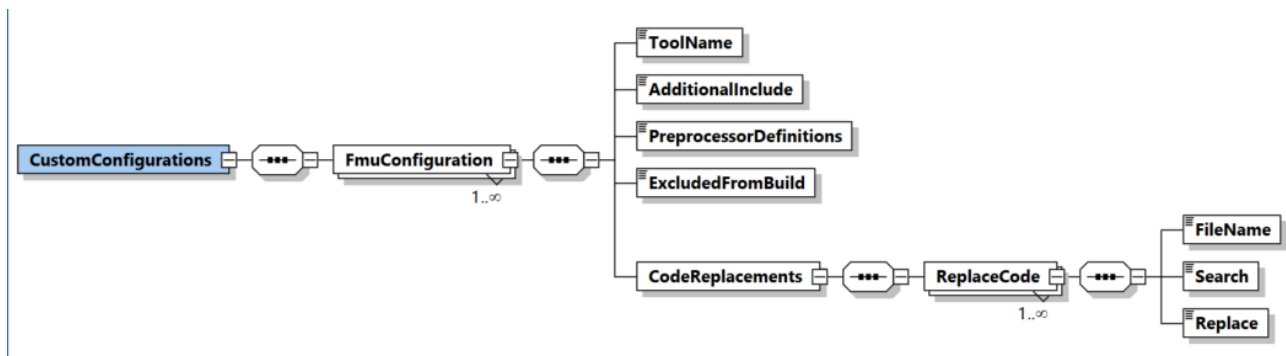
erfragt werden.

Die genauen Aufrufparameter für die einzelnen zur Verfügung stehenden Optionen sind in der Dokumentation “Benutzeroberfläche” angegeben (siehe Kapitel [Referenz Benutzeroberfläche](#) [► 27]).

11 Einbindung weiterer Tools

Um FMUs, die Quellcode enthalten, mithilfe des TE1420 in TcCOM-Module zu überführen, die in einer TwinCAT 3 Laufzeitumgebung ausgeführt werden können, müssen diese zunächst in ein Visual-Studio-Projekt überführt und dann übersetzt werden. Auch wenn der Aufbau des Quellcodes im FMI-Standard vorgeschrieben ist, kann es doch vorkommen, dass für FMUs aus verschiedenen Quellsystemen andere Einstellungen im Visual-Studio-Projekt benötigt werden. Dies können zusätzliche Include-Pfade, Präprozessoreinstellungen bis hin zu Ersetzungen im Quellcode sein. Damit eine Anbindung an weitere FMU exportierende Tools auch von anderen Tool-Herstellern oder Kunden ohne Hilfe von Beckhoff erfolgen kann, müssen die notwendigen Einstellungen in einer Xml-Datei hinterlegt werden. Der Ablageort dieser Dateien ist *C:\ProgramData\Beckhoff\TE1420-TargetForFmi*. In diesem Ordner befindet sich auch eine Template-Datei (*CustomFMU.xml*). Diese kann für die Anbindung weiterer Tools kopiert, entsprechend umbenannt und ausgefüllt werden. Für die Validierung der Datei befindet sich unter *...\TE1420-TargetForFmi\SDK\XML\CustomFmu.xsd* die entsprechende .xsd-Datei.

Xml-Schema der Erweiterungsdatei:



CustomFMU.xml Template:

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<CustomConfigurations
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="CustomFmu.xsd">
  <FmuConfiguration>
    <ToolName>CustomFmu</ToolName>
    <AdditionalInclude>$(ProjectDir);$(ProjectDir)sources;</AdditionalInclude>
    <PreprocessorDefinitions></PreprocessorDefinitions>
    <ExcludedFromBuild></ExcludedFromBuild>
    <CodeReplacements>
      <ReplaceCode>
        <FileName></FileName>
        <Search></Search>
        <Replace></Replace>
      </ReplaceCode>
    </CodeReplacements>
  </FmuConfiguration>
</CustomConfigurations>

```

Name	Bedeutung
CustomConfiguration	Root-Knoten der Konfigurationsdatei
FmuConfiguration	FmuConfiguration für jedes weitere Tool
ToolName	Name des Tools, so wie es unter FMU-Typ angezeigt werden soll (siehe Reiter 'General' [► 34]).
AdditionalInclude	Zusätzliche Include-Pfade separiert durch Kommata.
PreprocessorDefinitions	Präprozessordefinitionen, die beim Bauen der FMU im Visual Studio verwendet werden müssen, separiert durch Kommata.
ExcludedFromBuild	Dateien, die von einem Build ausgeschlossen werden sollen.
CodeReplacements	Oberknoten für alle Ersetzungen im Quellcode, falls welche benötigt werden.
ReplaceCode	Eintrag für jede Ersetzung im Quellcode
FileName	Name der Datei, in welcher eine Ersetzung durchgeführt werden muss.
Search	Code-String, nach dem gesucht werden muss.
Replace	Code-String, der den gesuchten String ersetzen soll.

12 FAQ

Warum kann meine FMU nicht importiert werden?

Für das TE1420 müssen die FMUs mit Source Code erzeugt werden. Diese werden dann mithilfe des TE1420 Target for FMI in ein TwinCAT Runtime-Module (TcCOM-Module) übersetzt.

Welche FMI-Versionen werden unterstützt?

Unterstützt werden sowohl FMI 2.0 als auch FMI 3.0

13 Beispiele

Beispiel-FMUs und zugehörige TwinCAT-3-Projekte können Sie auf der GitHub-Seite von Beckhoff herunterladen: [TE142x - FMU-Samples](#). Die Beispiele und deren Funktionen sind dort ebenfalls entsprechend erklärt.

14 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser Downloadfinder beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157
E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460
E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0
E-Mail: info@beckhoff.com
Internet: www.beckhoff.com

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

DSP System Toolbox, Embedded Coder, MATLAB, MATLAB Coder, MATLAB Compiler, MathWorks, Predictive Maintenance Toolbox, Simscape, Simscape™ Multibody™, Simulink, Simulink Coder, Stateflow and ThingSpeak are registered trademarks of The MathWorks, Inc.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Mehr Informationen:
www.beckhoff.com/te1420

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

