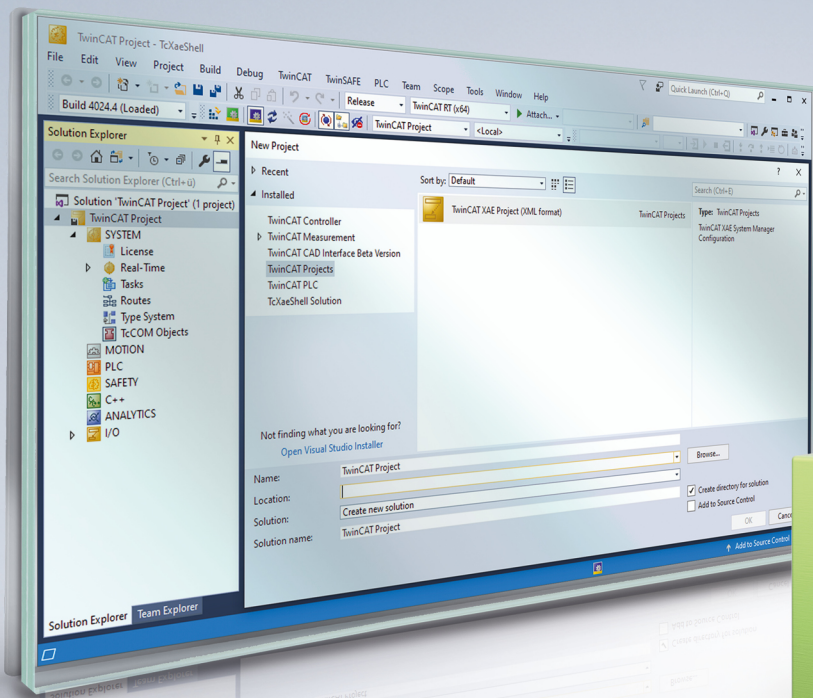


Handbuch | DE

# TE1030

## TwinCAT 3 | Documentation Generation





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit .....	7
<b>2</b>	<b>Übersicht</b> .....	<b>8</b>
<b>3</b>	<b>Installation</b> .....	<b>9</b>
<b>4</b>	<b>Grundlagen</b> .....	<b>10</b>
<b>5</b>	<b>Konfiguration</b> .....	<b>12</b>
<b>6</b>	<b>Vorschau der Dokumentation</b> .....	<b>14</b>
<b>7</b>	<b>Markups</b> .....	<b>17</b>
7.1	Übersicht .....	17
7.2	Beschreibungen der Markups .....	18
7.2.1	Delimiters .....	18
7.2.2	Description .....	18
7.2.3	Summary.....	19
7.2.4	Example .....	19
7.2.5	Header .....	20
7.2.6	List.....	20
7.2.7	Table .....	20
7.2.8	Cross references.....	21
7.2.9	Links.....	21
7.2.10	Code.....	22
7.2.11	Literal .....	23
7.2.12	Preliminary .....	23
7.2.13	Audience .....	23
7.2.14	Param.....	24
7.2.15	Returns.....	25
7.2.16	Note.....	25
7.2.17	Image .....	26
7.2.18	Character formats .....	28
<b>8</b>	<b>SPS-Bibliotheken dokumentieren</b> .....	<b>29</b>
<b>9</b>	<b>Beispiel</b> .....	<b>31</b>
<b>10</b>	<b>Support und Service</b> .....	<b>32</b>



# 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®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® 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 Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

## EtherCAT®

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.

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

## 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

Die Maschinenrichtlinie schreibt vor, dass mit jeder Maschine eine Dokumentation ausgeliefert werden muss und auch das Produktsicherheitsgesetz fordert eine Bedienungsanleitung für verkaufte Produkte. Durch diese gesetzlichen Vorgaben ist die Dokumentation von Maschinen und Produkten heute für jedes Unternehmen notwendig. Dabei kostet die Dokumentation Zeit, muss konsistent sein und soll immer gleich aussehen, auch wenn mehrere Personen an ihr arbeiten.

TwinCAT Documentation Generation ermöglicht die Anfertigung von Dokumentation innerhalb einer Maschinenanwendung oder SPS-Bibliothek. Dafür verwendet das Programm spezifische Markups innerhalb der Quellcode-Kommentare, um relevante Informationen zu markieren und für die Dokumentation zu formatieren. Auch der Verweis auf andere Dokumente, die zusätzliche Informationen enthalten, ist möglich.

Da die Dokumentation direkt aus der Programmierung generiert wird, kann sichergestellt werden, dass das Anwendungsprogramm und die Dokumentation konsistent sind. Zudem ist der zusätzliche Aufwand für die Entwickler moderat, weil die Generierung der Dokumentation auf der Basis von Markups erfolgt. Die Entwickler können in TwinCAT Documentation Generation Inhalte zur Dokumentation beitragen, ohne mit den typischen Autorenwerkzeugen vertraut zu sein.



## 3 Installation

### Systemvoraussetzungen

Technische Daten	Beschreibung
Betriebssystem	Windows 10
Zielplattform	Windows
Minimale TwinCAT-Version	TwinCAT 3.1.4026
TwinCAT-Lizenzen	TE1030 TwinCAT 3 Documentation Generation

### 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:

- TE1030 | TwinCAT 3 Documentation Generation

### Lizenzierung

TwinCAT 3 Documentation Generation ist ein Engineering Produkt. Die Lizenzierung erfolgt ausschließlich auf dem Engineering System. Die Beschreibung der Lizenzierung einer Vollversion finden Sie in der Dokumentation „[TwinCAT-3-Lizenzierung](#)“.

## 4 Grundlagen

Mit speziellen Kommentaren ist der Entwickler, der mit TwinCAT 3 ein Programm schreibt, in der Lage, erklärenden Text direkt in das TwinCAT-Projekt zu schreiben. Durch die Verwendung von speziellen Markups in den Kommentaren kann der Text automatisch ausgewertet werden. Zusätzlich wird auch die Struktur der Software genutzt, um Vererbungsbeziehungen und Abhängigkeiten darzustellen. Ein Inhaltsverzeichnis für die Dokumentation wird automatisch erzeugt. Hierfür werden keine weiteren Markups benötigt.

TwinCAT Documentation Generation verarbeitet mit Hilfe von Markups Dokumentationskommentare in Ihrem Code und formatiert diese zu einer Dokumentation, deren Namen und Speicherort Sie im Fenster **TE1030-Tc3DocGen** angeben. Als Ausgabeformat können Sie unter anderem HTML auswählen. Außerdem generiert Documentation Generation automatisch partielle Klassendiagramme, um die Abhängigkeiten der Code-Elemente zu zeigen.

Die automatische Generierung einer Dokumentation basiert auf den folgenden Informationen innerhalb eines TwinCAT-Projekts:

- Strukturinformationen
- Informationen, die im Quellcodekommentar markiert sind (Markups)
- Verwendete Bibliotheken

### Hierarchie der Elemente

In der Hierarchie der Elemente stehen die Markups Description, Summary und Example oben. Diese drei Markups sind die Voraussetzung für die Verwendung vieler anderer Markups.

Das folgende Beispiel zeigt, wie das Markup Code in das höhergestellte Markup Description eingebettet werden kann:

```
(*!<description>Jetzt folgt ein Codeblock.
```

```
<code> iCount:=iCount+1 </code>
```

```
</description>*)
```

In der generierten Dokumentation sieht das wie folgt aus:

#### Description

Now follows a code block.

```
iCount:=iCount+1
```

Die Dokumentation wird bei der Ausgabe automatisch in einzelne Abschnitte eingeteilt, die diese Hierarchie ebenfalls widerspiegelt:

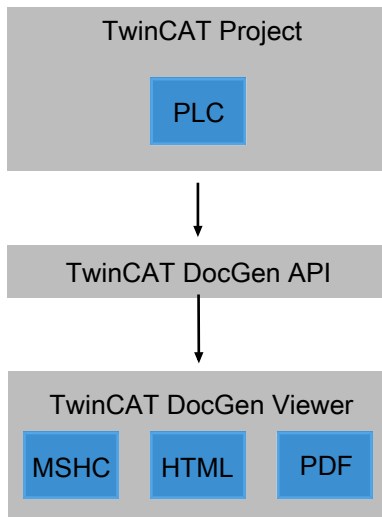
- Overview
  - Wird automatisch erzeugt und enthält auch das Inhaltsverzeichnis.
- Summary
- Description
- Declaration
- Example

### Ausgabeformate

TwinCAT 3 Documentation Generation gibt Ihnen die Möglichkeit Ihre Dokumentation in drei verschiedenen Ausgabeformaten zu erstellen:

- HTML
- PDF
- MSHC

Die Integration in TwinCAT ist entsprechend wie folgt aufgebaut:

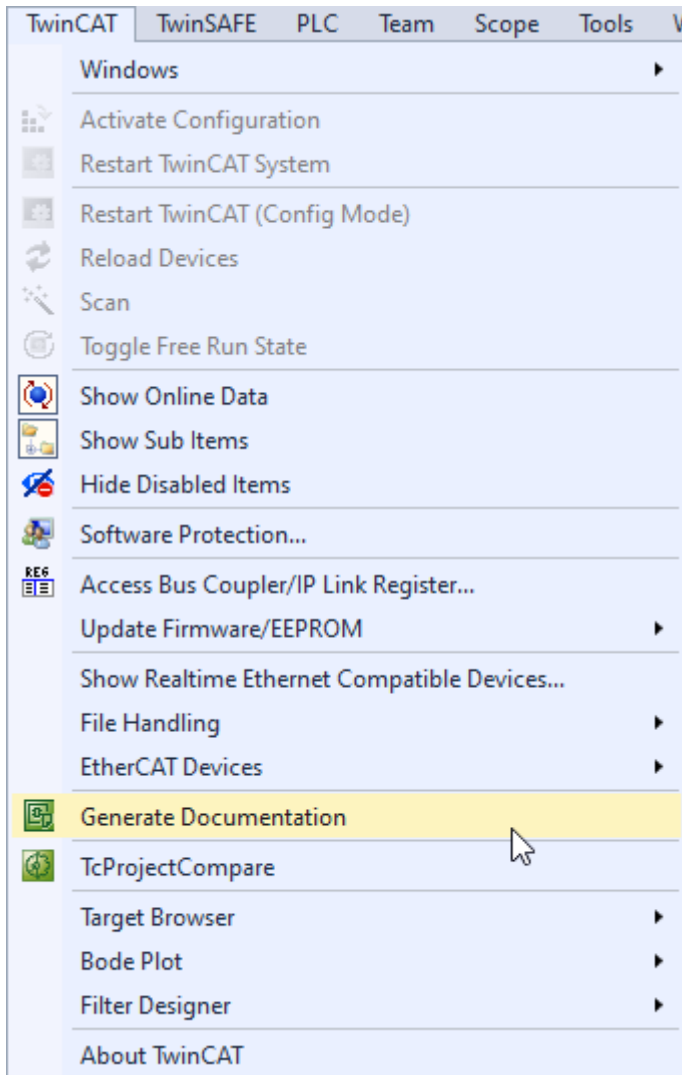


## 5 Konfiguration

Im Folgenden wird erklärt, wie Sie TwinCAT Documentation Generation öffnen und welche Einstellungen Sie beim Erstellen und Speichern einer Dokumentation vornehmen können.

### TwinCAT Documentation Generation öffnen

1. Klicken Sie auf die Registerkarte **TwinCAT**.
2. Klicken Sie auf **Generate Documentation**.

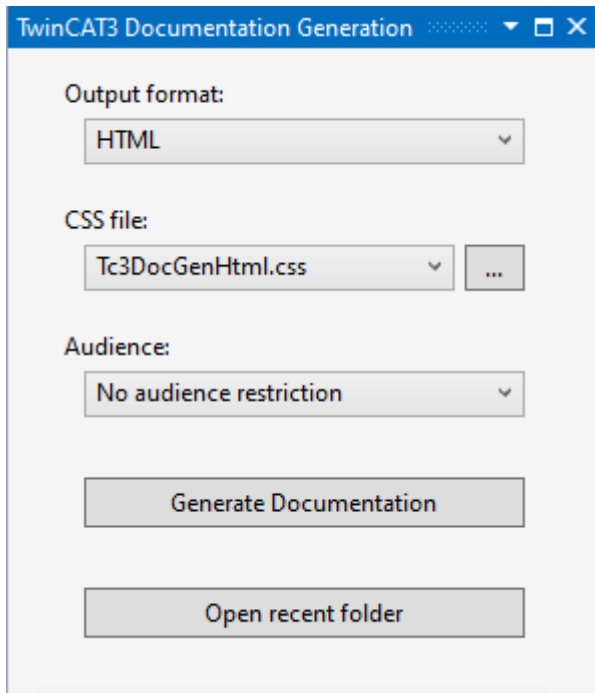


⇒ Das Fenster **TE1030-Tc3DocGen** öffnet sich.

### Dokumentation erzeugen und speichern

- ✓ Das Fenster **TE1030-Tc3DocGen** ist geöffnet.
1. Wählen Sie zuerst das Ausgabeformat aus.
  2. Wählen Sie im Drop-down-Menü **CSS file** ein Layout aus.
  3. Wählen Sie bei Bedarf die gewünschte Zielgruppe [► 23] aus.

4. Klicken Sie auf **Generate Documentation**.



⇒ Das Fenster **Ordner suchen** öffnet sich.

5. Wählen Sie einen Speicherort aus.

⇒ Die Dokumentation wird im ausgewählten Ordner abgelegt.

6. Sie können den Ordner, in dem das Dokument gespeichert wurde, mit der Schaltfläche **Open recent folder** direkt öffnen.

---

### ● Auswahl der Sprache bewirkt keine Übersetzung der Dokumentation

**i** Die ausgewählte Sprache setzt ein Sprachtag für die Information, welche Sprache die Dokumentation hat.

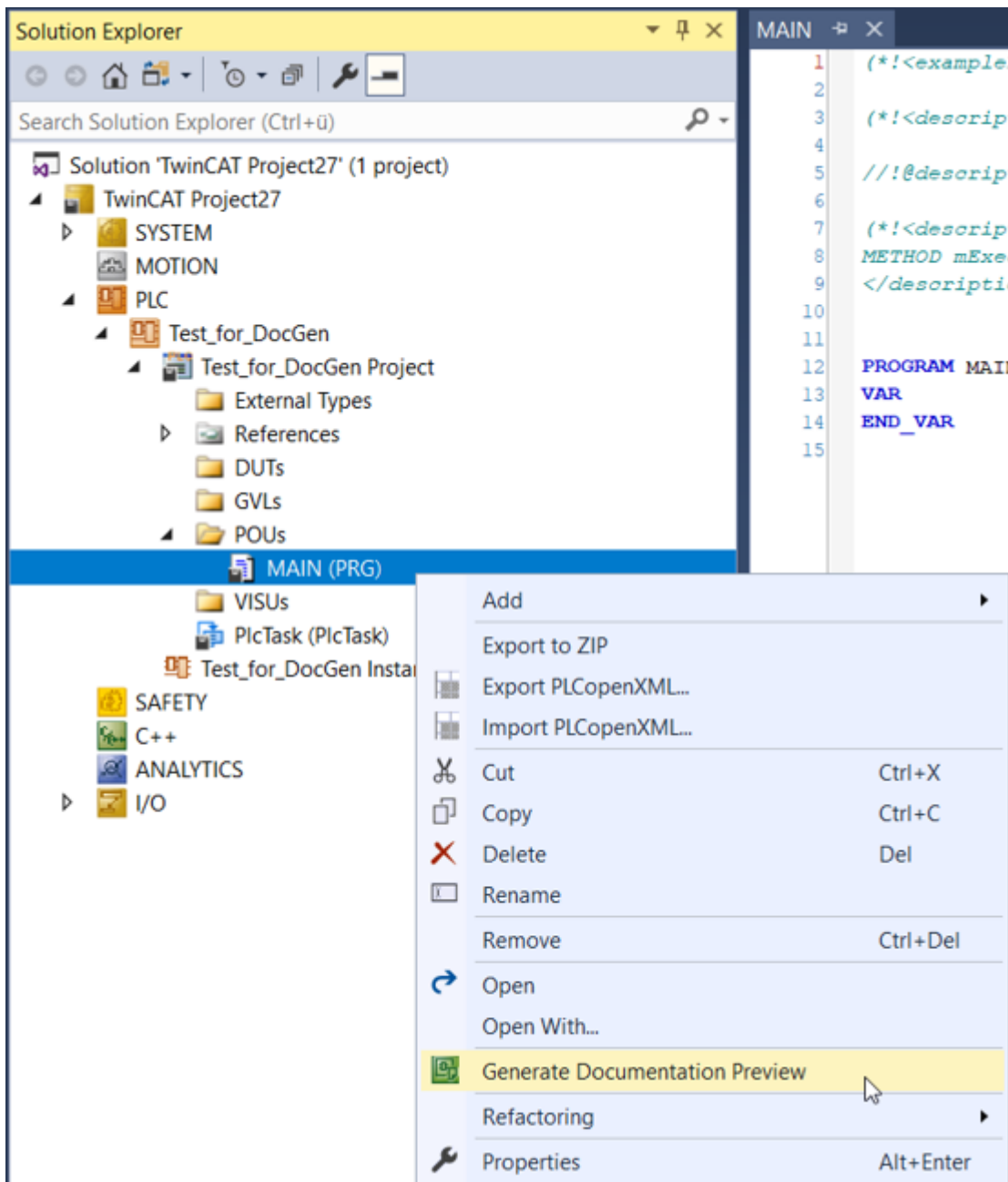
---

## 6 Vorschau der Dokumentation

Für Elemente, die in der SPS hängen, können Sie sich eine Vorschau dokumentieren lassen.

Gehen Sie zum Erstellen der Vorschau wie folgt vor:

1. Klicken Sie mit der rechten Maustaste auf das gewünschte Element im **Projektmappen-Explorer**.



2. Wählen Sie den Punkt **Generate Documentation Preview**.

⇒ Die Vorschau des generierten Dokuments öffnet sich in einem eigenen Fenster in TwinCAT.

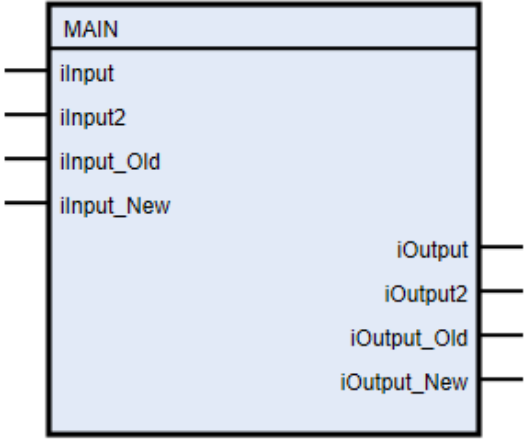
TE1030-Tc3DocGen Preview ▾ □ ×

Update preview

Beckhoff TcDocGen

**Program MAIN**

**Overview**



```

graph LR
    subgraph MAIN [MAIN]
        direction TB
        ilInput[ilInput]
        ilInput2[ilInput2]
        ilInput_Old[ilInput_Old]
        ilInput_New[ilInput_New]
        iOutput[iOutput]
        iOutput2[iOutput2]
        iOutput_Old[iOutput_Old]
        iOutput_New[iOutput_New]
    end
    ilInput --- MAIN
    ilInput2 --- MAIN
    ilInput_Old --- MAIN
    ilInput_New --- MAIN
    MAIN --- iOutput
    MAIN --- iOutput2
    MAIN --- iOutput_Old
    MAIN --- iOutput_New
        
```

Name	MAIN
Type	Program
Access Modifier	
Implements	
Extends	

**Summary**

This is a test summary

---

**Description**

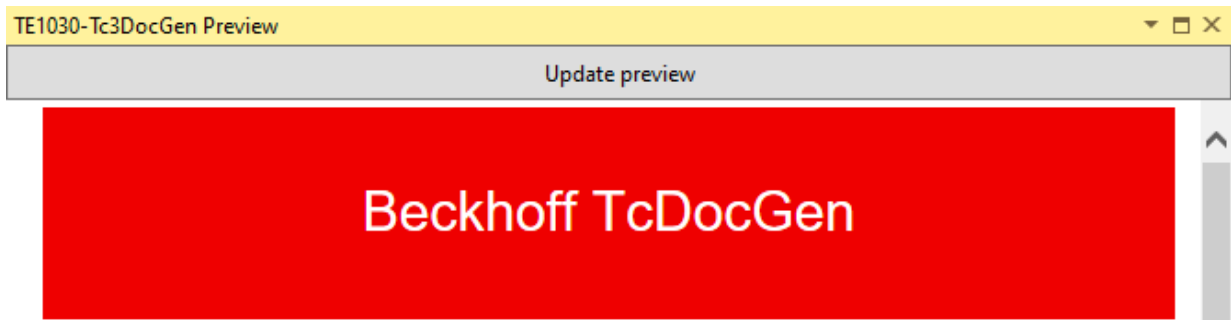
This is a description of the test summary as seen above.

---

**Vorschau aktualisieren**

Die Vorschau des generierten Dokuments kann im Vorschauenfenster aktualisiert werden.

1. Klicken Sie auf die Schaltfläche **Update preview**, um die Vorschau zu aktualisieren.



⇒ Die Vorschau wird sofort aktualisiert.



## 7 Markups

Im folgenden Kapitel geht es um die Markups, die dazu dienen, die Dokumentation zu erstellen. Die zur Dokumentationserstellung verwendeten Markups sind XML-basiert.

### Beispiel

```
//!<description> Dies ist eine Beschreibung. </description>
```

### Abgekürzte Formen der Tags

Wenn ein Dokumentationselement nur die Länge einer Zeile hat, kann die abgekürzte Form verwendet werden. In dieser Form werden die XML-Tags ohne eckige Klammern und schließende Elemente geschrieben. Anstelle der Klammern wird ein führendes @ zur Markierung der Tags verwendet.

### Beispiel

Normalform:

```
//!<summary> Zusammenfassung durch Verwendung der Normalform </summary>
```

Abgekürzte Form:

```
//!@summary Zusammenfassung durch Verwendung der abgekürzten Form
```

## 7.1 Übersicht

Im Folgenden finden Sie eine Übersicht über die Markups, die für die Dokumentationserstellung gebraucht werden.

Normalform	Abgekürzte Form	Beschreibung
<description> </description>	@description	Textabschnitt als Beschreibung gekennzeichnet.
<summary> </summary>	@summary	Textabschnitt als Zusammenfassung gekennzeichnet.
<example> </example>	-	Textabschnitt als Beispiel formatiert.
<h1> </h1> - <h6> </h6>	-	Überschriften bis Ebene 6
<code> </code>	@code	Abschnitt mit Quellcode
<literal> </literal>	@literal	Klartext, der nicht interpretiert werden soll.
-	\$crossreference	Querverweis
<see uri=""> </see>	@see	Link innerhalb eines Textabschnitts
<seealso uri=""> </seealso>	@seealso	Link außerhalb eines beliebigen Textabschnitts
<param name=""> </param>	@param	Parameterwert
<returns> </returns>	@returns	Rückgabewert
<ul> </ul>	@ul	Definiert eine ungeordnete Liste.
<ol> </ol>	@ol	Definiert eine nummerierte Liste.
<li> </li>	-	Element einer Liste
<table> </table>	-	Definiert eine Tabelle ähnlich wie bei HTML.
<audience> </audience>	-	Definiert eine Benutzergruppe für den markierten Textabschnitt.
<preliminary></preliminary>	@preliminary	Markiert einen Abschnitt als änderungsbedürftig.
<note type=""> </note>	-	Definiert Sicherheitshinweise (Gefahr, Warnung, Hinweis).
<image uri=""></image>	-	Bindet ein Bild ein.

## 7.2 Beschreibungen der Markups

Im Folgenden werden die einzelnen Markups genauer beschrieben und es gibt jeweils Beispiele der ausgeschriebenen Normalform und der abgekürzten Form.

### 7.2.1 Delimiters

Um die Dokumentation zu erzeugen, müssen in den Kommentaren Begrenzungszeichen eingefügt werden. Diese Begrenzungszeichen funktionieren in Kombination mit den Markups. Beispiele für den Einsatz der Begrenzungszeichen finden Sie in den Beispielen zu den einzelnen Markups.

Folgende Begrenzungszeichen sind erlaubt:

//!	Einzeiliger Dokumentationskommentar
(*! ...*)	Mehrzeiliger Dokumentationskommentar

### 7.2.2 Description

Dieses Markup-Element dient der Kennzeichnung eines Textblocks als Beschreibung. Beschreibungsblöcke können im Code an verschiedenen Stellen erstellt werden. In der generierten Dokumentation werden die einzelnen Beschreibungsblöcke dann zu einem Textblock zusammengefügt. Die Dokumentation kann also an den Stellen des SPS-Codes geschrieben werden, an denen es passt und muss nicht am Anfang oder Ende stehen.

Das Markup Description dient zusätzlich als Mantelelement und wird entsprechend benötigt, um verschiedene andere Elemente für die Dokumentation nutzen zu können. Auch in der abgekürzten Form des Markups Description können weitere Markups verwendet werden.

### Beispiel

Normalform:

Einzeiliger Kommentar:

```
///<description> Textblock, der als Beschreibung gekennzeichnet wird. </description>
```

Mehrzeiliger Kommentar:

```
(*!<description>Dieses ist eine Beschreibung.
```

```
<returns> Hier wird ein Rückgabewert genannt. </returns>
```

```
</description>*)
```

Abgekürzte Form:

```
///@description Dies könnte eine Beschreibung eines Code-Elements sein.
```

## 7.2.3 Summary

Dieses Markup-Element kann verwendet werden, um einen Text als Zusammenfassung eines Code-Elements zu markieren. Innerhalb der Summary dürfen Sie weitere Markups verwenden. Im Gegensatz zu der Description gibt es die Summary aber nur einmal pro Programmorganisationseinheit. Alle weiteren Stellen, an denen Sie das Markup verwenden, werden nicht in der Dokumentation ausgegeben. Verfassen Sie die Summary daher als zusammenhängenden Textblock.

### Beispiel

Normalform:

```
///<summary> Dies ist die Zusammenfassung der Beschreibung in der Normalform. </summary>
```

Abgekürzte Form:

```
///@summary Dies ist die Zusammenfassung der Beschreibung in der abgekürzten Form.
```

## 7.2.4 Example

Das Markup Example kann verwendet werden, um einen Textblock als Beispiel zu markieren. Beispiele können im Code an verschiedenen Stellen geschrieben werden. In der generierten Dokumentation werden die Beispiele dann unter der Überschrift Example untereinander dargestellt. Die Beispiele können also an den Stellen des Codes geschrieben werden, an denen sie Sinn ergeben und müssen nicht zusammenhängend am Anfang oder Ende stehen.

Innerhalb von Examples dürfen weitere Markups verwendet werden. Es gibt keine abgekürzte Form für dieses Markup.

### Beispiel

Normalform:

```
(*!<example>Dies ist die Beschreibung des folgenden Beispiels:
```

```
<code>iCount:=iCount+1;</code>
```

```
In diesem Beispiel wird die Erhöhung der Variablen $iCount um 1 angezeigt.</example>*)
```

## 7.2.5 Header

Dieses Markup-Element kann verwendet werden, um eine Überschrift zu formatieren. Sie können Ihre Inhalte mit den Überschriften bis zur sechsten Ebene gliedern. Dafür zählen Sie die Gliederungsnummer des Markups entsprechend bis zur „6“ hoch.

Das Markup Überschrift kann einzeln verwendet werden und muss in kein anderes Markup eingebunden werden. Außerdem können Sie es an jeder Stelle ihrer Dokumentation verwenden.

### Beispiel

```
<h1>Das ist eine Überschrift der ersten Ebene</h1>
```

```
<h2>Das ist eine Überschrift der zweiten Ebene</h2>
```

## 7.2.6 List

Diese Markierungen können zur Definition einer Liste verwendet werden. Es stehen folgende Listentypen zur Verfügung ungeordnete Liste `<ul>` oder geordnete Liste `<ol>`. Ähnlich wie bei HTML werden für die Elemente beider Listen die Markup-Listenelemente `<li>` verwendet. Ungeordnete und geordnete Listen müssen in eine Description, Summary oder ein Example eingebettet werden.

### Beispiel

Normalform:

```
(*!<description> Es folgt eine Liste
```

```
<ul>
```

```
<li> Inhalt von Element 1 </li>
```

```
<li> Listenpunkt </li>
```

```
<li> Listenpunkt </li>
```

```
</ul>
```

```
</description>*)
```

Abgekürzte Form:

```
!!@description @ul <li> Listenpunkt </li> <li> Listenpunkt </li> <li> Listenpunkt </li>
```

## 7.2.7 Table

Dieses Markup definiert eine Tabelle. Für eine Tabelle ist keine abgekürzte Form verfügbar. Sie müssen Tabellen in die Markups Description, Summary oder Example einbetten.

Die Tags definieren das Aussehen der Tabelle. Der Tag `<tr>` erzeugt eine Tabellenreihe. Der Tag `<td>` erzeugt eine Tabellenspalte.

### Beispiel

Normalform:

```
(*!<description> <table>
```

```
<tr>
```

```
<td> Inhalt von Zelle 1 </td>
```

```
<td> Inhalt von Zelle 2 </td>
```

```
</tr>
```

```
<tr>
<td> Inhalt von Zelle 3 </td>
<td> Inhalt von Zelle 4 </td>
</tr>
</table> </description>*)
```

### Tabelle mit Kopfzeile erzeugen

Sie haben außerdem die Möglichkeit die Tabelle mit Kopfzeile zu erzeugen. Dafür benutzen Sie statt `<td>` das Markup `<th>`.

### Beispiel

```
(*!<description> <table>
<tr>
<th> Kopfzelle 1 </th>
<th> Kopfzelle 2 </th>
</tr>
><tr>
<td> Inhalt von Zelle 1 </td>
<td> Inhalt von Zelle 2 </td>
</tr>
</table> </description>*)
```

## 7.2.8 Cross references

Um einen Verweis auf ein anderes Element innerhalb der Dokumentation zu erstellen, verwenden Sie das Zeichen "\$". Wenn sich das Verweisziel nicht innerhalb desselben Codeelements befindet, muss ein qualifizierter Name, also ein vollständiger Domainname, verwendet werden. TwinCAT Documentation Generation sucht automatisch nach diesem Element und erstellt einen Hyperlink in der Dokumentation. Den Querverweis verwenden Sie innerhalb der Markups Description, Summary und Example.

### Beispiel

Normalform:

```
//!<description> Ein Beispiel finden Sie im $Codeblock </description>
```

Abgekürzte Form:

```
//!@description Ein Beispiel finden Sie im $Codeblock
```

## 7.2.9 Links

Innerhalb Ihrer Dokumentation können Sie auf externe Adressen verlinken. Das Ziel des Links wird durch die Verwendung einer URI (Uniform Resource Identifier) angegeben. Die URI muss in normale Anführungszeichen gesetzt werden, damit die XML-Bibliothek die Zeichenkette korrekt verarbeiten kann. Links müssen Sie immer innerhalb einer Description, Summary oder eines Examples einsetzen. Sie können Links entweder in der gleichen Textzeile einfügen oder vor den Link einen Zeilenumbruch setzen. Dafür nutzen Sie die Markups „see“ und „seealso“.

## See

Verwenden Sie das Markup `<see>`, um einen Link innerhalb einer Textzeile zu erstellen.

### Beispiel

Normalform:

```
(*!<description> Weitere Informationen finden Sie auf unserer Homepage  
<see uri="http://www.beckhoff.de/TE1030"> Beckhoff Homepage </see>  
</description>*)
```

Abgekürzte Form:

```
!!@description Weitere Informationen finden Sie auf unserer Homepage @see uri=http://www.beckhoff.de/  
TE1030 Beckhoff Homepage
```

In der Dokumentation sieht das so aus:

### Description

You can find more information on our homepage [Beckhoff Homepage](http://www.beckhoff.de/TE1030)

### Seealso

Verwenden Sie das Markup `<seealso>`, um vor einem Link einen Zeilenumbruch zu erzeugen und so den Link in einen neuen Textabschnitt zu setzen.

### Beispiel

Normalform:

```
!!<description> Weitere Informationen finden Sie auf unserer Homepage <seealso uri="http://  
www.beckhoff.de/TE1030"> Beckhoff Homepage </seealso> </description>
```

Abgekürzte Form:

```
!! @description Weitere Informationen finden Sie auf unserer Homepage @seealso http://www.beckhoff.de/  
TE1030
```

In der Dokumentation sieht das so aus:

### Description

You can find more information on our homepage [Beckhoff Homepage](http://www.beckhoff.de/TE1030)

## 7.2.10 Code

Das Markup `<code>` kann verwendet werden, um einen Textblock als Quellcode zu kennzeichnen. Dieses Markup muss immer in eine Description, Summary oder ein Example eingebettet werden.

### Beispiel

Normalform:

```
(*!<description>Jetzt folgt ein Codeblock.  
<code> iCount:=iCount+1 </code>  
</description>*)
```

Abgekürzte Form:

```
(*!<description>Jetzt folgt ein Codeblock.
```

```
@code iCount:=iCount+1;  
</description>*)
```

### 7.2.11 Literal

Das Markup `<literal>` kann verwendet werden, um Text als reinen Text zu kennzeichnen, der nicht interpretiert werden soll. Diese Funktion kann hilfreich sein, um innerhalb des Quellcodes der Dokumentation Anmerkungen zu machen.

Literal verwenden Sie innerhalb der Markups Description, Summary und Example.

#### Beispiel

Normalform:

```
/*!<description> Diese Beschreibung enthält Text, der nicht interpretiert werden soll. Dieser folgt nun.  
<literal> Die Auszeichnung <b> fett </b> wird hier nicht interpretiert, sondern die Tags werden in die  
Dokumentation geschrieben. </literal> Hier folgt wieder Text, in dem die Auszeichnung <b> fett </b>  
interpretiert wird. </description>
```

Abgekürzte Form:

```
/*!@description Diese Beschreibung enthält Text, der nicht interpretiert werden soll. Dieser folgt nun. @literal  
Die Auszeichnung <b> fett </b> wird hier nicht interpretiert, sondern die Tags werden in die Dokumentation  
geschrieben.
```

### 7.2.12 Preliminary

Ermöglicht die Angabe, ob ein Abschnitt oder Element vorläufig ist und Änderungen unterliegt. Dieses Markup-Element ist nur eine Kennzeichnung und kann keine anderen internen Informationen enthalten. Preliminary müssen Sie in eine Description, Summary oder ein Example einbinden.

#### Beispiel

Normalform:

```
(*!<description>Jetzt folgt ein vorläufiger Textabschnitt.  
<preliminary> Vorläufig </preliminary>  
</description>*)
```

Abgekürzte Form:

```
/*!@description @preliminary Vorläufig
```

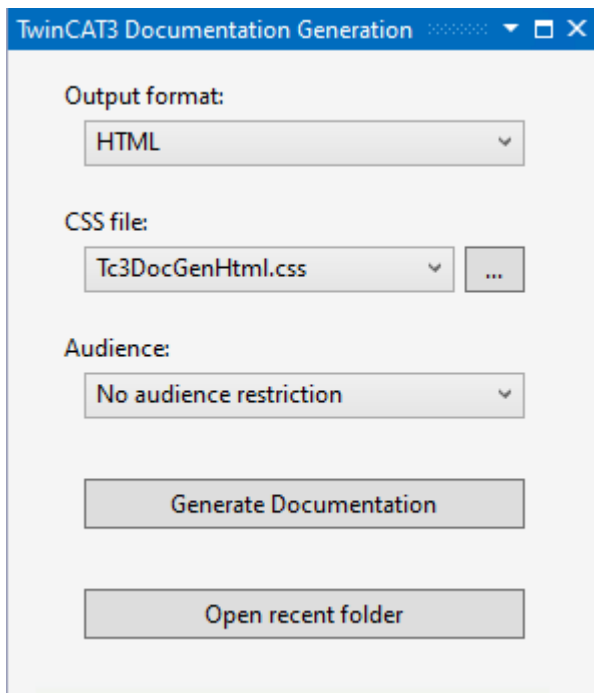
### 7.2.13 Audience

Mit diesem Markup können Sie definieren, dass Teile der Dokumentation nur für eine definierte Benutzergruppe generiert werden. Das Markup muss in eine Description, Summary oder ein Example eingebunden werden.

Die gewünschte Zielgruppe muss an zwei Stellen definiert werden.

Zum einen muss sie bei der Konfiguration der Dokumentation eingestellt werden.

1. Wählen Sie dafür im Drop-down-Menü **Audience** entweder „general“ oder „internal“ aus.



Zum anderen muss die Zielgruppe innerhalb des Markups als „audience type“ angegeben werden.

### Beispiel

Normalform:

```
(*!<description>Dieses ist ein normaler Teil der Dokumentation.
<audience type="internal">Dieser Teil wird nur für den internen Gebrauch angezeigt.</audience></
description >*)
```

Wenn die Zielgruppe an beiden Stellen eingestellt wurde, wird die Dokumentation entsprechend ausgegeben. Derzeit sind die beiden Zielgruppen „general“ und „internal“ verfügbar. Wenn Sie die Zielgruppe „internal“ einstellen, wird der Text innerhalb des Markups nicht in die Dokumentation ausgegeben.

## 7.2.14 Param

Das Markup Param erlaubt es, einen Parameterwert von z.B. Variablendeklarationen zu beschreiben. Es kann optional verwendet werden, wenn der Platz hinter der Parameterdefinition im Deklarationseditor nicht ausreicht, oder um die komplette Dokumentation an der gleichen Stelle zu haben.

Der Param steht für sich alleine und muss nicht in eine Description, eine Summary oder ein Example eingebunden werden. Durch das Markup Param erzeugen Sie eine Tabelle, in der die Variablen und ihre Werte aufgeführt werden.

### Beispiel

Normalform:

```
//!<param name="fFloatVar">Some float var</param>
```

```
//!<param name="iCount2">Some count var</param>
```

```
PROGRAM MAIN
```

```
VAR_OUTPUT
```

```
  fFloatVar : REAL;
```

```
  iCount2 :INT;
```

```
END_VAR
```



Abgekürzte Form:

```
///@param fFloatVar Some float var
```

```
///@param iCount2 Some count var
```

In der Dokumentation sieht das so aus:

### Declaration

```
PROGRAM MAIN
VAR_OUTPUT
fFloatVar : REAL;
iCount2 :INT;
END_VAR
```

### Output Variables

<b>fFloatVar</b>	REAL	Some float var
<b>iCount2</b>	INT	Some count var

## 7.2.15 Returns

Dieses Markup erlaubt es, den Rückgabewert einer Methode zu beschreiben.

### Beispiel:

Normalform:

```
///<description> <returns> Der Rückgabewert, wenn diese Methode größer als 0 sein muss.</returns> </description>
```

Abgekürzte Form:

```
///@description @returns Der Rückgabewert, wenn diese Methode größer als 0 sein muss.
```

## 7.2.16 Note

Das Markup Note verwenden Sie, um eine Sicherheitswarnung zu definieren. Das gesamte Markup müssen Sie entweder in eine Description, Summary oder ein Example einbetten. Für das Markup Note gibt es keine abgekürzte Form.

Eine Sicherheitswarnung kann vom Typ Information, Warnung oder Gefahr sein. Den Typ der Sicherheitswarnung definieren Sie innerhalb des Markups als „Note Type“ mit einem Signalwort. Das Signalwort wird dann entsprechend der Gefahrenstufe farbig hinterlegt.

Mögliche Typen, die Sie definieren können:

`<note type="hazard">` - Gefahr

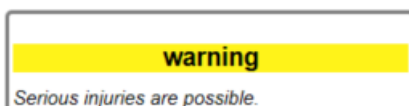
`<note type="warning">` - Warnung

`<note type="information">` - Hinweis

### Beispiel

Normalform:

```
///<description> <note type="warning">Schwere Verletzungen sind möglich.</note> </description>
```



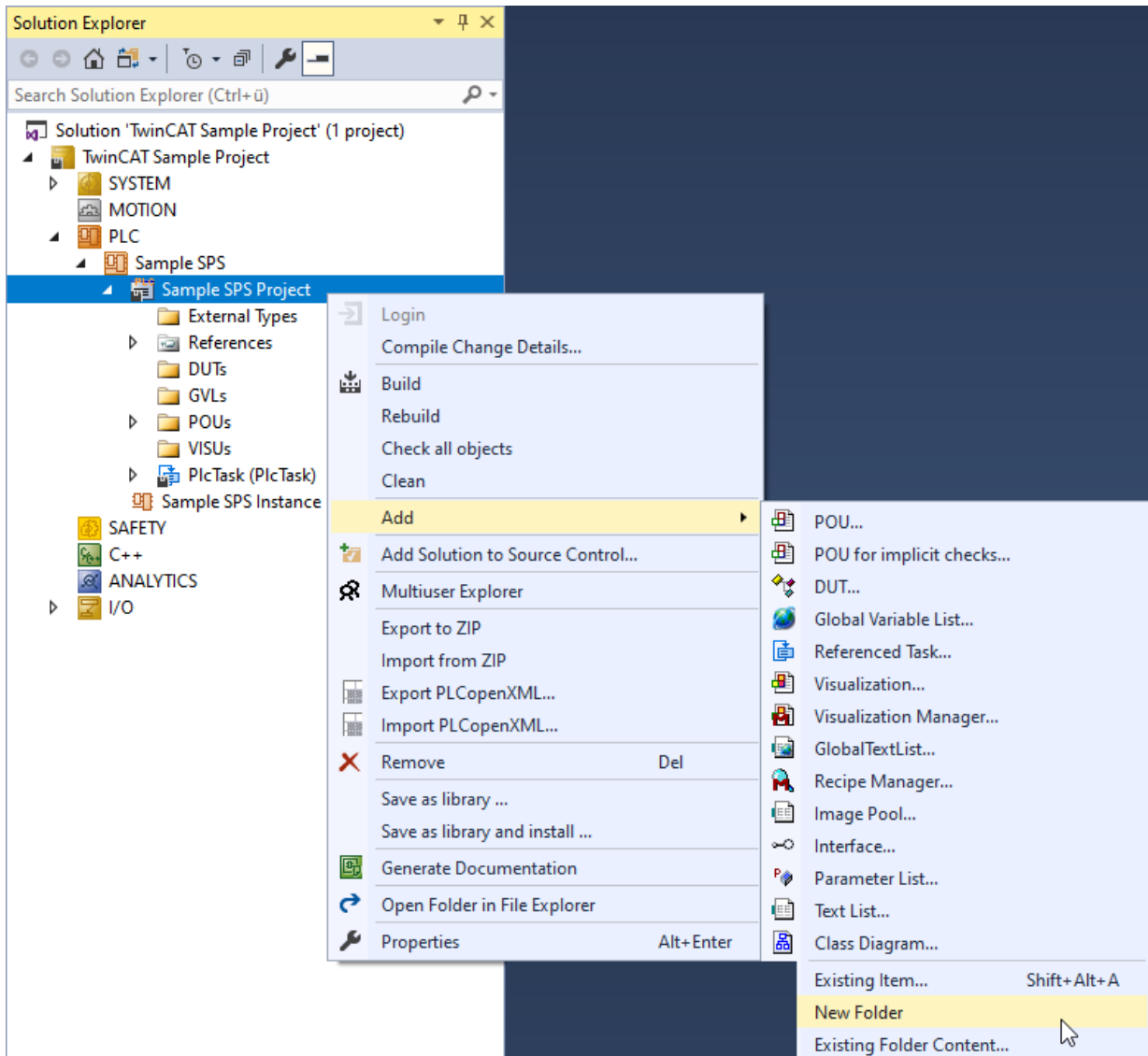
## 7.2.17 Image

Mit Hilfe des Markups Image, das in eine Description, Summary oder ein Example eingefügt werden muss, können Sie Bilder in Ihrer Dokumentation einbinden.

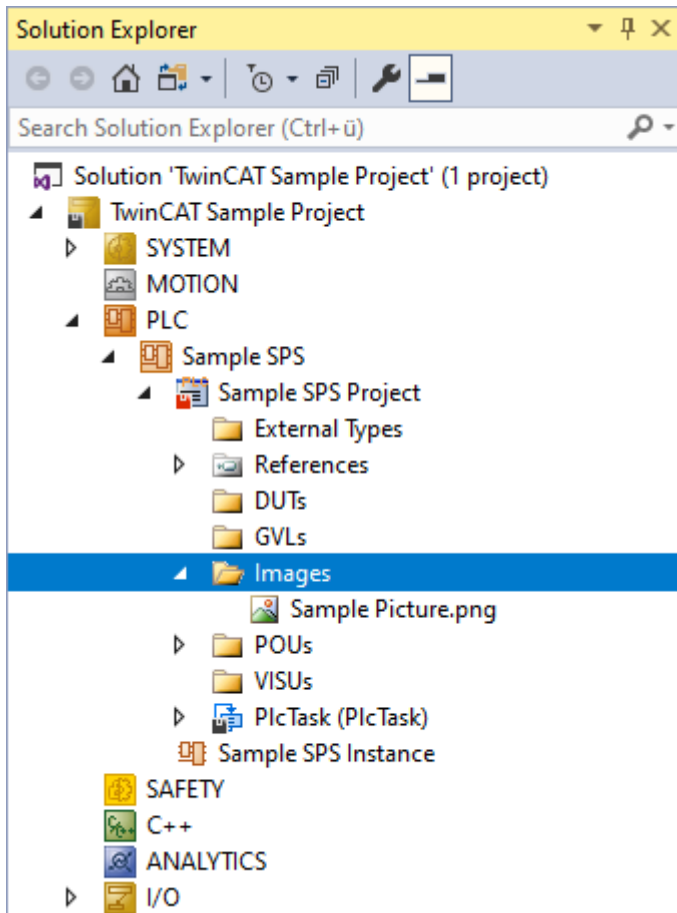
Um Bilder einzubinden, legen Sie einen Ordner namens „Images“ in dem Ordner des SPS-Projekts an. Der Ordner „Images“ darf Unterordner haben, um die Bilder zu strukturieren. Alle Verweise auf die Bilder erfolgen dann innerhalb dieses Ordners, also "Unterordner/Bildname".

Die Bilder wie folgt im SPS-Projektordner ablegen:

1. Legen Sie einen neuen Ordner unterhalb Ihres SPS-Projekts an.



2. Nennen Sie den Ordner „Images“.



3. Ziehen Sie alle Bilder, die Sie verwenden möchten, per Drag-and-Drop in den Ordner.

⇒ Sie können die abgelegten Bilder jetzt in Ihrer Dokumentation verwenden.

Die Bilder wie folgt in die Dokumentation einbinden:

1. Binden Sie das Markup Image entweder in das Markup Description, Summary oder Example ein.

2. Geben Sie innerhalb des Markups den Namen des Bildes an.

⇒ Das Bild wird in die Dokumentation ausgegeben.



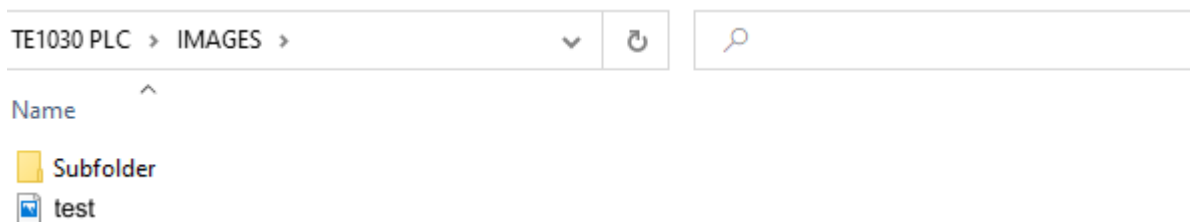
Damit das Bild gefunden werden kann, muss der Bildname übereinstimmen.

- Achten Sie darauf, den Namen des Bildes im Markup korrekt zu schreiben.

**Beispiel**

```
(*!<description>Dieses ist eine Beschreibung mit Bild.
<image uri="Sample Picture.png">Ersatztext</image></description>*)
```

Alternativ können Sie die Bilder über den Absoluten Speicherpfad in die Dokumentation einbinden.



```
(*!<description> Dieses ist eine Beschreibung mit Bild.
<image uri="C:/TE1030 PLC/IMAGES/test.jpg">Ersatztext</image> </description>*)
```



Wenn das Bild nicht angezeigt werden kann, wird stattdessen ein Ersatztext, den Sie erstellen, in der Dokumentation generiert.

---

### Bildergröße skalieren

Die eingebundenen Bilder können mit Hilfe der Markups „width“ und „height“ skaliert werden.

Normalform:

(\*!<description>Dieses ist eine Beschreibung mit Bild.

<image uri="test.jpg">Bildtitel</image>

width="10"

height="20"

</description>\*)

## 7.2.18 Character formats

Sie können verschiedene Zeichenformate für Ihre Dokumentation verwenden. Diese orientieren sich bei TwinCAT Documentation Generation an der HTML-Konvention.

### Fett

Sie haben die Möglichkeit einzelne Ausdrücke in einem Text fett zu kennzeichnen. Das funktioniert, indem Sie das Markup <b> verwenden.

Beispiel:

(\*!<description> In der folgenden Beschreibung wird ein Wort <b>fett</b> dargestellt. </description>\*)

### Kursiv

Sie haben die Möglichkeit einzelne Ausdrücke in einem Text kursiv zu kennzeichnen. Das funktioniert, indem Sie das Markup <i> verwenden.

Beispiel:

(\*!<description> In der folgenden Beschreibung wird ein Wort <i>kursiv</i> dargestellt. </description>\*)

### Code

Sie haben die Möglichkeit einzelne Variablenausdrücke in einem Text als Code zu kennzeichnen. Das funktioniert, indem Sie das Markup <c> verwenden.

Beispiel:

(\*!<description> Die Variable <c>iCount</c> wird jetzt mit diesem Textblock zusammen formatiert.  
<code>iCount:=iCount+1;</code> </description>\*)

### Unterstrichen

Sie haben die Möglichkeit einzelne Ausdrücke in einem Text zu unterstreichen. Das funktioniert, indem Sie das Markup <u> verwenden.

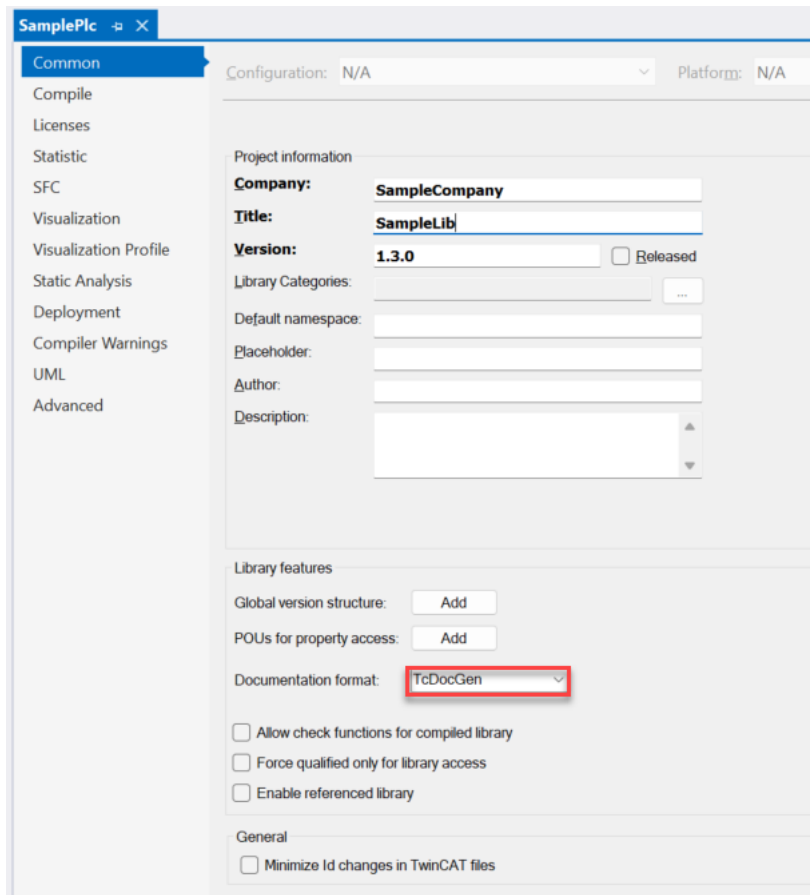
Beispiel:

(\*!<description> In der folgenden Beschreibung wird ein Wort <u>unterstrichen</u>. </description>\*)

## 8 SPS-Bibliotheken dokumentieren

Für SPS-Bibliotheken können Sie wie folgt eine Dokumentation erstellen:

- ✓ Dokumentieren Sie eine Bibliothek bereits, während Sie sie erstellen.
1. Öffnen Sie über das Kontextmenü des SPS-Projektknosens den Eigenschaftendialog des Projekts.
  2. Tragen Sie den Namen der Firma, den Titel der Bibliothek und die Versionsnummer ein.
  3. Wählen Sie als **Documentation format** „TcDocGen“ aus.



4. Erstellen und installieren Sie die Bibliothek.

⇒ Die Bibliothek kann nun wie gewohnt verwendet werden. Zudem kann sich der Nutzer der Bibliothek die Dokumentation im Bibliotheksmanager anzeigen lassen.

The screenshot displays the Beckhoff Library Manager interface. On the left, the Solution Explorer shows a project named 'SampleTcDocGen' with a sub-project 'SamplePlc' containing a 'SamplePlc Project' folder. The 'References' section lists 'SampleLib', 'Tc2\_Standard', 'Tc2\_System', and 'Tc3\_Module'. The 'Library Manager' window is open, showing a table of libraries used in the application:

Name	Namespace	Effective Version
SampleLib = SampleLib, * (SampleCompany)	SampleLib	1.3.0
Tc2_Standard = Tc2_Standard, * (Beckhoff Automation GmbH)	Tc2_Standard	3.4.5.0
Tc2_System = Tc2_System, * (Beckhoff Automation GmbH)	Tc2_System	3.6.4.0
Tc3_GlobalTypes = Tc3_GlobalTypes - SampleTcDocGen, 1.0.0.0 (Beckhoff Automation GmbH)	Tc3_GlobalTypes	1.0.0.0

The 'Contents of selected library' section shows a tree view of the 'SampleLib, 1.3.0 (SampleCompany)' library, with 'FB\_Cylinder' selected. The 'Details about selected library element' window shows the 'Documentation' tab for 'FB\_Cylinder'. The documentation features a red header with the text 'Beckhoff TcDocGen' and the title 'Function Block FB\_Cylinder'. Below the title, there is an 'Overview' section with a diagram showing the 'FB\_Cylinder' block and its connection to an 'ICylinder' block. The diagram includes a table with the following data:

FB_Cylinder
DTest
M_StateMachine - BOOL
P_TimerValue - LREAL

## 9 Beispiel

```
(*! @summary a program with a simple counter
<description>This is a sample program in which i will increase a counter variable. Of course the description
of this program could be longer than one line </description> *)
/*! @param iCount2 variable to increase the counter
PROGRAM MAIN
VAR
    btest AT%Q* :BOOL; /*! A sample variable not needed for this POU
    iCount2: INT; // This is my internal comment which will not be visible in the docu
    myPOU : POU    /*! This is a sample of a FB instanciated within this program
END_VAR

(*! <example> This sample shows how to increase the variable <c>iCount2</c>.
The following code can be used to increase $iCount:
<code>iCount2:=iCount2+1; </code></example>*)
/*! @summary sample function block for the PS meeting

FUNCTION_BLOCK FB_Cylinder IMPLEMENTS ICylinder
VAR_INPUT
    bTest :BOOL;
END_VAR
VAR

/*! @param tTimerValue internal TIME variable that is being accessed via property "P_Timervalue" (Get/Set)
tTimerValue    : TIME := T#500MS; //internal comment
iState         : INT;  /*! state of the state machine
bAtLeftPosFbk  : BOOL; //internal comment
bAtRightPosFbk: BOOL;  /*! software watchdog to reach the end position
TimerReach     : TON;  /*! software watchdog to reach the end position
TimerStart     : TON;  /*! Start timer
END_VAR

/*! <description> regular FB that additionally implements an interface "ICylinder" </description>
```

## 10 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](http://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](mailto: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](mailto: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](mailto:info@beckhoff.com)

Internet: [www.beckhoff.com](http://www.beckhoff.com)





Mehr Informationen:  
**[www.beckhoff.com/te1030](http://www.beckhoff.com/te1030)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

