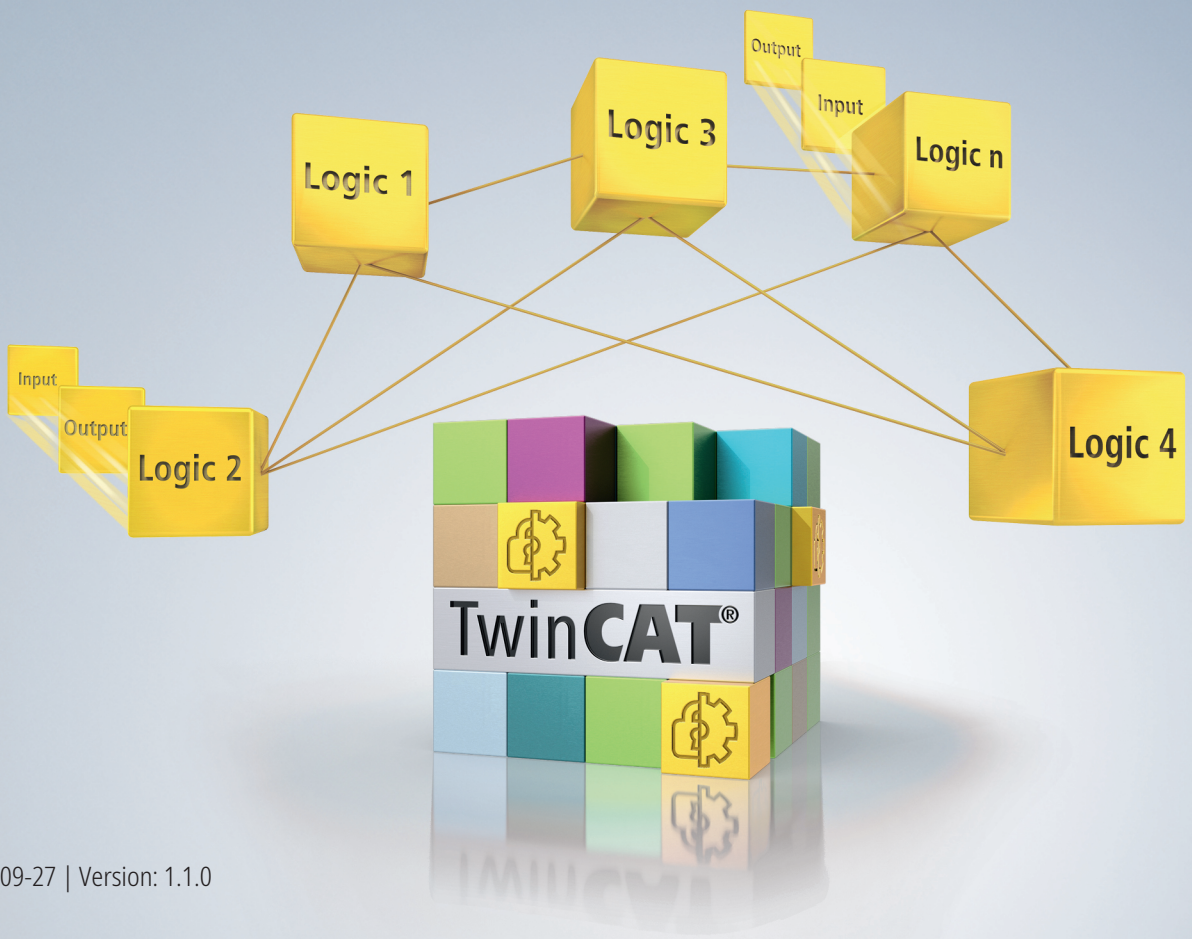
BECKHOFF New Automation Technology

Original documentation | EN

Import and export of XML files

TwinCAT 3 Safety Editor



2024-09-27 | Version: 1.1.0

Table of contents

1 Documentation notes

1.1 Disclaimer

Beckhoff products are subject to continuous further development. We reserve the right to revise the operating instructions at any time and without prior announcement. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in these operating instructions.

In these operating instructions we define all permissible use cases whose properties and operating conditions we can guarantee. The use cases we define are fully tested and certified. Use cases beyond this, which are not described in these operating instructions, require the approval of Beckhoff Automation GmbH & Co KG.

1.1.1 Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
The use of other brand names or designations by third parties may lead to an infringement of the rights of the owners of the corresponding designations.

1.1.2 Patents

The EtherCAT technology is protected by patent rights through the following registrations and patents with corresponding applications and registrations in various other countries:

- EP1590927
- EP1789857
- EP1456722
- EP2137893
- DE102015105702

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH.

Safety over EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH.

1.1.3 Limitation of liability

All components in this product as described in the operating instructions are delivered in a specific configuration of hardware and software, depending on the application regulations. Modifications and changes to the hardware and/or software configuration that go beyond the documented options are prohibited and nullify the liability of Beckhoff Automation GmbH & Co. KG.

The following is excluded from the liability:

- Failure to observe these operating instructions
- Improper use
- Use of untrained personnel
- Use of unauthorized spare parts

1.1.4 Copyright

1.2 Version numbers of the documentation

Issue	Comment
1.1.0	• Corrections • FBAdvPosMon added
1.0.0	• First released version
0.0.1	• Preliminary (internal only)

Currentness

Please check whether you are using the current and valid version of this document. The current version can be downloaded from the Beckhoff homepage at http://www.beckhoff.com/twinsafe. In case of doubt, please contact Technical Support (see Beckhoff Services).

Origin of the document

This original documentation is written in German. All other languages are derived from the German original.

Product features

The valid product properties are always those specified in the current manual. Further information given on the product pages of the Beckhoff homepage, in emails or in other publications is not authoritative.

1.3 References

No.	Issue	Title / description
[1]	1.4 or newer	**Manual Automation Interface TwinCAT 3** The document describes the basics, the procedure and the methods of the Automation Interface in TwinCAT 3.
[2]	1.0 or newer	**Beckhoff ELx9xx ESI file** The document contains an XML description of EtherCAT slaves of the ELx9xx group.
[3]	1.4.1 or newer	**Operating instructions for EL6910 TwinSAFE Logic module** The document contains a description of the Logic functions of the EL6910, and thus also of the AMI8911, and their programming
[4]	3.1.0 or newer	**Documentation for TwinSAFE Logic FB** The document describes the safety function blocks that are available in the EL6910, and thus also in the AMI8911 option card, and form the safety application.
[5]	1.8.0 or newer	**TwinSAFE Application Guide** The Application Guide provides the user with examples for the calculation of target failure measures for safety functions according to the standards DIN EN ISO 13849-1 and EN 62061 or EN 61508:2010, such as are typically used on machines.
[6]	2006/42/EC	**Directive 2006/42/EC of the European Parliament and of the Council of 17. May 2006 on machinery and amending Directive 95/16-7/EC (revised) of 29 June 2006** This directive, also known as the Machinery Directive, defines requirements for the placing on the market of machines and machine-like components, such as safety components.

1.4 Staff qualification

These operating instructions are intended exclusively for trained specialists in control technology and automation with the relevant knowledge.

The trained specialist personnel must ensure that the applications and use of the described product meet all safety requirements. This includes all applicable and valid laws, regulations, provisions and standards.

Trained specialists

Trained specialists have extensive technical knowledge from studies, apprenticeships or technical training. Understanding of control technology and automation is available. Trained specialists can:

- Independently identify, avoid and eliminate sources of hazard.
- Apply relevant standards and directives.
- Implement specifications from accident prevention regulations.
- Evaluate, prepare and set up the workplaces.
- Evaluate, optimize and execute work independently.

1.5 Safety and instruction

Read the contents that refer to the activities you have to perform with the product. Always read the chapter For your safety in the operating instructions.

Observe the warnings in the chapters so that you can handle and work with the product as intended and safely.

1.5.1 Explanation of symbols

Various symbols are used for a clear arrangement:

1. The numbering indicates an action that should be taken.
- The bullet point indicates an enumeration.

[…] The square brackets indicate cross-references to other text passages in the document.

[1] The number in square brackets indicates the numbering of a referenced document.

1.5.1.1 Signal words

The signal words used in the documentation are classified below.

Warning of personal injuries

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
Notes The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
Recommendations for action, assistance or further information on the product.

1.6 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

1.7 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 List of Abbreviations

Enclosed you will find a list of the abbreviations used and their meaning:

Abbreviation	Meaning
ESI	EtherCAT Slave Information
FB	Function block
FB group	Function block group
FSoE	FailSafe over EtherCAT
TwinSAFE SC	TwinSAFE Single Channel
User FB	User function block
User FB Libs	Libraries of the user function blocks
XSD	XML Schema Definition

3 For your safety

3.1 Duty of care

The operator must comply with all the requirements and notes specified in these operating instructions in order to fulfill his duty of care. This includes in particular that you

- read the entire documentation of the TwinSAFE component
- comply with the provisions defined in the chapter Limitation of liability [▶ 6].
- only operate the TwinSAFE component when it is in perfect working order.
- provide the operating instructions in a legible condition and complete at the place of use of the TwinSAFE component.

3.2 General safety instructions

Use in machines according to the Machinery Directive

Only use the TwinSAFE component in machines that comply with the Machinery Directive. This is how you ensure safe operation.

For more information, see document [6] at References [▶ 7].

Intended use

Any use of the TwinSAFE components that goes beyond the intended use described is not permitted.

Check safety functions

Perform a new acceptance of the safety functions each time you make a change to your safety project. This also includes reading out and checking the parameters and customizing settings currently active on the TwinSAFE safety controller.

Specify workflow

Specify a workflow for loading and customizing a safety project. This is how you ensure that the correct safety project for the application is activated on the TwinSAFE logic component.

4 Interface description

This document describes the data format of the import/export interface of the TwinCAT 3 Safety Editor.

If you use this description to create an import file, consider the requirements of the relevant norms and standards for the development of safety-relevant software or software tools. Files created externally (outside the TwinCAT 3 Safety Editor) must comply with the file schema. Check the file for correctness and freedom from errors.

Take the appropriate measures based on the IEC 61508 standard. According to this standard, this interface is a T3 tool.

In addition, essential protective measures of the TwinCAT 3 Safety Editor are overridden when importing an external generation of the XML application file. You should therefore adhere to the following criteria:

- Uniqueness of the FB and addresses
- Correctness of the connections, links and data types
- Correctness of the parameters and project size

This means in particular:

- Correct safe parameters of the connections, groups and FBs
- Correct, clear and intended processing sequences of connections, groups and FBs
- Size and distribution (position, length, data type) of the connection telegrams in individual signals. This applies to both connection directions.
- Unique names of groups, networks, FBs, variables, etc. in the respective context.
- Logic (type) features and limits used
- No multiple writing of inputs (FB or group inputs, connection or standard signal outputs)
- Correct user FBs
- Correct references to user FB libs
- Correct and unambiguous selected logic and selected logic type
- Correct assignment of backup/restore settings to connections
- Correct MultiSetting assignments to FBs, connections and System Manager nodes as well as correct parameter settings
- Correct specification of the process image
- Correct mapping of logic, connections and standard signals to System Manager nodes

Correctness cannot be checked by the TwinCAT 3 Safety Editor.

This interface will be expanded in the future as part of the adaptation to cybersecurity requirements. Due to the legal situation, it is not possible to maintain the interface in its current form.

Ask your Beckhoff contact person about the upcoming modifications and the roadmap.

5 System description

5.1 General

In TwinCAT 3 you have 2 options for exporting or importing a safety project in XML format.

- The TwinCAT 2 XML format is used to exchange safety projects between the TwinCAT 2 and TwinCAT 3 development environments.
- The TwinCAT 3 XML format allows you to make your safety project available outside TwinCAT 3 in XML format.
  This allows you to save the project in XML format for Source Control, for example. The XML schema of this XML file also provides the ability to create a project.

This documentation is limited to export and import in TwinCAT 3 XML format.

5.2 System requirements

The following versions are required for XML export or import:

TwinCAT	TE9000	XML	XSD
3.1 4020.0 - 3.1 4022.0	No separate setup, integrated in TwinCAT	0.24	Not available
3.1 4022.2 - 3.1 4024.7		0.27	
3.1 4024.10 or newer	1.0.0 or newer	0.30	
	1.4.6	0.31	0.31

All TwinSAFE components that can be used in TwinCAT 3 are suitable as target systems, with the exception of the TwinCAT Safety PLC.

The current TwinCAT 3 software can be found on the Beckhoff website at TwinCAT 3 download / eXtended Automation Engineering (XAE).

The current TwinCAT 3 Safety Editor can be found on the Beckhoff website at TwinCAT 3 Safety Editor.

5.3 System limits

XML export

You need a valid safety project for the XML export. You cannot export non-valid projects.

XML import

Not all functions may be available in older TwinCAT 3 versions.

If you import an XML file that was created with a newer version than the TwinCAT version into which you want to import, it is possible that some data or even the entire project will not be imported.

To ensure that all data is retained during import, avoid importing into an older TwinCAT version.

5.4 Intended use

Operate the TwinSAFE component exclusively for the intended activities defined in this documentation, taking into account the prescribed values.

The functions allow the use of Beckhoff TwinSAFE components in the area of machine safety. The TwinSAFE components are designed for machine safety functions and directly associated industrial automation tasks.

TwinSAFE components are therefore only approved for applications with a defined "fail-safe state". This safe state is the wattless state. Fail-safety according to the relevant standards is required.

⚠ WARNING

Improper use

Any use which exceeds the permissible written values or which does not observe other specifications from these operating instructions or other documents of the overall documentation is considered to be not in accordance with the intended use and is therefore prohibited.

This applies in particular to the use cases defined by Beckhoff Automation, which have been fully tested and certified and whose properties and operating conditions can be guaranteed. Use cases beyond this are regarded as inappropriate and require the approval of Beckhoff Automation.

Improper use will result in loss of safety and invalidation of certifications and approval.

6 Functioning

There are 2 different ways to export or import your safety project, as described in the following chapters.

6.1 Manual

This chapter contains a description of the manual procedure, which is carried out in the TwinCAT 3 development environment via the context menu of the safety project.

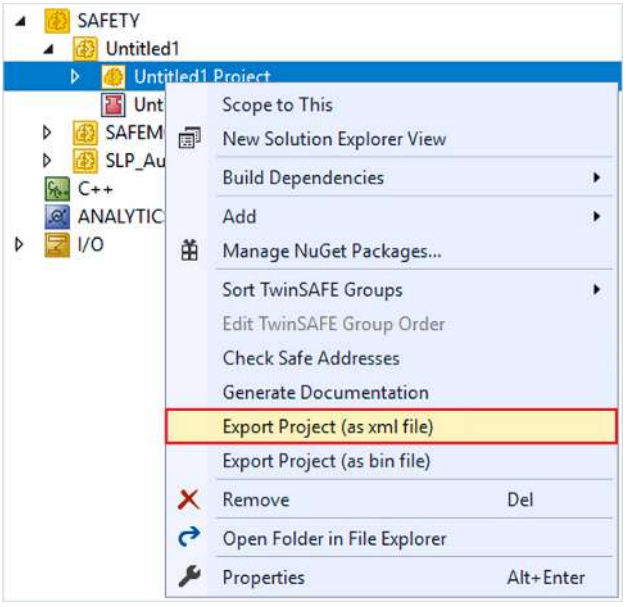
6.1.1 Export

Meet export requirements

- valid safety project

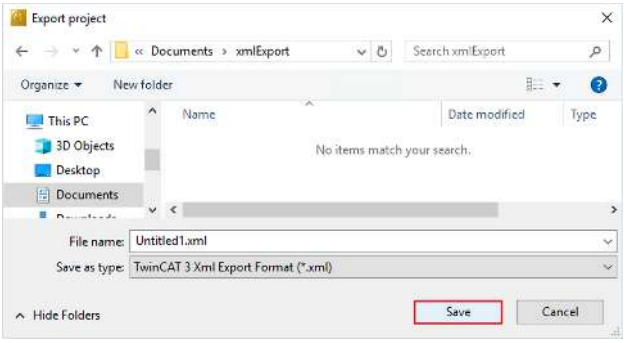
  ⇨ You can check the validity of your safety project using the "Verify Safety project" or "Verify Safety project / I/O configuration" function buttons in the menu bar.

Proceed as follows to export a safety project:



1. Right-click on Safety project
2. Click on "Export project as XML file"

3. Select storage location
4. Select the format "TwinCAT 3 Xml Export Format (*xml)" in the "File type" line

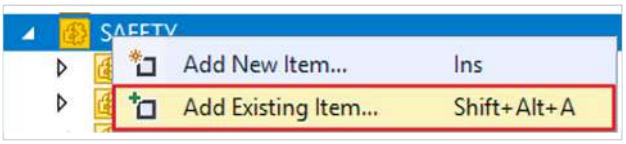


5. Confirm storage location with "Save"

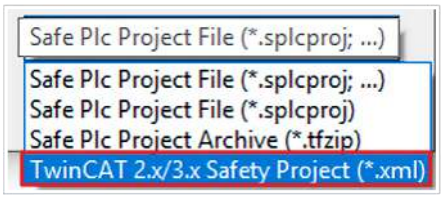
6.1.2 Import

Fulfill import requirements

- Valid data in your XML file
- XML file valid for XSD file

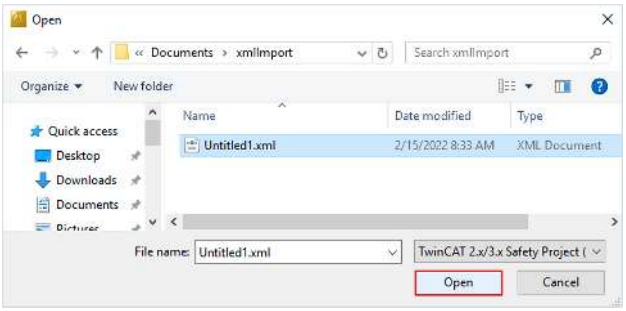


1. Right-click on Safety application
2. Click on "Add existing element..."



Your Explorer opens and you can select a file.

3. Click on "TwinCAT 2.x/3.x Safety Project (*xml)" in the search bar on the right to search for XML files



4. Select storage location
5. Select file
6. Confirm with “Open”

6.2 Automation Interface

The TwinCAT Automation Interface enables you to automatically generate and edit TwinCAT XAE configurations using programming codes or script codes.

The Automation Interface supports XML export and XML import from TwinCAT 3 version 3.1.4020.0 or newer.

For more information on the Automation Interface, see the document [1] at References [▶ 7].

Manual testing of the Automation Interface can be found in chapter Test Automation Interface [▶ 101].

6.2.1 Export

This chapter shows a sample of what a programmatic application can look like in C# code.

Manual testing of the Automation Interface can be found in chapter Manual export [▶ 101].

Methods of the Automation Interface

When exporting with the Automation Interface, use the following methods:

- LookupTreeItem
- ProduceXml
- ConsumeXml

⇨ For more information on these methods, see the Beckhoff Information System.

Usings

You need the following usings for this sample:

- System.Collections
- System.Xml
- System.Xml.XPath
- TCatSysManagerLib

Proceed as follows to export a Safety project with the Automation Interface:

```
/// <summary>
/// Export all Safety Projects in TC3 XML format.
/// </summary>
/// <param name="systemManager">System Manager</param>
private void Tc3SafetyProjectXmlExport(ITcSysManager systemManager)
{
```

1. Look up the main item of the safety tree in the TwinCAT 3 project

```
  ITcSmTreeItem treeItem = systemManager.LookupTreeItem("TISC");
```

2. Get the enumerator of the main item of the safety tree to run through the safety projects

```
  IEnumerator enumerator = treeItem.GetEnumerator();
  while (enumerator.MoveNext())
  {
```

3. Get safety project

```
    ITcSmTreeItem safetyProjectParentNode = enumerator.Current as ITcSmTreeItem;
    ITcSmTreeItem safetyProjectNode = safetyProjectParentNode.LookupChild(safetyProjectParent
    Node.Name + " Project");
    if (safetyProjectNode != null)
    {
```

4. Export XML on the safety project tree item

```
      string producedXml = safetyProjectNode.ProduceXml();
      XmlDocument xmlDoc = new XmlDocument();
      xmlDoc.LoadXml(producedXml);
```

5. Navigate to the tree-relevant parameters "Active", "File" and "Format"
6. Change values

```
        XPathNavigator navigator = xmlDoc.CreateNavigator();
        XPathNavigator navigator_Export = navigator.SelectSingleNode(@"/TreeItem/SafetyProjectDef/
        Commands/Export");
        if (navigator_Export == null) continue;
        XPathNavigator navigator_Export_Active = navigator_Export.SelectSingleNode("Active");
        if (navigator_Export_Active == null) continue;
        XPathNavigator navigator_Export_File = navigator_Export.SelectSingleNode("File");
        if (navigator_Export_File == null) continue;
        XPathNavigator navigator_Export_Format = navigator_Export.SelectSingleNode("Format");
        if (navigator_Export_Format == null) continue;
        navigator_Export_Active.SetValue(XmlConvert.ToString(true));
        navigator_Export_File.SetValue(string.Format(@"C:\{0}.xml", safetyProjectNode.Name));
        navigator_Export_Format.SetValue("TwinCAT 3 Xml Export Format");
```

7. Import the modified XML file to generate the exported XML file of the safety project

```
        safetyProjectNode.ConsumeXml(xmlDoc.InnerXml);
    }
  }
}
```

6.2.2 Import

This chapter shows a sample of what a programmatic application can look like in C# code.

Manual testing of the Automation Interface can be found in chapter Manual import [▶ 102].

Methods of the Automation Interface

When exporting with the Automation Interface, use the following methods:

- LookupTreeItem
- CreateChild

⇨ For more information on these methods, see the Beckhoff Information System.

Using

For this sample you need the following Using:

- TCatSysManagerLib

Proceed as follows to import a safety project:

1. Look up main item of the safety tree in TwinCAT 3 project

```
ITcSmTreeItem safety = systemManager.LookupTreeItem("TISC");
```

2. Import safety project

```
ITcSmTreeItem newProject = safety.CreateChild("NameOfProject", 0, null,
pathToProjectOrTfzipFile);
```

The information follows on "CreateChild":

"Project name", SubType, null, path file to the project

You can use 3 different values as SubType:

- 0: Copy project to solution directory
- 1: Move project to solution directory
- 2: Use original project location (if used, use "" as project name parameter)

You can choose between 3 different import formats as the source template for your safety project:

- *.splcproj file
- *.tfzip safety project archive file
- *.xml file

7 XSD file

An XML Schema Definition (XSD) defines the structure of XML files.

The XSD describes, among other things:

- Data types for elements and attributes
- XML Schema instances and instance groups
- Value ranges
- Sequences
- Minimum and optional information

If you create your own XML files with external tools, you can use the XSD file to ensure that your XML file is valid for TE9000. This ensures a correct import.

The XSD file is valid for XML version 0.31 or later. Validation of the XML file against the XSD is also only possible from XML version 0.31.

The following chapters describe the individual elements and attributes of the XSD file.

7.1 TwinCATExport

Fig. 1: TwinCATExport

Element	TwinCATExport
Contents	sophisticated
Description	Root node of the exported XML file

7.1.1 TwinCATExport attribute

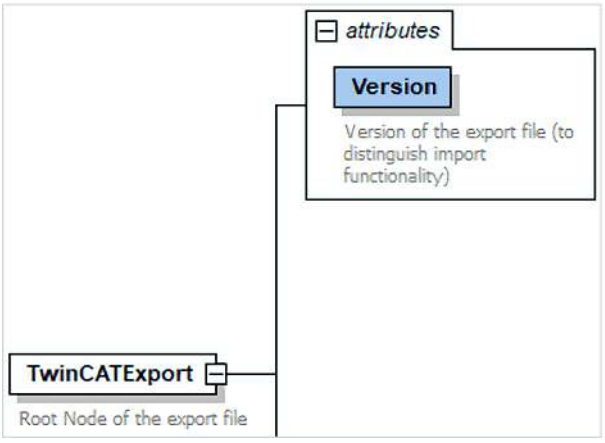


Fig. 2: TwinCATExport attribute

Attribute	Version
Data type	xs:string
Restriction	Fixed to used TE9000 version See chapter System requirements [▶ 15].
Use	required
Description	Version of the export file

As XML code, this looks as follows, for example:

```
<TwinCATExport Version="0.31">
</TwinCATExport>
```

7.1.2 BaseConfiguration

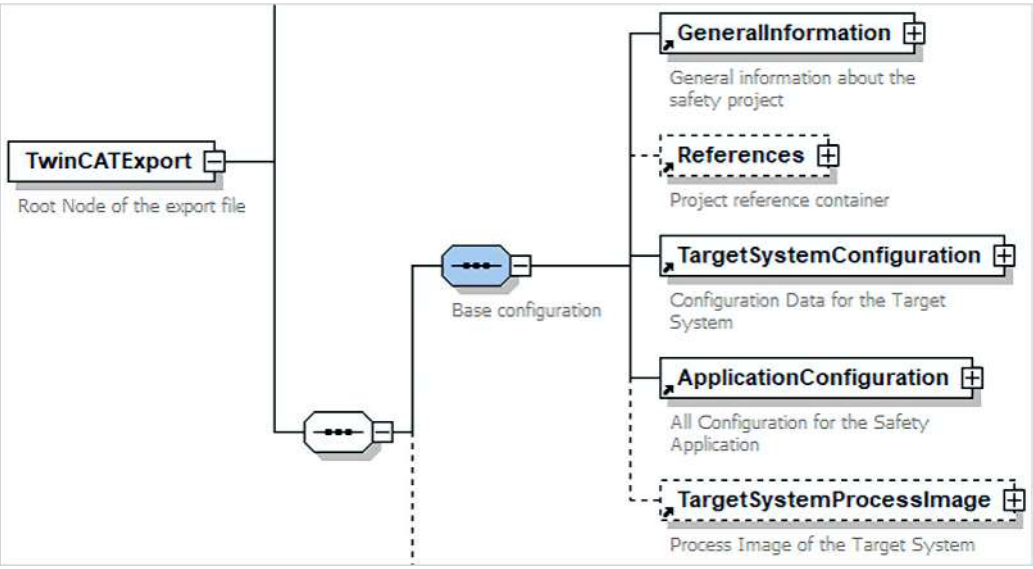


Fig. 3: Base configuration

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	GeneralInformation	References	TargetSystem-Configuration	Application-Configuration	TargetSystem-ProcessImage
minOcc	1	0	1	1	0
maxOcc	1	1	1	1	1
Contents	sophisticated	sophisticated	sophisticated	sophisticated	sophisticated
Description	General information about the safety project See chapter GeneralInformation [▶ 23].	References of the safety project See chapter References [▶ 24].	Configuration information for the target system See chapter TargetSystemConfiguration [▶ 25].	Complete configuration for the safety application See chapter ApplicationConfiguration [▶ 31].	Process image of the target system See chapter TargetSystemProcessImage [▶ 67].

As XML code, this looks as follows, for example:

```
<TwinCATExport Version="0.31">
  <GeneralInformation>
  </GeneralInformation>
  <References>
  </References>
  <TargetSystemConfiguration>
  </TargetSystemConfiguration>
  <ApplicationConfiguration>
  </ApplicationConfiguration>
  <TargetSystemProcessImage>
  </TargetSystemProcessImage>
</TwinCATExport>
```

7.1.2.1 GeneralInformation

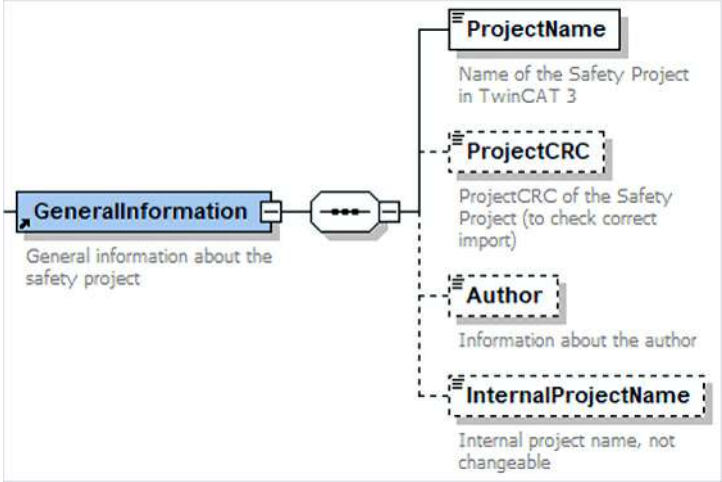


Fig. 4: GeneralInformation

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	ProjectName	ProjectCRC	Author	InternalProject-Name
minOcc	1	0	0	0
maxOcc	1	1	1	1
Data type	xs:string	HexDecUInt32Value	xs:string	xs:string
Contents	simple	simple	simple	simple
Description	Name of the safety project in the TE9000 project	CRC of the safety project	Information about the author who was specified when the project was created	Name of the safety project that was defined when the project was created

As XML code, this looks as follows, for example:

```
<GeneralInformation>
  <ProjectName>Example</ProjectName>
  <ProjectCRC>0xFB2D</ProjectCRC>
  <Author>Person1</Author>
  <InternalProjectName>Example</InternalProjectName>
</GeneralInformation>
```

7.1.2.2 References

You can currently reference the user function block libraries here.

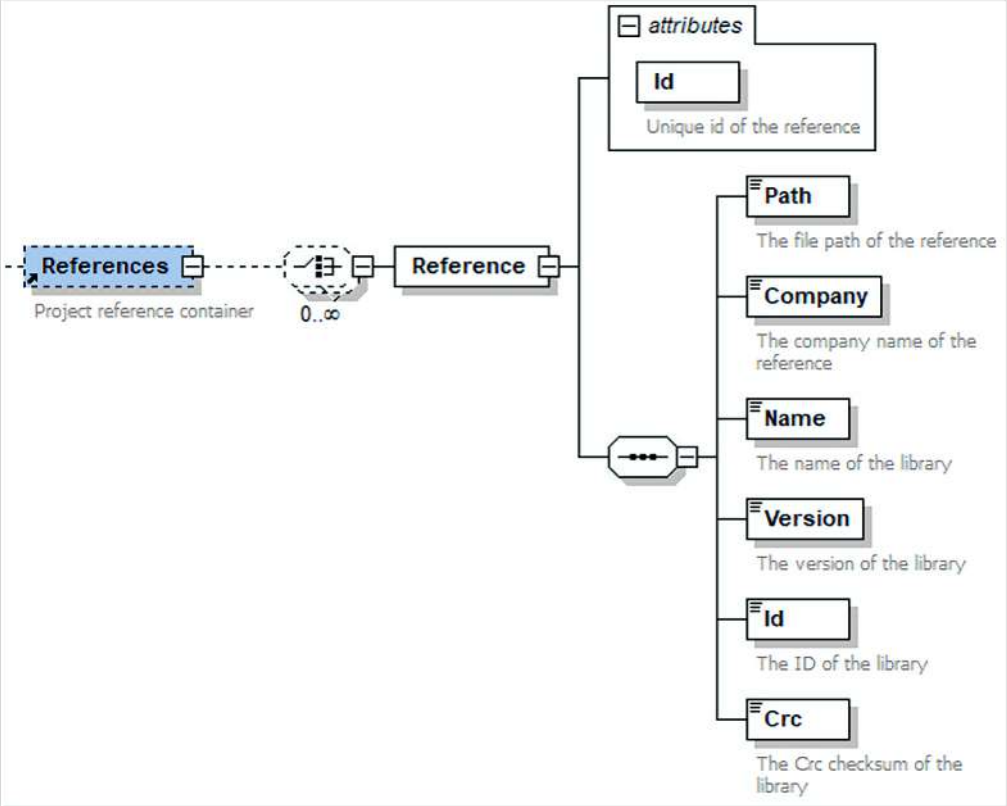


Fig. 5: References

Element	Reference
minOcc	1
maxOcc	1
Contents	sophisticated
Description	Project reference

Attribute	Id
Data type	xs:ID
Use	required
Description	Unique identification number of the reference See chapter ID rules [▶ 75].

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Path	Company	Name	Version	Id	Crc
minOcc	1	1	1	1	1	1
maxOcc	1	1	1	1	1	1
Data type	xs:string	xs:string	xs:string	xs:string	xs:string	xs:unsignedInt
Contents	simple	simple	simple	simple	simple	simple
Description	File path of the reference	Company name of the reference	Library name	Version of the library	ID of the library	CRC checksum of the library

As XML code, this looks as follows, for example:

```
<References>
  <Reference Id="r1">
    <Path>C:\TwinCAT\3.1\Components\Safety\Managed Libraries\Beckhoff\TestLib\1.0\TestLib.ufblib</
Path>
    <Company>Beckhoff</Company>
    <Name>TestLib</Name>
    <Version>1.0</Version>
    <Id>f7e6a540-3480-4c19-b0b9-a75d5d5100e9</Id>
    <Crc>802636370</Crc>
  </Reference>
</References>
```

7.1.2.3 TargetSystemConfiguration

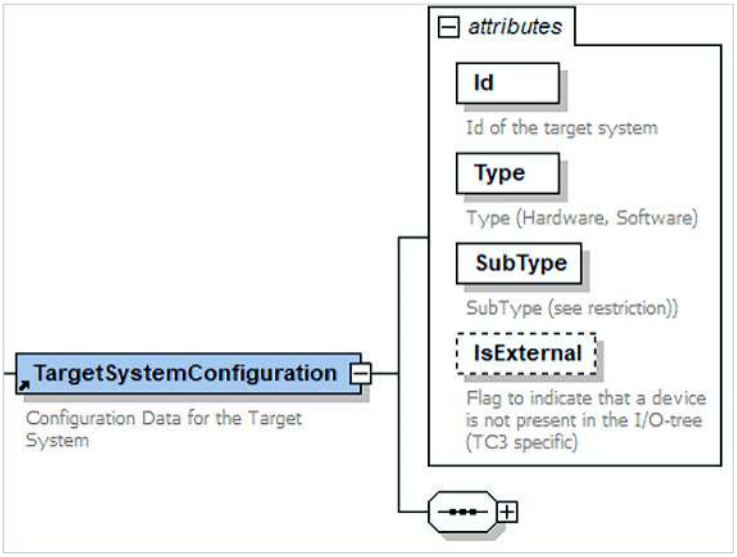


Fig. 6: TargetSystemConfiguration attribute

Attributes	Id	Type	SubType	IsExternal
Data type	xs:ID	xs:string	xs:string	xs:boolean
Use	required	required	required	required
Restriction	/	Hardware, Software	AMI891x, AMP891x, AX891x, EJ1914, EJ1918, EJ1957, EJ2914, EJ2918, EJ6910, EK1960, EL1918, EL2911, EL2912, EL6900, EL6910, EL6930, EP1918, EP1957, EP2918	/
Description	ID of the target system within the XML file The default ID is "t1". See chapter ID rules [▶ 75].	Target system type	Target system sub-type	Indicates that a device is not available in the I/O tree

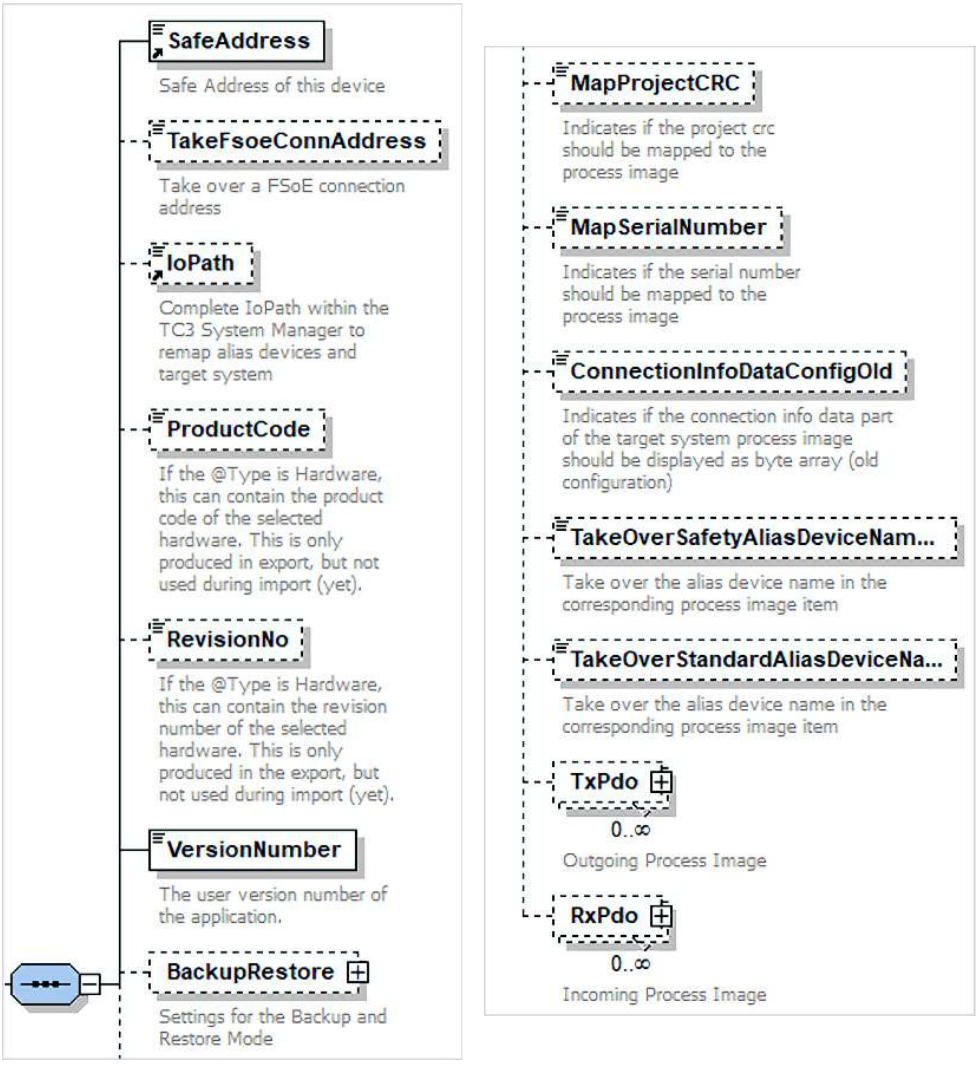


Fig. 7: TargetSystemConfiguration elements

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	SafeAddress	TakeFsoeConnAddress	IoPath	ProductCode	RevisionNo
minOcc	1	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:int	xs:boolean	xs:string	xs:int	xs:int
Contents	simple	simple	simple	simple	simple
Description	Safe address of the device The value range of the safe address depends on the device used.	Take over an FSoE address of a connection Currently not used.	Complete I/O path in TwinCAT 3 System Manger to remap alias devices and target systems	Shows the product code of the hardware used Is only used in export but not in import.	Shows the revision number of the hardware used Is only used in export but not in import.

Elements	VersionNumber	BackupRestore	MapProjectCRC	MapSerialNumber	ConnectioninfoDataConfigOld
minOcc	1	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedShort		xs:boolean	xs:boolean	xs:boolean
Contents	simple	sophisticated	simple	simple	simple

Elements	VersionNumber	BackupRestore	MapProjectCRC	MapSerialNum-ber	Connectioninfo-DataConfigOld
Description	Number of the user-specific version	Settings for the "Backup & Re-store" function See chapter BackupRestore [▶ 28].	Indicates whether the project CRC should be created in the process image	Indicates whether the serial number should be created in the process image	Indicates whether the part of the connection info data in the process image of the target system should be displayed as a byte arrangement

Elements	TakeOver-SafetyAliasDevice-NamesInProcessIm-age	TakeOverStandard-AliasDevice-NamesInProcessIm-age	TxPdo	RxPdo
minOcc	0	0	0	0
maxOcc	1	1	unlimited	unlimited
Data type	xs:boolean	xs:boolean	ProcessImage	ProcessImage
Contents	simple	simple	sophisticated	sophisticated
Description	Take over the alias device name in the process image of the target system	Take over the alias device name in the process image of the target system	Output process image See chapter TxPdo [▶ 29]. Currently not used.	Input process image See chapter RxPdo [▶ 31]. Currently not used.

As XML code, this looks as follows, for example:

```
<TargetSystemConfiguration Id="t1" Type="Hardware" SubType="AX891x" IsExternal="false">
  <SafeAddress>1</SafeAddress>
  <TakeFsoeConnAddress>false</TakeFsoeConnAddress>
  <IoPath>TIID^Device 2 (EtherCAT)^Term 30 (AX8620-0000-0104)^Drive 33 (AX8206-0210-0104)</IoPath>
  <ProductCode>537813010</ProductCode>
  <RevisionNo>6815954</RevisionNo>
  <VersionNumber>1</VersionNumber>
  <BackupRestore Activated="false" RestoreUserAdministration="false"
NumberOfDevicesWithMatchingCRC="0"/>
  <MapProjectCRC>false</MapProjectCRC>
  <MapSerialNumber>false</MapSerialNumber>
  <ConnectionInfoDataConfigOld>false</ConnectionInfoDataConfigOld>
  <TakeOverSafetyAliasDeviceNamesInProcessImage>false</TakeOverSafetyAliasDeviceNamesInProcessImage>
  <TakeOverStandardAliasDeviceNamesInProcessImage>false</
TakeOverStandardAliasDeviceNamesInProcessImage>
</TargetSystemConfiguration>
```

7.1.2.3.1 BackupRestore

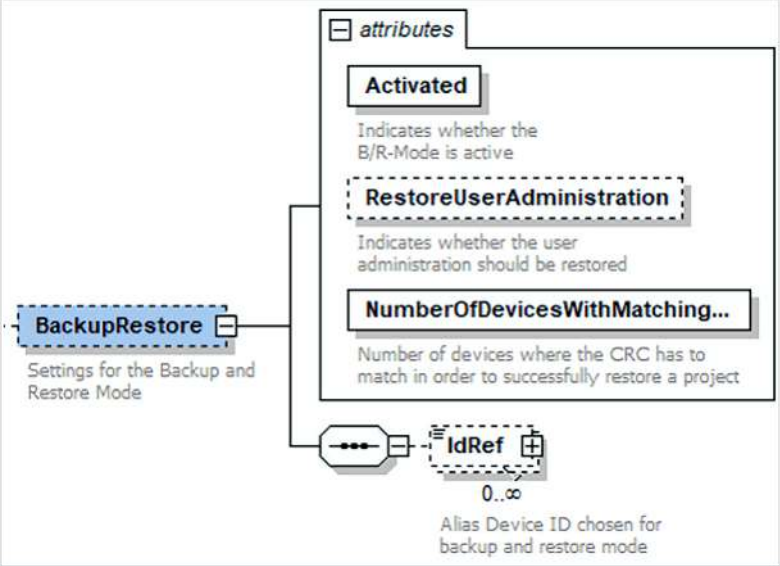


Fig. 8: TargetSystemConfiguration - BackupRestore

Attributes	Activated	RestoreUserAdministration	NumberOfDevicesWith-MatchingCRC
Data type	xs:boolean	xs:boolean	xs:int
Use	required	optional	required
Description	Indicates whether the Backup&Restore function is active	Indicates whether the user administration should be restored	Number of devices where the CRC has to match in order to successfully restore a project

For a valid XML file, the order of the elements must be strictly adhered to.

Element	IdRef
minOcc	0
maxOcc	unlimited
Data type	xs:IDREF
Contents	sophisticated
Description	The identification number of the alias device chosen for the Backup&Restore function See ID attribute in chapter SafetyAliasDevice [▶ 57].

As XML code, this looks as follows, for example:

```
<BackupRestore Activated="true" RestoreUserAdministration="false"
NumberOfDevicesWithMatchingCRC="2">
  <IdRef StoreProjectCrcInSlave="true" StoreSlaveProjectCrcInMaster="true"
StoreMasterProjectCrcInSlave="false" ReadProjectCrcFromMaster="false">g1_a3</IdRef>
  <IdRef StoreProjectCrcInSlave="true" StoreSlaveProjectCrcInMaster="false"
StoreMasterProjectCrcInSlave="false" ReadProjectCrcFromMaster="false">g1_a6</IdRef>
</BackupRestore>
```

7.1.2.3.1.1 IdRef

Fig. 9: BackupRestore - IdRef

Attributes	StorePro-jectCrcInSlave	StoreSlavePro-jectCrcInMaster	StoreMasterPro-jectCrcInSlave	ReadProjectCr-cFromMaster
Data type	xs:boolean	xs:boolean	xs:boolean	xs:boolean
Use	required	required	required	required
Description	Indicates whether the project CRC is to be stored in the slave	Indicates whether the project CRC of the slave should be stored in the master	Indicates whether the project CRC of the master should be stored in the slave	Indicates whether the project CRC is to be read by the master

7.1.2.3.2 TxPdo

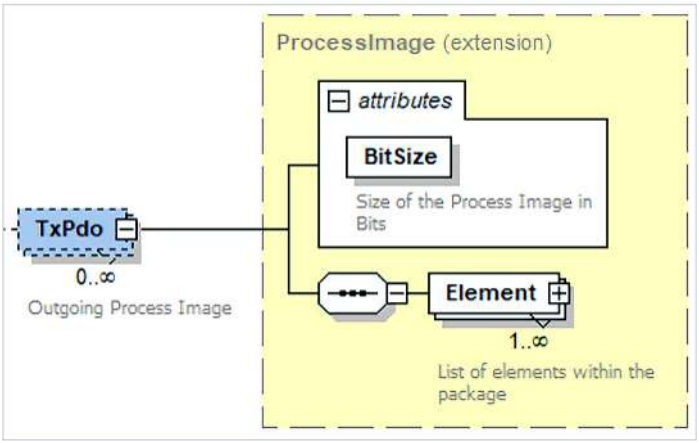


Fig. 10: TargetSystemConfiguration - TxPdo

Attribute	BitSize
Data type	xs:int
Use	required
Description	Size of the process image in bits

Element	Element
minOcc	1
maxOcc	unlimited
Contents	sophisticated
Description	List of elements

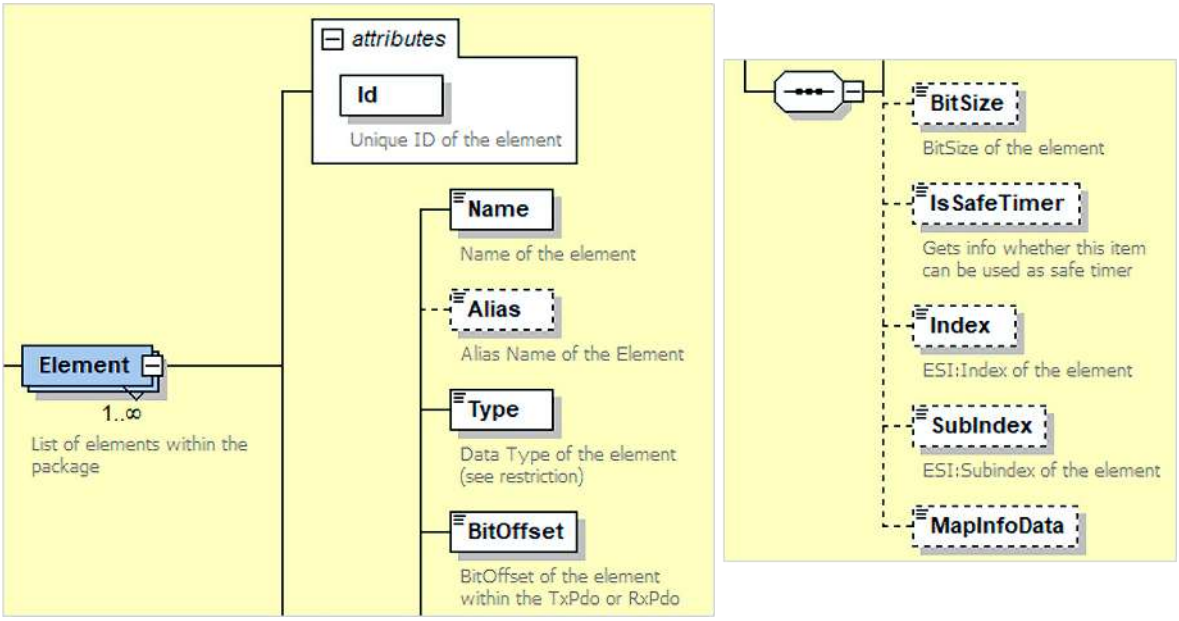


Fig. 11: TxPdo – element

Attribute	Id
Data type	xs:ID
Use	required
Description	Unique ID of the element The ID depends on the context, see chapter ID rules [▶ 75].

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Name	Alias	Type	BitOffset	BitSize
minOcc	1	0	1	1	0
maxOcc	1	1	1	1	1
Data type	xs:string	xs:string	xs:string	xs:int	xs:int
Contents	simple	simple	simple	simple	simple
Restriction	/	/	BIT, BYTE, USINT, UINT, INT, UDINT, DINT	/	/
Description	Name of the element	Alias name of the element	Data type of the element	Bit offset of the element in the process image	Bit size of the element

Elements	IsSafeTimer	Index	SubIndex	MapInfoData
minOcc	0	0	0	0
maxOcc	1	1	1	1
Data type	xs:boolean	/	/	xs:boolean
Contents	simple	/	/	simple
Description	Indicates whether the element can be used as a safe timer	Index from ESI device description	Sub-index from ESI device description	Take over as information data in I/O process image

As XML code, this looks as follows, for example:

```
<TxPdo BitSize="4096">
  <Element Id="g1_a2_lo1">
    <Name>ChA_STO_1</Name>
    <Type>BIT</Type>
    <BitOffset>0</BitOffset>
    <BitSize>1</BitSize>
```

```
    <IsSafeTimer>false</IsSafeTimer>
    <Index>49281</Index>
    <SubIndex>1</SubIndex>
    <MapInfoData>false</MapInfoData>
  </Element>
  <Element Id="g1_a2_lo2">
    <Name>ChA_STO_2</Name>
    <Type>BIT</Type>
    <BitOffset>1</BitOffset>
    <BitSize>1</BitSize>
    <IsSafeTimer>false</IsSafeTimer>
    <Index>49281</Index>
    <SubIndex>2</SubIndex>
    <MapInfoData>false</MapInfoData>
  </Element>
  ...
</TxPdo>
```

7.1.2.3.3 RxPdo

The input process image "RxPdo" corresponds to the output process image "TxPdo" in terms of its attributes and elements. For more information, see the chapter TxPdo [▶ 29].

As XML code, this looks as follows, for example:

```
<RxPdo BitSize="48">
  <Element Id="g1_a3_i1">
    <Name>Safe Data Byte 0[0]</Name>
    <Type>BIT</Type>
    <BitOffset>8</BitOffset>
    <BitSize>1</BitSize>
    <IsSafeTimer>false</IsSafeTimer>
  </Element>
  <Element Id="g1_a3_i2">
    <Name>Safe Data Byte 0[1]</Name>
    <Type>BIT</Type>
    <BitOffset>9</BitOffset>
    <BitSize>1</BitSize>
    <IsSafeTimer>false</IsSafeTimer>
  </Element>
  ...
</RxPdo>
```

7.1.2.4 ApplicationConfiguration

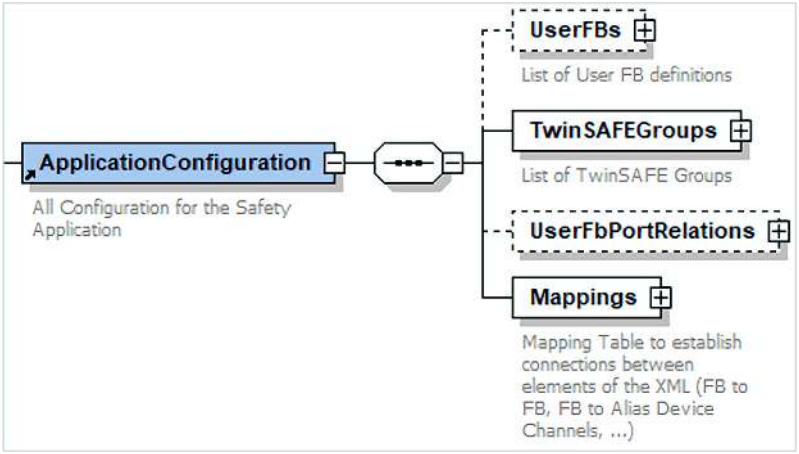


Fig. 12: ApplicationConfiguration

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	UserFBs	TwinSAFEGroups	UserFbPortRelations	Mappings
minOcc	0	1	0	1
maxOcc	1	1	1	1
Contents	sophisticated	sophisticated	sophisticated	sophisticated

Elements	UserFBs	TwinSAFEGroups	UserFbPortRela-tions	Mappings
Description	List of user function block definitions See chapter UserFbs [▶ 32].	List of TwinSAFE groups See chapter TwinSAFEGroups [▶ 51].	Relations between the user function block port identification numbers and the function block port identification numbers See chapter UserFbPortRelations [▶ 65].	Mapping table to establish connections between the elements of the XML file (between different function blocks, between a function block and the channel of an alias device) See chapter Mappings [▶ 66].

As XML code, this looks as follows, for example:

```
<ApplicationConfiguration>
  <UserFBs>
  </UserFBs>
  <TwinSAFEGroups>
  </TwinSAFEGroups>
  <UserFbPortRelations>
  </UserFbPortRelations>
  <Mappings>
  </Mappings>
</ApplicationConfiguration>
```

7.1.2.4.1 UserFbs

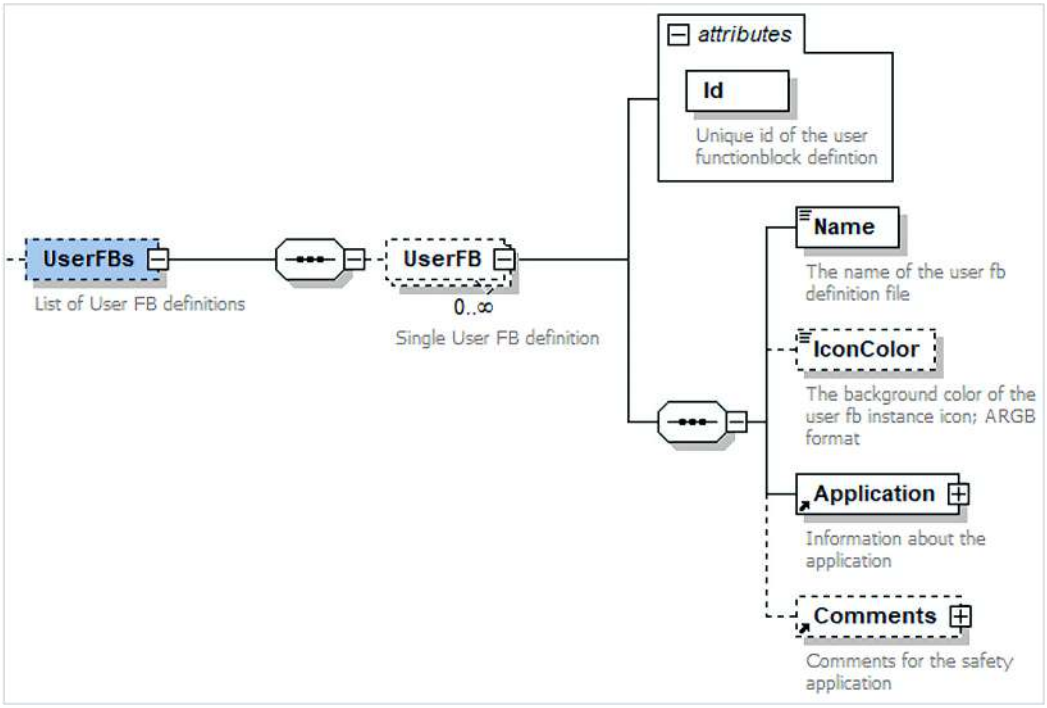


Fig. 13: ApplicationConfiguration - UserFBs

Element	UserFB
minOcc	0
maxOcc	unlimited
Contents	sophisticated
Description	Single user function block definition

For a valid XML file, the order of the elements must be strictly adhered to.

Attribute	Id
Data type	xs:ID
Use	required
Description	Unique ID of the user function block definition See chapter ID rules [▶ 75].

Elements	Name	IconColor	Application	Comments
minOcc	1	0	1	0
maxOcc	1	1	1	1
Contents	simple	simple	sophisticated	sophisticated
Description	File name of the user function block definition	Background color of the user function block instance icon Format: ARGB	Information about the application	Comments for the safety application

As XML code, this looks as follows, for example:

```
<UserFBs>
  <UserFB Id="u1">
    <Name>UserFB1</Name>
    <IconColor>-990068</IconColor>
    <Application>
    </Application>
    <Comments>
    </Comments>
  </UserFB>
</UserFBs>
```

7.1.2.4.1.1 Application

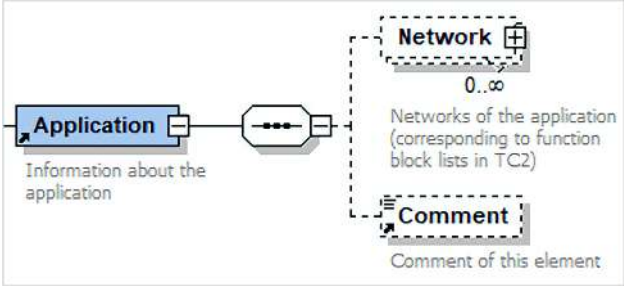


Fig. 14: Application

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Network	Comment
minOcc	0	0
maxOcc	unlimited	1
Data type	/	xs:string
Contents	sophisticated	simple
Description	Networks of the application (corresponding to the function block lists inTwinCAT 3) See chapter Network [▶ 34].	Comment of this element

As XML code, this looks as follows, for example:

```
<Application>
  <Network Id="u1_n1" OrderId="1">
  </Network>
  <Comment>ScalingOffset20</Comment>
</Application>
```

7.1.2.4.1.1.1 Network

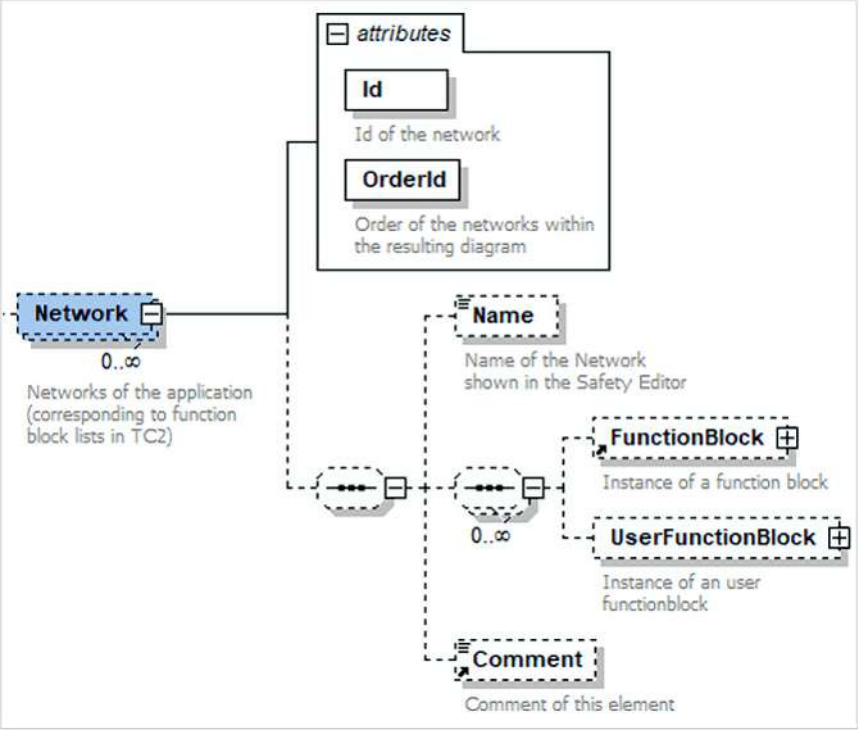


Fig. 15: Network

Attributes	Id	OrderId
Data type	xs:ID	xs:int
Use	required	required
Description	ID of the network See chapter ID rules [▶ 75].	Order of the networks in the safety application diagram

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Name	FunctionBlock	UserFunctionBlock	Comment
minOcc	0	0	0	0
maxOcc	1	1	1	1
Data type	xs:string	/	/	xs:string
Contents	simple	sophisticated	sophisticated	simple
Description	Name of the Network shown in the Safety Editor	Instance of a function block See chapter FunctionBlock [▶ 35].	Instance of a user function block See chapter UserFunctionBlock [▶ 43].	Comment of this element

As XML code, this looks as follows, for example:

```
<Network Id="g1_n1" OrderId="1">
  <Name>Network1</Name>
  <FunctionBlock Id="g1_n1_f1" OrderId="1">
  </FunctionBlock>
  <UserFunctionBlock Id="g1_n1_u2" MinOrderId="2" MaxOrderId="4">
  </UserFunctionBlock>
  <Comment>Test Network1</Comment>
</Network>
```

7.1.2.4.1.1.1.1 FunctionBlock

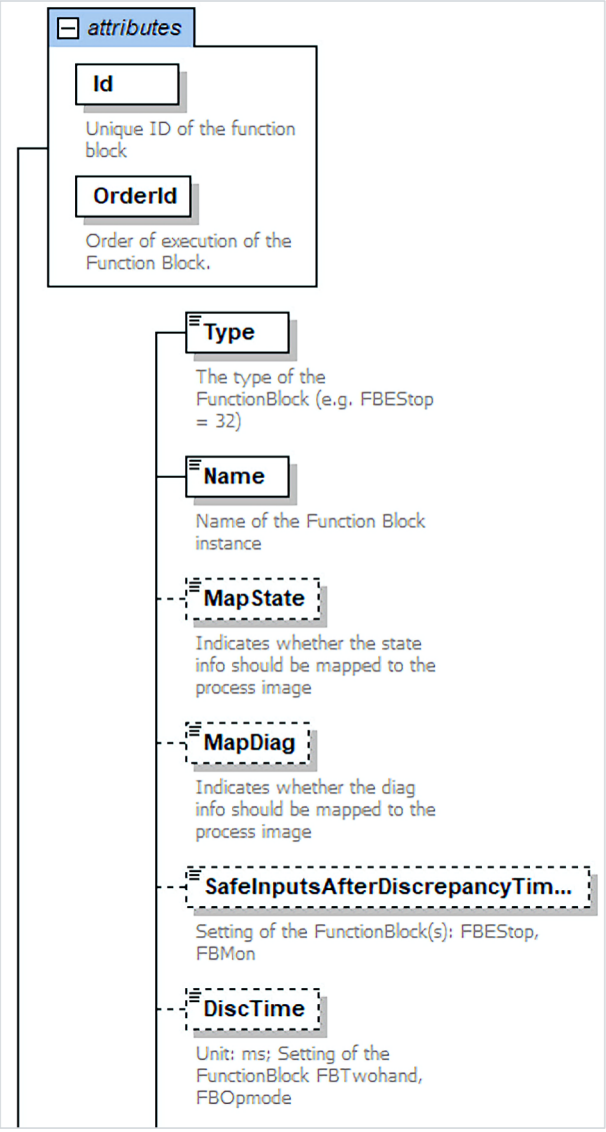


Fig. 16: FBFunctionBlock (excerpt)

For more information about function blocks, see document [4] in chapter References [▶ 7].

Attributes	Id	OrderId
Data type	xs:ID	xs:unsignedInt
Use	required	required
Description	ID of the function block See chapter ID rules [▶ 75].	Order of execution of the function blocks

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Type	Name	MapState	MapDiag	SafeInputsAf-terDiscrepancy-TimeError
minOcc	1	1	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedInt	xs: string	xs:boolean	xs:boolean	xs:boolean
Contents	simple	simple	simple	simple	simple

Elements	Type	Name	MapState	MapDiag	SafeInputsAfterDiscrepancyTimeError
Description	Type of function block See chapter FBTypes [▶ 77].	Name of the function block instance	Indicates whether the state information is to be mapped to the process image of the target system	Indicates whether the diag information is to be mapped to the process image of the target system	Setting the FBEstop and FBMon function blocks

Elements	DiscTime	OutputDelayTime	ManualRestartActive	ManualFunctionTestActive	SwitchOffTime
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:int	xs:int	xs:boolean	xs:boolean	xs:int
Contents	simple	simple	simple	simple	simple
Description	Setting of the function blocks FBTwohand and FBOpMode; unit: ms	Setting of the function blocks FBEstop, FBMon and FBMuting; unit: ms	Setting of the function blocks FBMon and FBOpMode	Setting of the function block FBMon	Setting of the function block FBEdm; unit: ms

Elements	SwitchOnTime	SequentialInputsActive	FilterTime	MaxMutingTime	BackwardsMode
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:int	xs:boolean	xs:int	xs:int	xs:boolean
Simple	simple	simple	simple	simple	simple
Description	Setting of the function block FBEdm; unit: ms	Setting of the function block FBMuting	Setting of the function block FBMuting; unit: ms	Setting of the function block FBMuting; unit: ms	Setting of the function block FBMuting

Elements	TofTime	TonTime	ConnId	CompareMode	ToleranceTime
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:float	xs:float	xs:int	xs:int	xs:unsignedInt
Contents	simple	simple	simple	simple	simple
Restriction	/	/	/	0 = 1oo2 1 = 2oo3 2 = 3oo5	/
Description	Setting of the function block FBEdm; unit: ms	Setting of the function block FBMuting;	Setting of the function block FBMuting; unit: ms	Setting of the function block FBMuting; unit: ms	Setting of the function block FBMuting

Elements	AllowedDeviation	DivisionRounding	PresetValue	CounterLimit	MinValue
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedInt	xs:integer	xs:long	xs:long	xs:long
Contents	simple	simple	simple	simple	simple

Elements	AllowedDeviation	DivisionRounding	PresetValue	CounterLimit	MinValue
Description	Setting of the function block FBCompare	Setting of the function blocks FBDiv and FBScale	Setting of the function block FBCounter	Setting of the function blocks FBCounter and FBViolationCnt	Setting of the function block FBLimit

Elements	MaxValue	MultiplicationFirst	ScalingFactorMul	ScalingFactorDiv	ScalingOffset
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:long	xs:boolean	xs:unsignedShort	xs:unsignedShort	xs:int
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBLimit	Setting of the function block FBScaling	Setting of the function block FBScaling	Setting of the function block FBScaling	Setting of the function block FBScaling

Elements	MoveDetectionTime	PressStartDelayTime	MinPositionChange	AllowedPositionJitter	ExcentricMode
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedShort	xs:unsignedShort	xs:unsignedInt	xs:unsignedShort	xs:boolean
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor; True = ExcentricMode False = PendulumMode

Elements	StopDetectionTime	MaxPosition	TDC1LowerLimit	TDC1UpperLimit	TDC2LowerLimit
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedShort	xs:unsignedInt	xs:long	xs:long	xs:long
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor

Elements	TDC2UpperLimit	BDCLowerLimit	BDCUpperLimit	OverrunMax	Overflow
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:long	xs:long	xs:long	xs:long	xs:boolean
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBCamMonitor	Setting of the function block FBSpeed

Elements	TimeInterval	Outside	LoadSensingTable	IncValue	DecValue
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedShort	xs:boolean	/	xs:unsignedShort	xs:unsignedShort
Contents	simple	simple	sophisticated	simple	simple
Description	Setting of the function block FBSpeed	Setting of the function block FBLoadSensing	Setting of the function block FBLoadSensing See chapter LoadSensingTable [▶ 40].	Setting of the function block FBViolationCnt	Setting of the function block FBViolationCnt

Elements	LimitPos	LimitNeg	TargetValue	Offset	MaxTime
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedInt	xs:unsignedInt	xs:unsignedInt	xs:unsignedShort	xs:unsignedShort
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBSLI	Setting of the function block FBSLI	Setting of the function block FBEnvelope	Setting of the function block FBEnvelope	Setting of the function block FBEnvelope

Elements	TimeAfterInTarget	OSSDCheckOptional	Watchdog	StartsWithRemainingTime	OverflowAllowed
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedShort	xs:boolean	xs:unsignedShort	xs:boolean	xs:boolean
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBEnvelope	Setting of the function block FBMuting	Setting of the function block FBScaling	Setting of the function block FBTon2	Setting of the function block FBSLI2

Elements	ExceededSupport	MultiturnMode	SingleturnPosLL	SingleturnPosUL	MultiturnPosLL
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:boolean	xs:boolean	xs:unsignedInt	xs:unsignedInt	xs:int
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBSLI2	Setting of the function block FBSLP	Setting of the function block FBSLP	Setting of the function block FBSLP	Setting of the function block FBSLP

Elements	MultiturnPosUL	ScalingFactorNuminator	ScalingFactorDenominator	ScalingUnit	EncoderMask
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:int	xs:double	xs:double	xs:string	xs:unsignedLong
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBSLP	Setting of the function block FBSLP	Setting of the function block FBSLP	Setting of the function block FBSLP	Setting of the function block FBSLP

Elements	EncoderSub-Mask	BrakeTestInter-val	MinCurrentIq	MaxPositionDe-viation	MinDuration
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedLong	xs:unsignedInt	xs:int	xs:unsignedInt	xs:unsigned-Short
Contents	simple	simple	simple	simple	simple
Description	Setting of the function block FBSLP	Setting of the function block FBSBT	Setting of the function block FBSBT	Setting of the function block FBSBT	Setting of the function block FBSBT

Elements	MaxDuration	Feedback	Outports	Inports	Comment
minOcc	0	0	1	1	0
maxOcc	1	1	1	1	1
Data type	xs:unsignedShort	xs:unsigned short	FBPorts	FBPorts	xs:string
Contents	simple	simple	sophisticated	sophisticated	simple
Description	Setting of the function block FBSBT	Setting of the function block FBAdvPosMon: ChA Primary Feedback = 0; ChA Secondary Feedback = 1; ChB Primary Feedback = 2; ChB Secondary Feedback = 3	Output process image See chapter Outports [▶ 41].	Input process image See chapter Inports [▶ 43].	Comment of this element

As XML code, this looks as follows, for example:

```
<FunctionBlock Id="u1_n1_f1" OrderId="1">
  <Type>71</Type>
  <Name>FBScaling1</Name>
   <MapState>false</MapState>
  <MapDiag>false</MapDiag>
  <DivisionRounding>2</DivisionRounding>
  <MultiplicationFirst>true</MultiplicationFirst>
  <ScalingFactorMul>1</ScalingFactorMul>
  <ScalingFactorDiv>1</ScalingFactorDiv>
  <ScalingOffset>20</ScalingOffset>
  <Watchdog>0</Watchdog>
  <Outports>
  </Outports>
  <Inports>
  </Inports>
</FunctionBlock>
```

7.1.2.4.1.1.1.1 LoadSensingTable

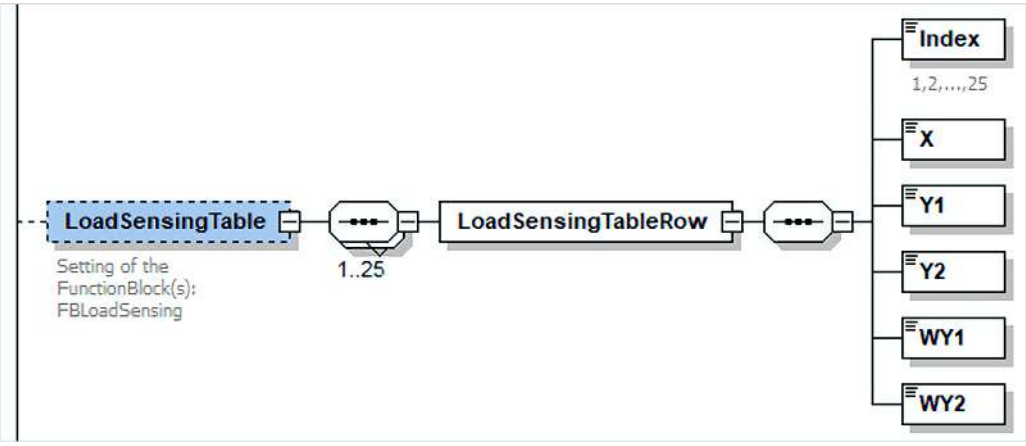


Fig. 17: LoadSensingTable

For a valid XML file, the order of the elements must be strictly adhered to.

Element	LoadSensingTableRow
minOcc	1
maxOcc	1
Data type	/
Contents	sophisticated
Description	Settings of the function block FBLoadSensing - table row

Elements	Index	X	Y1
minOcc	1	1	1
maxOcc	1	1	1
Data type	xs:int	xs:long	xs:long
Contents	simple	simple	simple

Elements	Y2	WY1	WY2
minOcc	1	1	1
maxOcc	1	1	1
Data type	xs:long	xs:long	xs:long
Contents	simple	simple	simple

As XML code, this looks as follows, for example:

```
<LoadSensingTable>
  <LoadSensingTableRow>
    <Index>1</Index>
    <X>0</X>
    <Y1>4</Y2>
    <Y2>14</Y2>
    <WY1>5</WY1>
    <WY2>13</WY2>
  </LoadSensingTableRow>
  <LoadSensingTableRow>
    <Index>2</Index>
    <X>1</X>
    <Y1>5</Y2>
    <Y2>15</Y2>
    WY1>6</WY1>
    <WY2>14</WY2>
  </LoadSensingTableRow>
  ...
  <LoadSensingTableRow>
    <Index>25</Index>
    <X>0</X>
    <Y1>0</Y2>
    <Y2>0</Y2>
    <WY1>0</WY1>
```

```
    <WY2>0</WY2>
  </LoadSensingTableRow>
</LoadSensingTable>
```

7.1.2.4.1.1.1.2 Outports

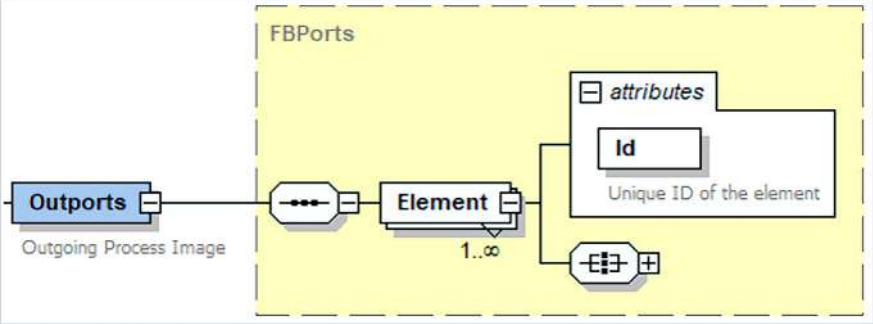


Fig. 18: Outports

For a valid XML file, the order of the elements must be strictly adhered to.

Element	Element
minOcc	1
maxOcc	unlimited
Contents	sophisticated
Description	Element in the output process image

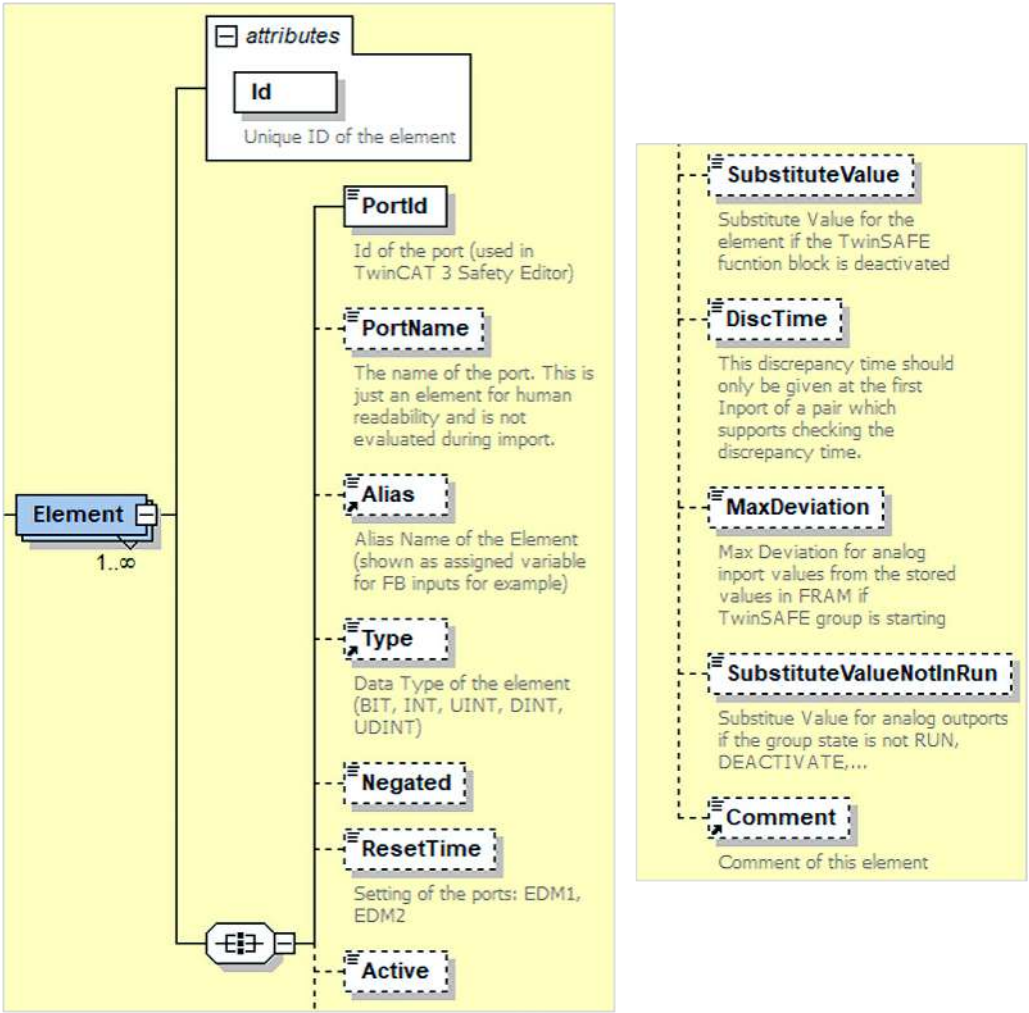


Fig. 19: Outport element

Attribute	Id
Data type	xs:ID
Use	required
Description	ID of an output element See chapter ID rules [▶ 75].

Elements	PortId	PortName	Alias	Type
minOcc	1	0	0	0
maxOcc	1	1	1	1
Data type	HexDecUInt32Value	xs:string	xs:string	xs:string
Contents	simple	simple	simple	simple
Restriction	/	/	/	BIT, BYTE, USINT, UINT, INT, UDINT, DINT
Description	ID of the port (used in the TwinCAT 3 Safety Editor) See chapter PortIds [▶ 78].	Port name This is not evaluated during import, but is only used for readability	Alias name of the element (shown as an assigned variable for function block inputs, for example)	Data type of the element

Elements	Negated	ResetTime	Active	SubstituteValue
minOcc	0	0	0	0
maxOcc	1	1	1	1
Data type	xs:boolean	xs:unsignedShort	xs:boolean	/
Contents	simple	simple	simple	simple
Description		Port settings: EDM1, EDM2		Substitute value in the event that the TwinSAFE group is deactivated

Elements	DiscTime	MaxDeviation	SubstituteValueNot-InRun	Comment
minOcc	0	0	0	0
maxOcc	1	1	1	1
Data type	xs:int	xs:unsignedShort	xs:long	xs:string
Contents	simple	simple	simple	simple
Description	This discrepancy time should only be specified on the first input of a pair that supports the discrepancy time check	maximum deviation for analog input values from the values stored in the permanent memory when the TwinSAFE group starts	Substitute values for analog outputs if the group state is not RUN, DEACTIVATE,..	Comment of this element

As XML code, this looks as follows, for example:

```
<Outports>
  <Element Id="u1_n1_f1_o1">
    <PortId>33554503</PortId>
    <PortName>Error</PortName>
    <Type>BIT</Type>
    <SubstituteValue>0</SubstituteValue>
    <SubstituteValueNotInRun>0</SubstituteValueNotInRun>
  </Element>
  <Element Id="u1_n1_f1_o2">
    <PortId>33620039</PortId>
    <PortName>StuckAtError</PortName>
    <Type>BIT</Type>
    <SubstituteValue>0</SubstituteValue>
    <SubstituteValueNotInRun>0</SubstituteValueNotInRun>
```

```
  </Element>
  <Element Id="u1_n1_f1_o3">
    <PortId>34603079</PortId>
    <PortName>AnalogOut</PortName>
    <Alias>scaleOut</Alias>
    <Type>UINT</Type>
    <SubstituteValue>10</SubstituteValue>
    <SubstituteValueNotInRun>15</SubstituteValueNotInRun>
  </Element>
</Outports>
```

7.1.2.4.1.1.1.3 Inports

The input process image "Inports" corresponds to the output process image "Outports" in terms of its attributes and elements. For more information, see the chapter Outports [▶ 41].

As XML code, this looks as follows, for example:

```
<Inports>
  <Element Id="u1_n1_f1_i1">
    <PortId>17825863</PortId>
    <PortName>AnalogIn</PortName>
    <Alias>scaleIn</Alias>
    <Type>UINT</Type>
    <Negated>false</Negated>
    <Active>true</Active>
    <MaxDeviation>0</MaxDeviation>
  </Element>
</Inports>
```

7.1.2.4.1.1.1.2 UserFunctionBlock

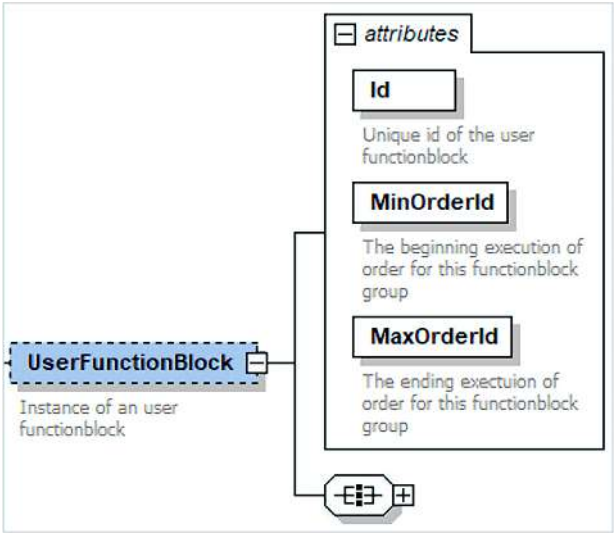


Fig. 20: UserFunctionBlock attributes

Attributes	Id	MinOrderId	MaxOrderId
Data type	xs:ID	xs:unsignedInt	xs:unsignedInt
Use	required	required	required
Description	ID of the function block See chapter ID rules [▶ 75].	Minimum Order of Execution	Maximum Order of Execution

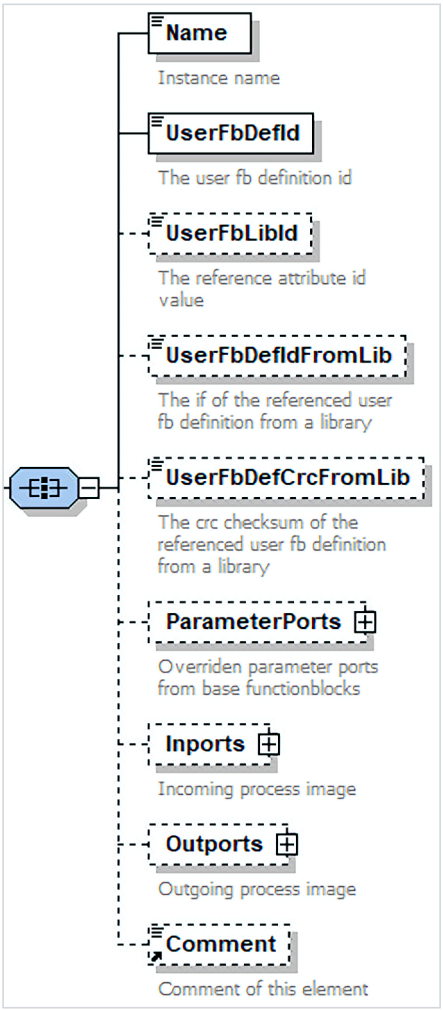


Fig. 21: UserFunctionBlock elements

Elements	Name	UserFbDefId	UserFbLibId	UserFbDefId-FromLib	UserFbDefCr-cFromLib
minOcc	1	1	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:string	xs:string	xs:string	xs:string	xs:unsignedInt
Contents	simple	simple	simple	simple	simple
Description	Name of the instance	ID of the user function block Is formed from Reference.Id + "_" UserFB.Id (example: r1_u1) See Reference.Id attribute in chapter References [▶ 24] and UserFB.Id attribute in chapter UserFbs [▶ 32].	ID of the reference attribute See Reference.Id attribute in chapter References [▶ 24].	ID of the referenced user function block definition from a library	CRC checksum of the referenced user function block definition from a library

Elements	ParameterPorts	Inports	Outports	Comment
minOcc	0	0	0	0
maxOcc	1	1	1	1

Elements	ParameterPorts	Inports	Outports	Comment
Data type	/	/	/	xs:string
Contents	sophisticated	sophisticated	sophisticated	simple
Description	Overridden parameter ports from the basic function block See chapter ParameterPorts [▶ 46].	Input process image See chapter Inports [▶ 47].	Output process image See chapter Outports [▶ 48].	Comment of this element

As XML code, this looks as follows, for example:

```
<UserFunctionBlock Id="g1_n1_u2" MinOrderId="2" MaxOrderId="4">
  <Name>UserFB</Name>
  <UserFbDefId>r1_u1</UserFbDefId>
  <UserFbLibId>r1</UserFbLibId>
  <UserFbDefIdFromLib>88e7f0fe-8ac3-4eab-b02e-cb7df03efbdb</UserFbDefIdFromLib>
  <UserFbDefCrcFromLib>3494585285</UserFbDefCrcFromLib>
  <ParameterPorts>
  </ParameterPorts>
  <Inports>
  </Inports>
  <Outports>
  </Outports>
  <Comment>User FB Test Ton And</Comment>
</UserFunctionBlock>
```

7.1.2.4.1.1.1.2.1 ParameterPorts

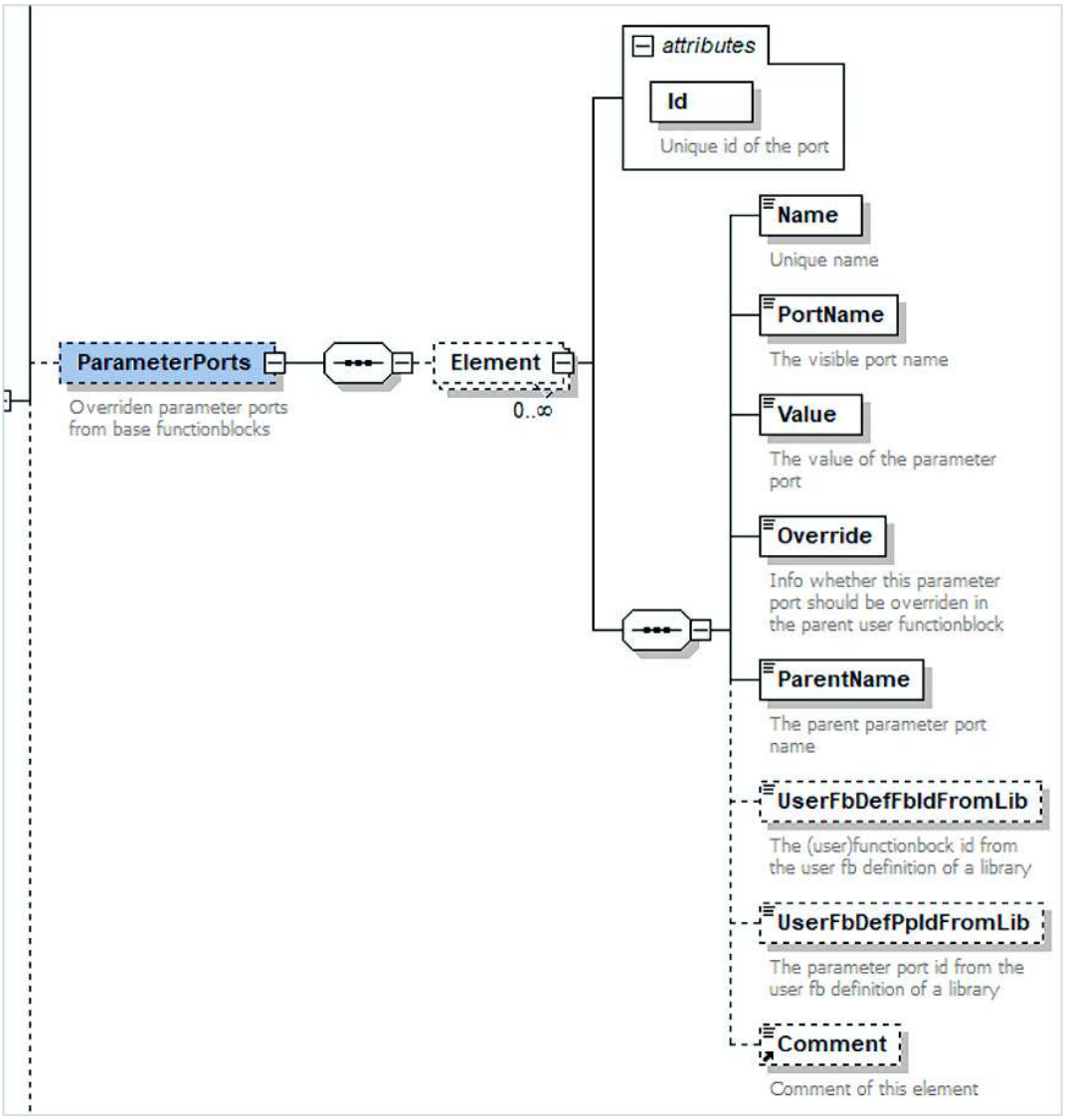


Fig. 22: ParameterPorts

Attributes	Id
Data type	xs:ID
Use	required
Description	ID of the port See chapter ID rules [▶ 75].

Elements	Name	PortName	Value	Override
minOcc	1	1	1	1
maxOcc	1	1	1	1
Data type	xs:string	xs:string	xs:string	xs:boolean
Contents	simple	simple	simple	simple
Description	Unique name	Visible name of the port	Value of the parameter port	Info whether this parameter port should be overridden in the parent user function block

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	ParentName	UserFbDefFbId-FromLib	UserFbDefPpId-FromLib	Comment
minOcc	1	0	0	0
maxOcc	1	1	1	1
Data type	xs:string	xs:string	xs:string	xs:string
Contents	simple	simple	simple	simple
Description	Name of the parent parameter port	the (user) function block ID from the user function block definition of a library	the parameter port ID from the user function block definition of a library	Comment of this element

As XML code, this looks as follows, for example:

```
<ParameterPorts>
  <Element Id="g1_n1_u2_p1">
    <Name>Network1_FBTon1_Delay Time (s)</Name>
    <PortName>FBTon1_Delay Time (s)</PortName>
    <Value>2</Value>
    <Override>true</Override>
    <ParentName>FBTon1_Delay Time (s)</ParentName>
    <UserFbDefFbIdFromLib>6d61e205-2f4c-4988-8f9d-d0a517134733</UserFbDefFbIdFromLib>
    <UserFbDefPpIdFromLib>aaffefce-ed0a-45ce-aa16-bdee52f4cd58</UserFbDefPpIdFromLib>
  </Element>
  <Element Id="g1_n1_u2_p2">
    <Name>Network1_FBTon11_Delay Time (s)</Name>
    <PortName>FBTon2_Delay Time (s)</PortName>
    <Value>5</Value>
    <Override>true</Override>
    <ParentName>FBTon2_Delay Time (s)</ParentName>
    <UserFbDefFbIdFromLib>68bc8331-0910-4cb1-946f-b4e78cee5a22</UserFbDefFbIdFromLib>
    <UserFbDefPpIdFromLib>97c4e7f3-e650-4da6-90bb-60b7fb549555</UserFbDefPpIdFromLib>
  </Element>
</ParameterPorts>
```

7.1.2.4.1.1.2.2 Inports

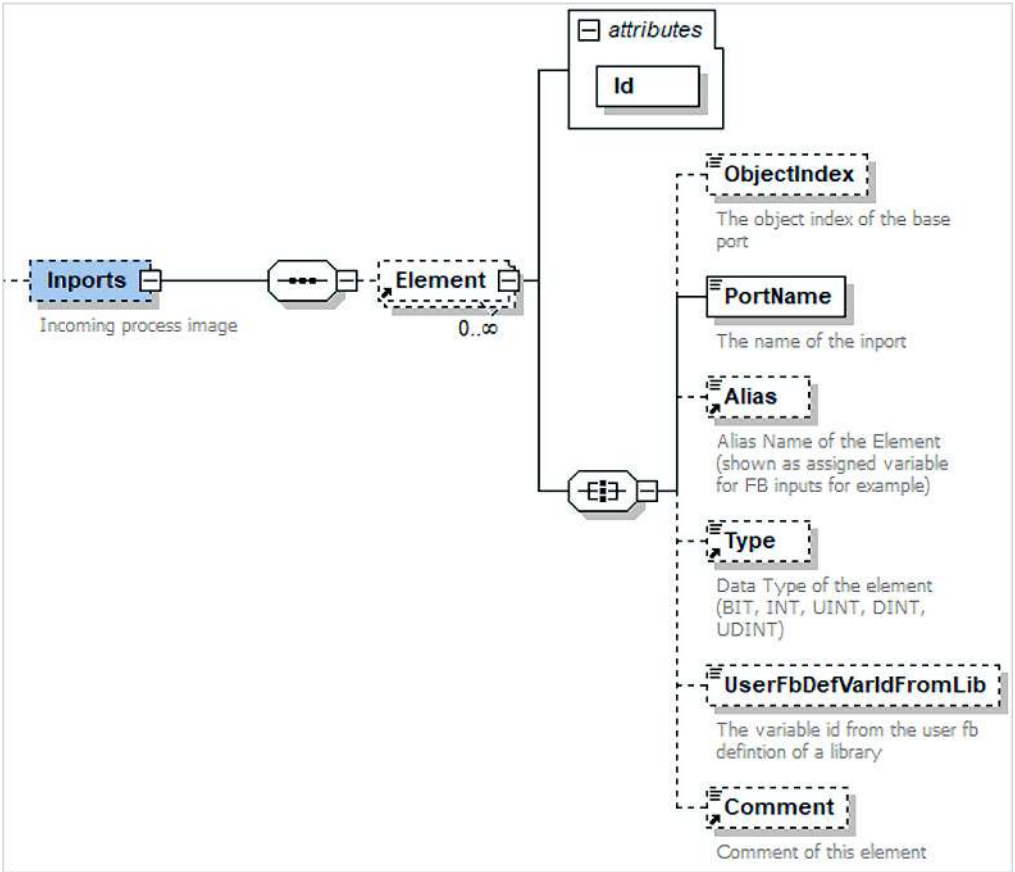


Fig. 23: UserFunctionBlock Inports

Element	Element
minOcc	0
maxOcc	unlimited
Data type	/
Contents	sophisticated
Description	Element in the input process image

Attribute	Id
Data type	xs:ID
Use	required
Description	ID of an element See chapter ID rules [▶ 75].

Elements	ObjectIndex	PortName	Alias
minOcc	0	1	0
maxOcc	1	1	1
Data type	xs:int	/	xs:string
Contents	simple	simple	simple
Description	Object index of the base port	Name of the inport	Alias name of the element (shown as assigned variable for function block inputs, for example)

Elements	Type	UserFbDefVarIdFromLib	Comment
minOcc	0	0	0
maxOcc	1	1	1
Data type	xs:string	xs:string	xs:string
Contents	simple	simple	simple
Restriction	BIT, BYTE, USINT UINT, INT, UDINT, DINT	/	/
Description	Data type of the element	Variable ID of a user function block definition from the library	Comment of this element

As XML code, this looks as follows, for example:

```
<Inports>
  <Element Id="g1_n1_u2_i1">
    <UserFbDefVarIdFromLib>2110a823-9d12-4d01-8b3e-8877bbf961ff</UserFbDefVarIdFromLib>
    <PortName>FBTon1_TonIn</PortName>
    <Type>BIT</Type>
  </Element>
  <Element Id="g1_n1_u2_i2">
    <UserFbDefVarIdFromLib>4ae048c3-839a-435f-b80d-5583e891b1e4</UserFbDefVarIdFromLib>
    <PortName>FBTon2_TonIn</PortName>
    <Type>BIT</Type>
  </Element>
</Inports>
```

7.1.2.4.1.1.2.3 Outports

The output process image "Outports" corresponds to the input process image "Inports" in terms of its attributes and elements. For more information, see the chapter Inports [▶ 47].

This can look like the following XML code:

```
<Outports>
  <Element Id="g1_n1_u2_o1">
    <ObjectIndex>1</ObjectIndex>
    <UserFbDefVarIdFromLib>8a6f4e1f-eca4-4f83-8bc9-f263dd3e7856</UserFbDefVarIdFromLib>
    <PortName>FBTon1_TonOut</PortName>
    <Alias>out1</Alias>
```

```
    <Type>BIT</Type>
  </Element>
  <Element Id="g1_n1_u2_o2">
    <ObjectIndex>1</ObjectIndex>
    <UserFbDefVarIdFromLib>5d5de48d-8a8d-4cd3-8bd6-6b262d808729</UserFbDefVarIdFromLib>
    <PortName>FBAnd1_AndOut</PortName>
    <Alias>out2</Alias>
    <Type>BIT</Type>
  </Element>
</Outports>
```

7.1.2.4.1.2 Comments

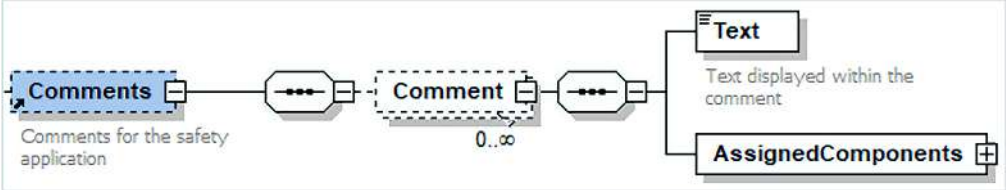


Fig. 24: UserFB's comments

Element	Comment
minOcc	0
maxOcc	unlimited
Contents	sophisticated
Description	Comment

Elements	Text	AssignedComponents
minOcc	1	1
maxOcc	1	1
Data type	xs:string	/
Contents	simple	sophisticated
Description	Text displayed within the comment	Assigned components See chapter AssignedComponents [▶ 50].

As XML code, this looks as follows, for example:

```
<Comments>
  <Comment>
    <Text>TestComment1</Text>
    <AssignedComponents>
    <Component IdRef="u1_n1_f1" />
    </AssignedComponents>
  </Comment>
  <Comment>
    <Text>TestComment2</Text>
    <AssignedComponents>
    <Component IdRef="u1" />
    </AssignedComponents>
  </Comment>
</Comments>
```

7.1.2.4.1.2.1 AssignedComponents

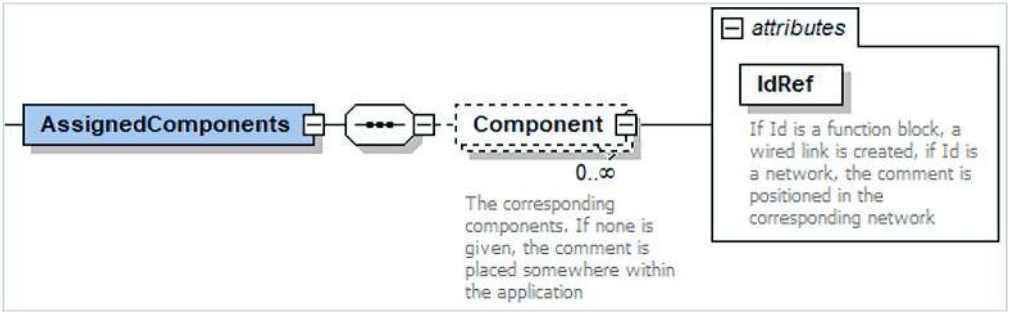


Fig. 25: UserFB's comments

For a valid XML file, the order of the elements must be strictly adhered to.

Element	Component
minOcc	0
maxOcc	unlimited
Contents	sophisticated
Description	Corresponding component

Attribute	IdRef
Data type	xs:IDREF
Use	required
Description	If the ID is a function block, a wired link is created. See FunctionBlock.Id attribute or UserFunctionBlock.Id attribute in chapter FunctionBlock [▶ 35] or UserFunctionBlock [▶ 43]. If the ID is a network, the comment is positioned in the corresponding network. See Network.Id attribute in chapter Network [▶ 34].

As XML code, this looks as follows, for example:

```
<AssignedComponents>
  <Component IdRef="u1_n1_f1"/>
</AssignedComponents>
```

7.1.2.4.2 TwinSAFEGroups

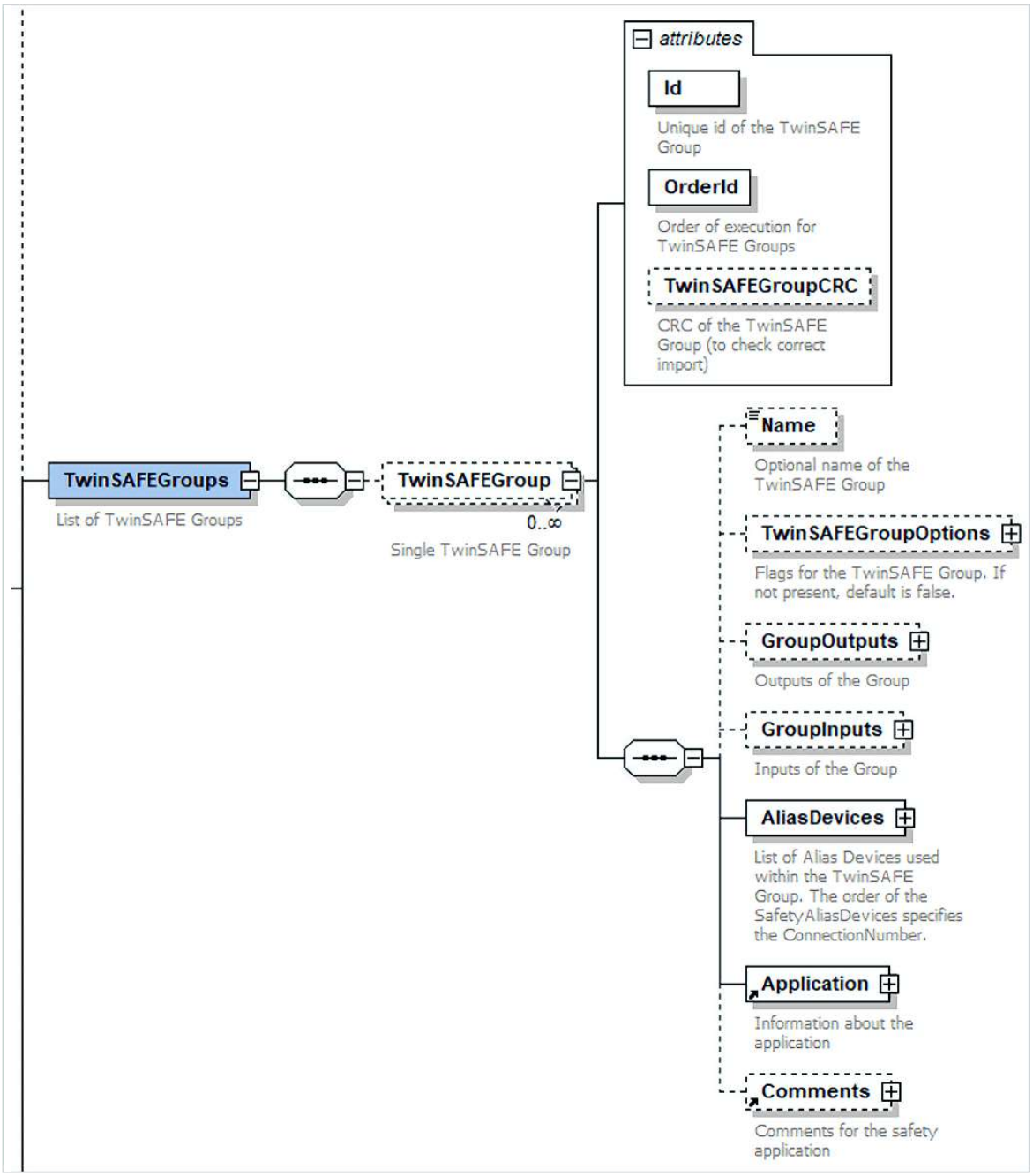


Fig. 26: TwinSAFEGroups

For a valid XML file, the order of the elements must be strictly adhered to.

Element	TwinSAFEGroup
minOcc	0
maxOcc	unlimited
Contents	sophisticated
Description	Single TwinSAFE group

Attributes	Id	OrderId	TwinSAFEGroupCRC
Data type	xs:ID	xs:int	xs:int
Use	required	required	optional
Description	Unique ID of the TwinSAFE group	Order of execution for the TwinSAFE group	CRC of the TwinSAFE group Currently not used.

Attributes	Id	OrderId	TwinSAFEGroupCRC
	See chapter ID rules [▶ 75].		

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Name	TwinSAFEGroupOptions	GroupOutputs	GroupInputs
minOcc	0	0	0	0
maxOcc	1	1	1	1
Data type	xs:string	TwinSAFEGroupOptions	GroupPorts	GroupPorts
Contents	simple	sophisticated	sophisticated	sophisticated
Description	Name of the TwinSAFE group	Identification of the TwinSAFE group See chapter TwinSAFEGroupOptions [▶ 53].	Outputs of the TwinSAFE group See chapter GroupOutputs [▶ 54].	Inputs of the TwinSAFE group See chapter GroupInputs [▶ 55].

Elements	AliasDevices	Application	Comments
minOcc	1	1	0
maxOcc	1	1	1
Contents	sophisticated	sophisticated	sophisticated
Description	List of alias devices used in the TwinSAFE group See chapter AliasDevices [▶ 56].	Information about the application See chapter Application [▶ 65].	Comments for the safety application See chapter Comments [▶ 65].

As XML code, this looks as follows, for example:

```
<TwinSAFEGroups>
  <TwinSAFEGroup Id="g1" OrderId="0">
    <Name>TwinSafeGroup1</Name>
    <TwinSAFEGroupOptions>
    </TwinSAFEGroupOptions>
    <GroupOutputs>
    </GroupOutputs>
    <GroupInputs>
    </GroupInputs>
    <AliasDevices>
    </AliasDevices>
    <Application>
    </Application>
  </TwinSAFEGroup>
  <TwinSAFEGroup Id="g2" OrderId="1">
    <Name>TwinSafeGroup2</Name>
    <TwinSAFEGroupOptions>
    </TwinSAFEGroupOptions>
    <GroupOutputs>
    </GroupOutputs>
    <GroupInputs>
    </GroupInputs>
    <AliasDevices>
    </AliasDevices>
    <Application>
    </Application>
  </TwinSAFEGroup>
</TwinSAFEGroups>
```

7.1.2.4.2.1 TwinSAFEGroupOptions

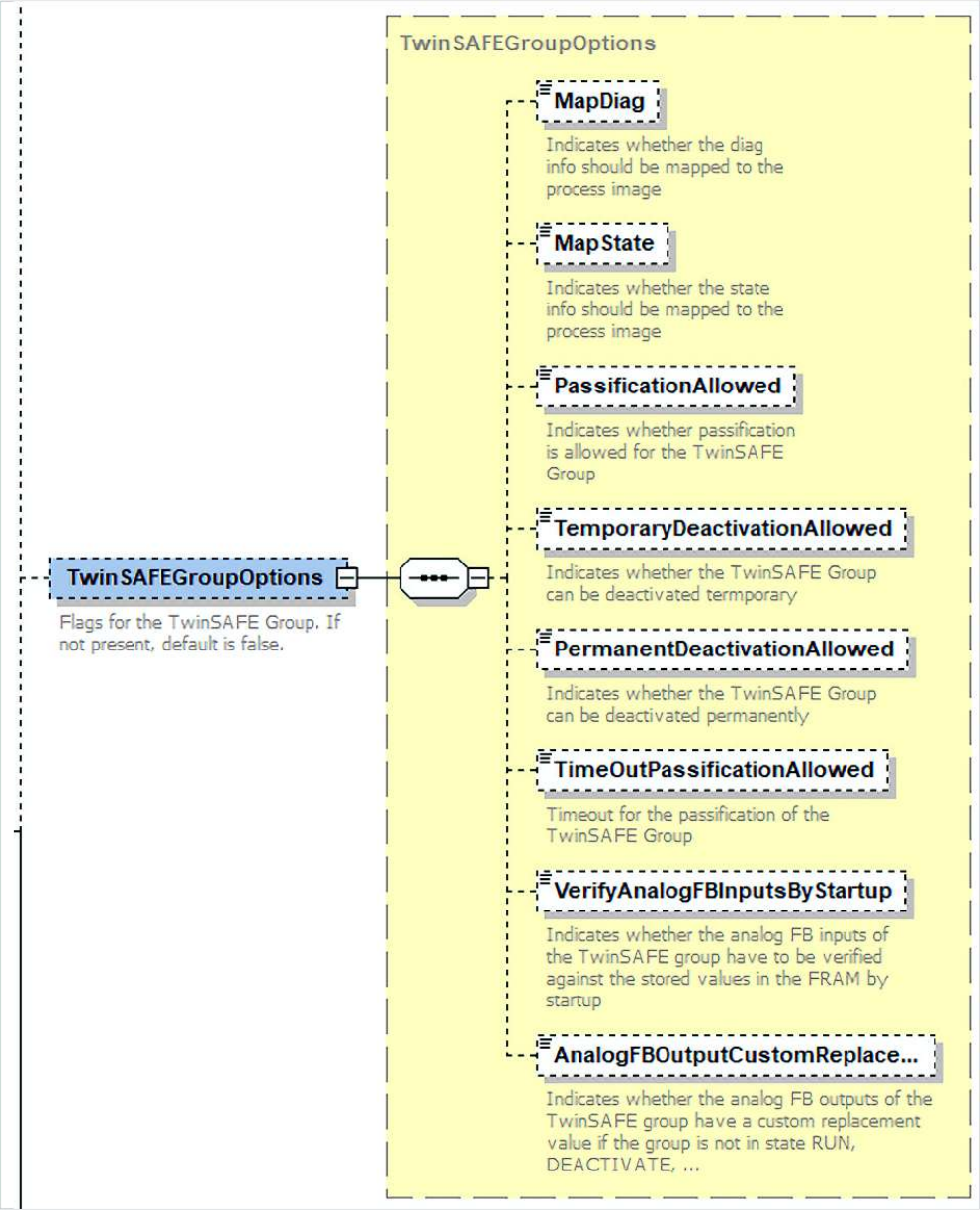


Fig. 27: TwinSAFEGroup Options

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	MapDiag	MapState	PassificationAllowed	TemporaryDeactivationAllowed
minOcc	0	0	0	0
maxOcc	1	1	1	1
Data type	xs:boolean	xs:boolean	xs:boolean	xs:boolean
Contents	simple	simple	simple	simple
Description	Indicates whether the diag information should be mapped to the process image	Indicates whether the state information should be mapped to the process image	Indicates whether passivation is allowed for the TwinSAFE group	Indicates whether the TwinSAFE group can be deactivated temporary

Elements	PermanentDeactivationAllowed	TimeOutPassificationAllowed	VerifyAnalogFBInputsByStartup	AnalogFBOutputCustomReplacementValues
minOcc	0	0	0	0

Elements	PermanentDeactiva-tionAllowed	TimeOutPassifica-tionAllowed	VerifyAnalogFBIn-putsByStartup	AnalogFBOutput-CustomReplace-mentValues
maxOcc	1	1	1	1
Data type	xs:boolean	xs:unsignedShort	xs:boolean	xs:boolean
Contents	simple	simple	simple	simple
Description	Indicates whether the TwinSAFE group can be deactivated permanently	Timeout for passivation of the TwinSAFE group	Indicates whether the analog function block inputs of the TwinSAFE group must be compared with the values stored in the permanent memory at startup	Indicates whether the analog function block outputs of the Twinsafe group have a user-defined substitute value if the group is not in the RUN, DEACTIVATE ... state

As XML code, this looks as follows, for example:

```
<TwinSAFEGroupOptions>
  <MapDiag>false</MapDiag>
  <MapState>false</MapState>
  <PassificationAllowed>false</PassificationAllowed>
  <TemporaryDeactivationAllowed>false</TemporaryDeactivationAllowed>
  <PermanentDeactivationAllowed>false</PermanentDeactivationAllowed>
  <TimeOutPassificationAllowed>10000</TimeOutPassificationAllowed>
  <VerifyAnalogFBInputsByStartup>false</VerifyAnalogFBInputsByStartup>
  <AnalogFBOutputCustomReplacementValues>false</AnalogFBOutputCustomReplacementValues>
</TwinSAFEGroupOptions>
```

7.1.2.4.2.2 GroupOutputs

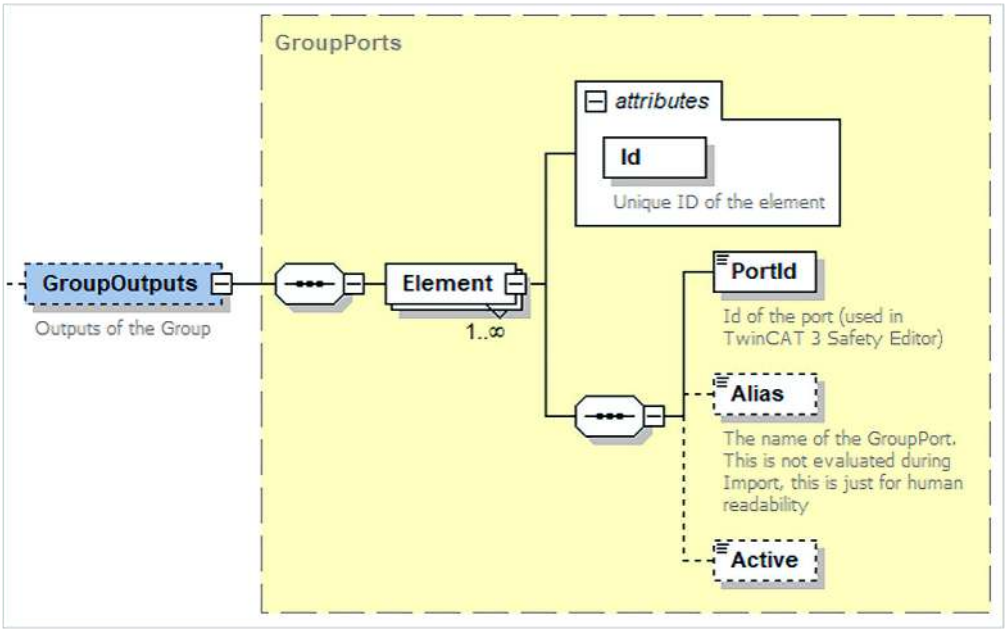


Fig. 28: GroupOutputs

For a valid XML file, the order of the elements must be strictly adhered to.

Element	Element
minOcc	1
maxOcc	unlimited
Contents	sophisticated
Description	Element of a group output

Attributes	Id
Data type	xs:ID
Use	required
Description	Unique ID of the element See chapter ID rules [▶ 75].

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	PortId	Alias	Active
minOcc	1	0	0
maxOcc	1	1	1
Data type	HexDecUInt32Value	xs:string	xs:boolean
Contents	simple	simple	simple
Description	ID of the port (used in the TwinCAT 3 Safety Editor) See chapter PortIds [▶ 78].	Name of the group port See chapter PortIds [▶ 78]. This is not evaluated during import, but is only used for readability.	Indicates whether the port is in use This is not evaluated during import, but is only used for readability.

As XML code, this looks as follows, for example:

```
<GroupOutputs>
  <Element Id="g1_o1">
    <PortId>33619968</PortId>
    <Alias>FbErr</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_o2">
    <PortId>33685504</PortId>
    <Alias>ComErr</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_o3">
    <PortId>33751040</PortId>
    <Alias>OutErr</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_o4">
    <PortId>33816576</PortId>
    <Alias>OtherErr</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_o5">
    <PortId>33882112</PortId>
    <Alias>ComStartup</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_o6">
    <PortId>33947648</PortId>
    <Alias>FbDeactive</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_o7">
    <PortId>34013184</PortId>
    <Alias>FbRun</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_o8">
    <PortId>34078720</PortId>
    <Alias>InRun</Alias>
    <Active>true</Active>
  </Element>
</GroupOutputs>
```

7.1.2.4.2.3 GroupInputs

The attributes and elements of the "GroupInputs" element correspond to those of the "GroupOutputs" element. For more information, see the chapter GroupOutputs [▶ 54].

As XML code, this looks as follows, for example:

```
<GroupInputs>
  <Element Id="g1_i1">
    <PortId>16842752</PortId>
    <Alias>RunStop</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_i2">
    <PortId>16908288</PortId>
    <Alias>ErrAck</Alias>
    <Active>true</Active>
  </Element>
  <Element Id="g1_i3">
    <PortId>16973824</PortId>
    <Alias>ModuleFault</Alias>
    <Active>true</Active>
  </Element>
</GroupInputs>
```

7.1.2.4.2.4 AliasDevices

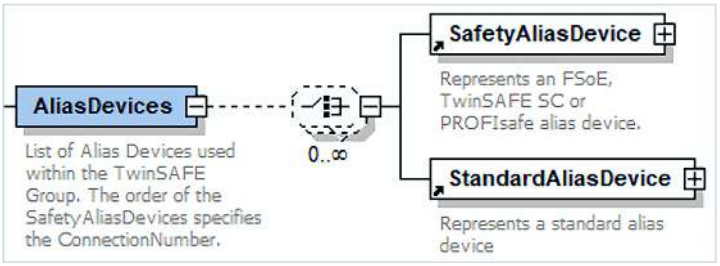


Fig. 29: AliasDevices

Elements	SafetyAliasDevices	StandardAliasDevices
minOcc	1	1
maxOcc	1	1
Contents	sophisticated	sophisticated
Description	Represents an FSoE, TwinSAFE SC or PROFIsafe alias device See chapter SafetyAliasDevice [▶ 57].	Represents a standard alias device See chapter StandardAliasDevice [▶ 63].

As XML code, this looks as follows, for example:

```
<AliasDevices>
  <StandardAliasDevice Id="g1_a1" OrderId="1">
  </StandardAliasDevice>
  <SafetyAliasDevice Id="g1_a2">
   </SafetyAliasDevice>
  <SafetyAliasDevice Id="g1_a3" IsExternal="true">
  </SafetyAliasDevice>
  <StandardAliasDevice Id="g1_a4" OrderId="2">
  </StandardAliasDevice>
  <StandardAliasDevice Id="g1_a5" OrderId="3">
  </StandardAliasDevice>
  <SafetyAliasDevice Id="g1_a6" IsExternal="false">
  </SafetyAliasDevice>
</AliasDevices>
```

7.1.2.4.2.4.1 SafetyAliasDevice

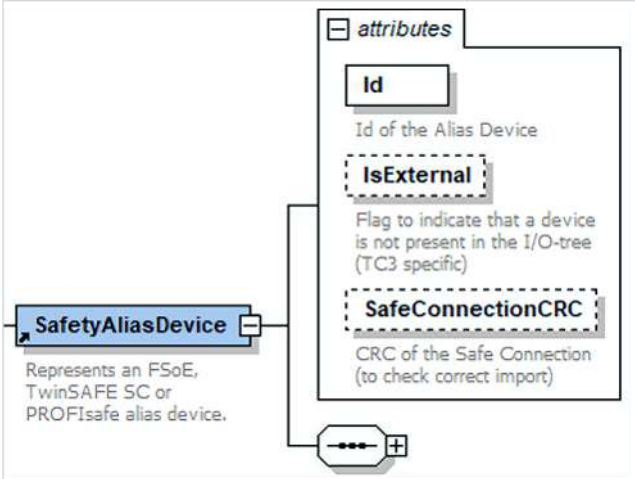


Fig. 30: SafetyAliasDevices attributes

Attributes	Id	IsExternal	SafeConnectionCRC
Data type	xs:ID	xs:boolean	xs:int
Use	required	optional	optional
Description	ID of the alias device See chapter ID rules [▶ 75].	Flag to indicate that a device is not present in the I/O tree (TwinCAT 3 specific)	CRC of the safe connection Currently not used.

For a valid XML file, the order of the elements must be strictly adhered to.

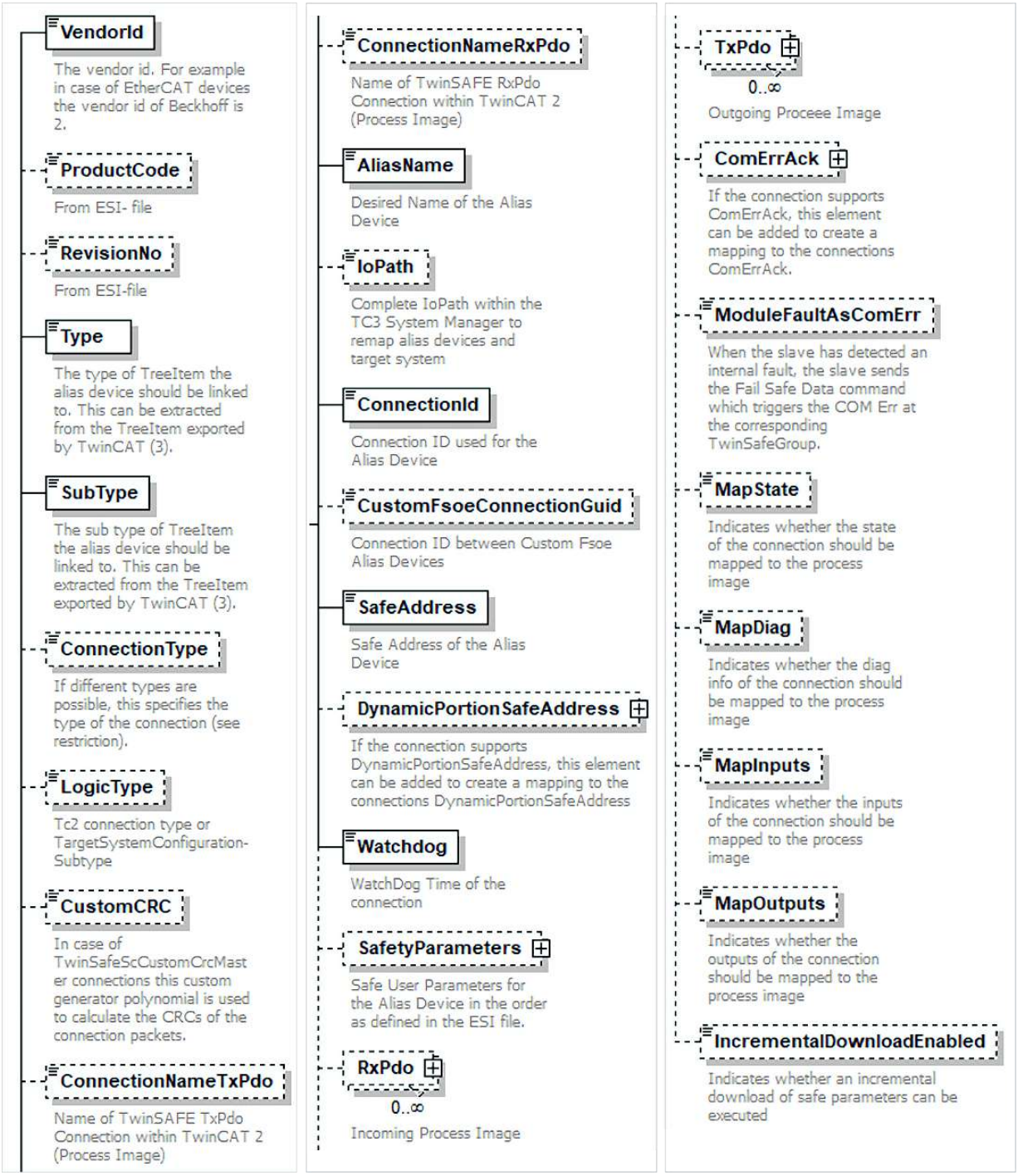


Fig. 31: SafetyAliasDevices elements

Elements	VendorId	ProductCode	RevisionNo	Type
minOcc	1	0	0	1
maxOcc	1	1	1	1
Data type	xs:int	xs:int	xs:int	xs:int
Contents	simple	simple	simple	simple
Description	Vendor ID	Product code Is not evaluated in the TwinSAFE editor	Revision number Is not evaluated in the TwinSAFE editor	Device type

Elements	SubType	ConnectionType	LogicType
minOcc	1	0	0
maxOcc	1	1	1

Elements	SubType	ConnectionType	LogicType
Data type	xs:int	xs:string	xs:string
Contents	simple	simple	simple
Restriction	/	FSoEMaster, FSoESlave, TwinSafeScCrc1master, TwinSafeScCrc2master, TwinSafeScCrc3master, TwinSafeScCrc4master, TwinSafeScCrc5master, TwinSafeScCrc6master, TwinSafeScCrc7master, TwinSafeScCrc8master, TwinSafeScCustomCrcMaster, PROFIsafeMaster, PROFIsafeSlave, Local	Possible settings: • None • KL6904 • EL6900EL6930 • EL6910EJ6910 • All target systems named in TargetSystemConfiguration /SubType except EL6900, EL6930, EL6910 and EJ6910
Description	Subtype of the device	Specific connection type if different connection types are available See chapter ConnectionType [▶ 88].	TwinCAT 3 connection type or subtype of the TargetSystemConfiguration See chapter LogicType [▶ 89].

Elements	CustomCRC	ConnectionNameTx-Pdo	ConnectionNameRx-Pdo	AliasName
minOcc	0	0	0	1
maxOcc	1	1	1	1
Data type	xs:unsingedInt	xs:string	xs:string	xs:string
Contents	simple	simple	simple	simple
Description	In the case of TwinSafeScCustomCrcMaster connections, this custom generator polynomial is used to calculate the CRCs of the connection packets	Name of the TwinSAFE TxPdo connection in TwinCAT 2 (process image) Currently not used.	Name of the TwinSAFE RxPdo connection in TwinCAT 2 (process image) Currently not used.	Desired name of the alias device

Elements	IoPath	ConnectionId	CustomFsoeConnectionGuid	SafeAddress
minOcc	0	1	0	1
maxOcc	1	1	1	1
Data type	xs:string	xs:positiveInteger	xs:string	xs:unsignedShort
Contents	simple	simple	simple	simple
Description	Complete I/O path within the TwinCAT 3 System Manager to remap alias devices and target systems	Connection ID used for the alias device	Connection ID between user-defined FSoE alias devices	Safe address of the alias device

Elements	DynamicPortion-SafeAddress	Watchdog	SafetyParameters	RxPdo
minOcc	0	1	0	0
maxOcc	1	1	1	unlimited
Data type	/	xs:unsignedShort	/	ProcessImage
Contents	sophisticated	simple	sophisticated	sophisticated

Elements	DynamicPortion-SafeAddress	Watchdog	SafetyParameters	RxPdo
Description	If a SafetyAliasDevice supports DynamicPortionSafeAddress, a mapping can be created with this element See chapter DynamicPortionSafeAddress [▶ 61].	Watchdog time of the connection	Safe user parameters for the alias devices See chapter SafetyParameters [▶ 61].	Input process image See chapter RxPdo [▶ 62].

Elements	TxPdo	ComErrAck	ModuleFaultAsCom-Err	MapState
minOcc	0	0	0	0
maxOcc	unlimited	1	1	1
Data type	ProcessImage	/	xs:boolean	xs:boolean
Contents	sophisticated	sophisticated	simple	simple
Description	Output process image See chapter TxPdo [▶ 62].	If the connection supports ComErrAck, this element can be added to create a mapping to the ComErrAck connections See chapter ComErrAck [▶ 62].	If the slave has detected an internal error, the slave sends the fail-safe data command, which triggers the COM Err at the corresponding TwinSAFE group	Indicates whether the state information of the connection should be mapped to the process image

Elements	MapDiag	MapInputs	MapOutputs	IncrementalDown-loadEnabled
minOcc	0	0	0	0
maxOcc	1	1	1	1
Data type	xs:boolean	xs:boolean	xs:boolean	xs:boolean
Contents	simple	simple	simple	simple
Description	Indicates whether the diag information of the connection should be mapped to the process image	Indicates whether the connection inputs are to be mapped to the process image Selection of SafetyAliasDevices via TxPdo and RxPdo	Indicates whether the connection outputs are to be mapped to the process image	Indicates whether an incremental download of the safety parameters can be executed

As XML code, this looks as follows, for example:

```
<SafetyAliasDevice Id="g1_a6" IsExternal="false">
  <VendorId>2</VendorId>
  <Type>60</Type>
  <SubType>17827710</SubType>
  <ConnectionType>FSoEMaster</ConnectionType>
  <LogicType>None</LogicType>
  <AliasName>EL1918, 8 digital inputs (FW 2)_1</AliasName>
  <ConnectionId>9</ConnectionId>
  <SafeAddress>23</SafeAddress>
  <DynamicPortionSafeAddress Id="g1_a6_d" />
  <Watchdog>100</Watchdog>
  <SafetyParameters>
  </SafetyParameters>
  <RxPdo>
  </RxPdo>
  <TxPdo>
  </TxPdo>
  <ComErrAck Id="g1_a6_c" />
  <ModuleFaultAsComErr>false</ModuleFaultAsComErr>
```

```
  <MapState>false</MapState>
  <MapDiag>false</MapDiag>
  <MapInputs>false</MapInputs>
  <MapOutputs>false</MapOutputs>
  <IncrementalDownloadEnabled>false</IncrementalDownloadEnabled>
</SafetyAliasDevice>
```

7.1.2.4.2.4.1.1 DynamicPortionSafeAddress

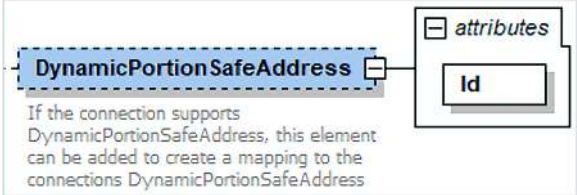


Fig. 32: DynamicPortSafeAddress

Attributes	Id
Data type	xs:ID
Use	required
Description	ID of the safe address See chapter ID rules [▶ 75].

As XML code, this looks as follows, for example:

```
<DynamicPortionSafeAddress Id="g1_a6_d" />
```

7.1.2.4.2.4.1.2 SafetyParameters

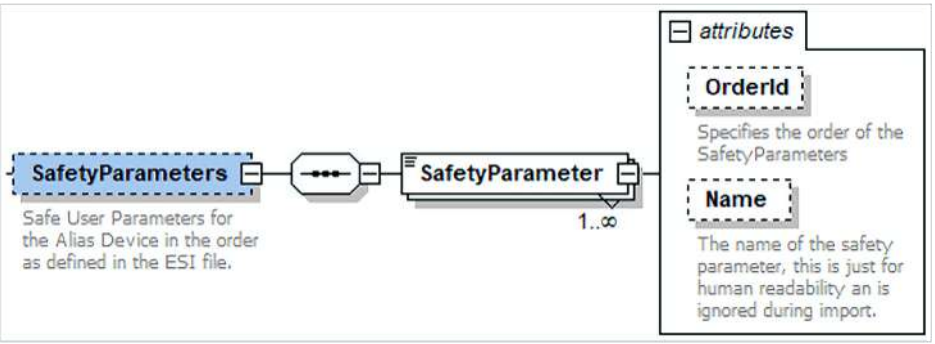


Fig. 33: SafetyParameters

For a valid XML file, the order of the elements must be strictly adhered to.

Element	SafetyParameter
minOcc	1
maxOcc	unlimited
Data type	xs:string
Contents	sophisticated
Description	Safety parameter value

Attributes	OrderId	Name
Data type	xs:int	xs:string
Use	optional	optional
Description	Specifies the order of the safety parameters	Name of the safety parameter This is not evaluated during import, but is only used for readability

As XML code, this looks as follows, for example:

```
<SafetyParameters>
  <SafetyParameter OrderId="1" Name="1018:01 Identity:Vendor ID">2</SafetyParameter>
  <SafetyParameter OrderId="2" Name="F050:01 Detected modules:SubIndex 001">17827710</
SafetyParameter>
  <SafetyParameter OrderId="3" Name="8000:01 FSIN Module 1 Settings Common:ModuloDiagTestPulse">0</
SafetyParameter>
  <SafetyParameter OrderId="4" Name="8000:02 FSIN Module 1 Settings
Common:MultiplierDiagTestPulse">1</SafetyParameter>
  <SafetyParameter OrderId="5" Name="8000:04 FSIN Module 1 Settings Common:Diag TestPulse
active">1</SafetyParameter>
  <SafetyParameter OrderId="6" Name="8000:05 FSIN Module 1 Settings Common:Module Fault Link
active">1</SafetyParameter>
  <SafetyParameter OrderId="7" Name="8001:01 FSIN Module 1 Settings Channel:InputFilterTime">10</
SafetyParameter>
  <SafetyParameter OrderId="8" Name="8001:02 FSIN Module 1 Settings
Channel:DiagTestPulseFilterTime">2</SafetyParameter>
  ...
</SafetyParameters>
```

7.1.2.4.2.4.1.3 RxPdo

The input process image "RxPdo" of the safety alias devices corresponds in its attributes and elements to the output process image "TxPdo" of the target system. For more information, see the chapter TxPdo [▶ 29].

7.1.2.4.2.4.1.4 TxPdo

The output process image "TxPdo" of the safety alias devices corresponds in its attributes and elements to the output process image "TxPdo" of the target system. For more information, see the chapter TxPdo [▶ 29].

7.1.2.4.2.4.1.5 ComErrAck

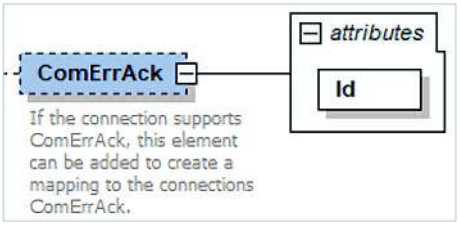


Fig. 34: ComErrAck

Attributes	Id
Data type	xs:ID
Use	required
Description	ID See chapter ID rules [▶ 75].

As XML code, this looks as follows, for example:

```
<ComErrAck Id="g1_a6_c" />
```

7.1.2.4.2.4.2 StandardAliasDevice

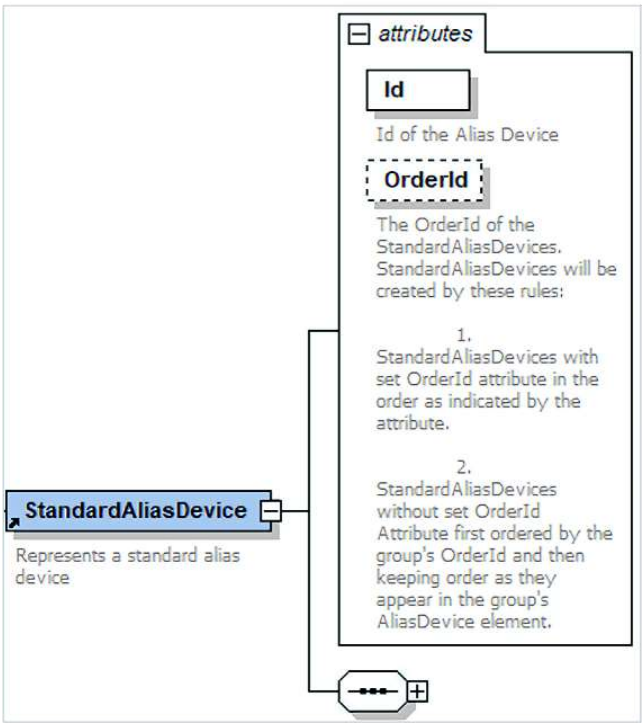


Fig. 35: StandardAliasDevice Attributes

Attributes	Id	OrderId
Data type	xs:ID	xs:unsignedInt
Use	required	optional
Description	ID of the alias device See chapter ID rules [▶ 75].	Order ID of the standard alias device Standard alias devices are created according to the following rules: 1. Standard alias devices with set OrderId attribute in the order specified by the attribute 2. Standard alias devices without a set OrderId attribute first in the order of the OrderId of the group and then in the order in which they appear in the AliasDevice element of the group

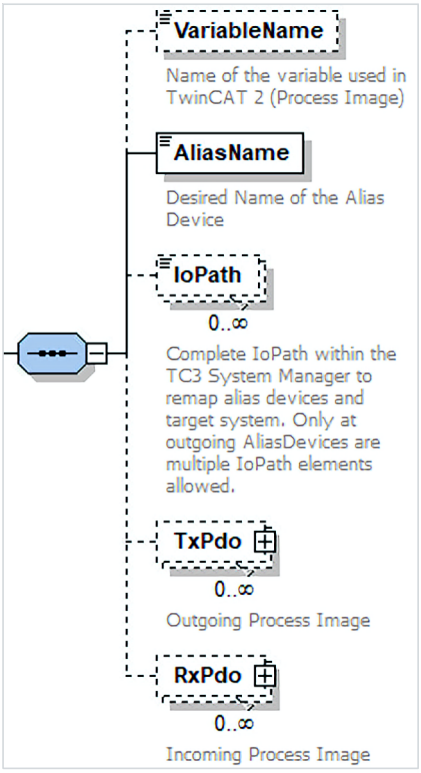


Fig. 36: StandardAliasDevice elements

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	VariableName	AliasName	IoPath	TxPdo	RxPdo
minOcc	0	1	0	0	0
maxOcc	1	1	unlimited	unlimited	unlimited
Data type	xs:string	xs:string	xs:string	ProcessImage	ProcessImage
Contents	simple	simple	simple	sophisticated	sophisticated
Description	Variable name as used in TwinCAT 2 (process image) Currently not used.	Desired name of the alias device	Complete I/O path within the TwinCAT 3 System Manager to remap alias devices and target systems	Output process image See chapter TxPdo [▶ 64].	Input process image See chapter RxPdo [▶ 65].

As XML code, this looks as follows, for example:

```
<StandardAliasDevice Id="g1_a1" OrderId="1">
  <AliasName>ErrorAcknowledgement</AliasName>
  <IoPath>TIRT^Task 2^Outputs^ErrAck</IoPath>
  <RxPdo BitSize="1">
    <Element Id="g1_a1_i1">
      <Name>In</Name>
      <Type>BIT</Type>
      <BitOffset>0</BitOffset>
      <BitSize>1</BitSize>
      <IsSafeTimer>false</IsSafeTimer>
    </Element>
  </RxPdo>
</StandardAliasDevice>
```

7.1.2.4.2.4.2.1 TxPdo

The output process image "TxPdo" of the standard alias devices corresponds in its attributes and elements to the output process image "TxPdo" of the target system. For more information, see the chapter TxPdo [▶ 29].

7.1.2.4.2.4.2.2 RxPdo

The input process image "RxPdo" of the standard alias devices corresponds in its attributes and elements to the output process image "TxPdo" of the target system. For more information, see the chapter TxPdo [▶ 29].

7.1.2.4.2.5 Application

The application of the TwinSAFE groups corresponds in its attributes and elements to the application of the UserFBs. For more information, see the chapter Application [▶ 33].

7.1.2.4.2.6 Comments

The comments on the TwinSAFE groups correspond in their attributes and elements to the comments on the UserFBs. For more information, see the chapter Comments [▶ 49].

7.1.2.4.3 UserFbPortRelations

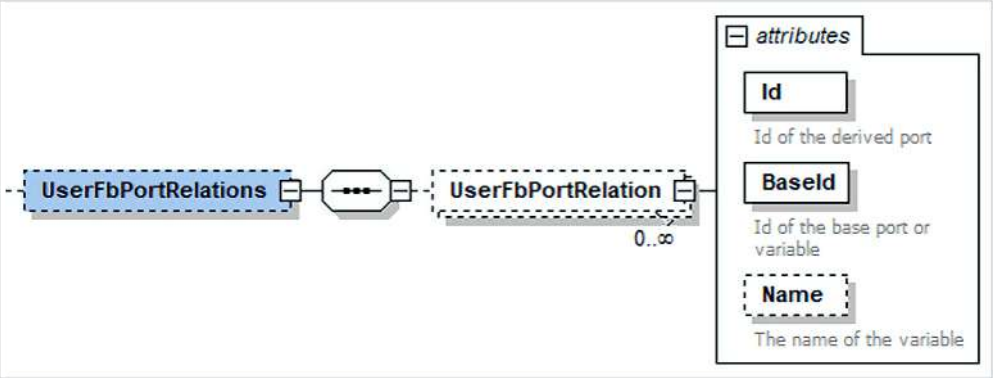


Fig. 37: UserFBPortRelations

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	UserFbPortRelation
minOcc	0
maxOcc	unlimited
Contents	sophisticated
Description	Relation between the user function block ports

Attributes	Id	BaseId	Name
Data type	xs:string	xs:string	xs:string
Use	required	required	optional
Description	Id of the derived port See chapter ID rules [▶ 75].	ID of the base port or variable	Name of the variable

As XML code, this looks as follows, for example:

```
<UserFbPortRelations>
  <UserFbPortRelation Id="g1_n1_u2_p1" BaseId="r1_u1_n1_f1_p1" />
  <UserFbPortRelation Id="g1_n1_u2_p2" BaseId="r1_u1_n1_f3_p1" />
  <UserFbPortRelation Id="g1_n1_u2_i1" BaseId="r1_u1_n1_f1_i1" Name="FBTon1_TonIn" />
  <UserFbPortRelation Id="g1_n1_u2_i2" BaseId="r1_u1_n1_f3_i1" Name="FBTon2_TonIn" />
  <UserFbPortRelation Id="g1_n1_u2_o1" BaseId="r1_u1_n1_f1_o1" Name="FBTon1_TonOut" />
  <UserFbPortRelation Id="g1_n1_u2_o2" BaseId="r1_u1_n1_f2_o1" Name="FBAnd1_AndOut" />
  <UserFbPortRelation Id="g1_n2_u2_p1" BaseId="u1_n1_f1_p1" />
  <UserFbPortRelation Id="g1_n2_u2_p2" BaseId="u1_n1_f1_p2" />
  <UserFbPortRelation Id="g1_n2_u2_p3" BaseId="u1_n1_f1_p3" />
  <UserFbPortRelation Id="g1_n2_u2_i1" BaseId="u1_n1_f1_i1" Name="scaleIn" />
  <UserFbPortRelation Id="g1_n2_u2_o1" BaseId="u1_n1_f1_o3" Name="scaleOut" />
</UserFbPortRelations>
```

7.1.2.4.4 Mappings

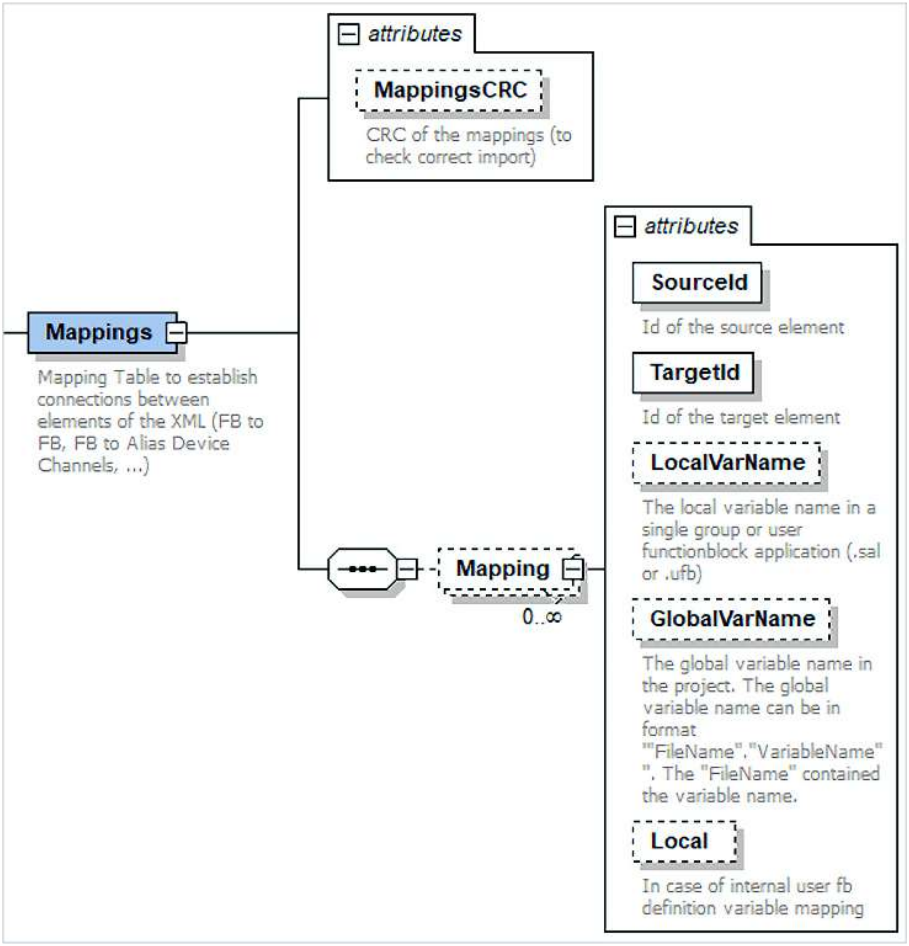


Fig. 38: Mappings

Attribute	MappingsCRC
Data type	xs:int
Use	optional
Description	CRC of the mapping Currently not used.

For a valid XML file, the order of the elements must be strictly adhered to.

Element	Mapping
minOcc	0
maxOcc	unlimited
Contents	sophisticated
Description	Linking

Attributes	SourceId	TargetId	LocalVarName	GlobalVarName	Local
Data type	xs:IDREF	xs:IDREF	xs:string	xs:string	xs:boolean
Use	required	required	optional	optional	optional
Description	ID of the source element	ID of the target element	Name of the variable within a group or a user function block definition (*.sal or *.ufb)	Name of the global variable in a project Possible formats: • FileName • VariableName	Internal link, for internal use of the user function blocks

As XML code, this looks as follows, for example:

```
<Mappings>
  <Mapping TargetId="g1_i2" SourceId="g1_a1_i1" GlobalVarName="GVL1.ErrAck"/>
  <Mapping TargetId="g2_i2" SourceId="g1_a1_i1" GlobalVarName="GVL1.ErrAck"/>
  <Mapping TargetId="g1_i1" SourceId="g1_a4_i1" GlobalVarName="GVL1.Run"/>
  <Mapping TargetId="g2_i1" SourceId="g1_a4_i1" GlobalVarName="GVL1.Run"/>
  <Mapping TargetId="g2_n1_f1_i2" SourceId="g1_n2_f1_o1" GlobalVarName="GVL1.SafeOutF1"/>
  <Mapping TargetId="g1_n1_f1_i1" SourceId="g1_a3_i1" LocalVarName="in"/>
  <Mapping TargetId="g1_n2_u2_i1" SourceId="g1_a2_li20" LocalVarName="scaleIn"/>
  <Mapping TargetId="g1_a5_o1" SourceId="g1_n2_u2_o1" LocalVarName="scaleOut"/>
  <Mapping TargetId="g1_n2_f1_i1" SourceId="g1_n1_u2_o1" LocalVarName="out1"/>
  <Mapping TargetId="g1_a2_lo1" SourceId="g1_n1_u2_o2" LocalVarName="out2"/>
  <Mapping TargetId="g1_a2_lo2" SourceId="g1_n1_u2_o2" LocalVarName="out2"/>
  <Mapping TargetId="g1_a2_lo3" SourceId="g1_n1_u2_o2" LocalVarName="out2"/>
  <Mapping TargetId="g1_a2_lo4" SourceId="g1_n1_u2_o2" LocalVarName="out2"/>
  <Mapping TargetId="g1_a2_lo6" SourceId="g1_n1_u2_o2" LocalVarName="out2"/>
  <Mapping TargetId="g1_a3_o1" SourceId="g1_a6_i1" LocalVarName="Variable1"/>
  <Mapping TargetId="g1_n1_u2_i1" SourceId="g1_n1_f1_o1"/>
  <Mapping TargetId="g1_n1_u2_i2" SourceId="g1_n1_f1_o1"/>
</Mappings>
```

7.1.2.5 TargetSystemProcessImage

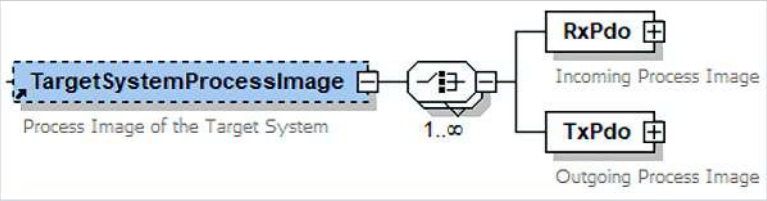


Fig. 39: TargetSystemProcessImage

Elements	RxPdo	TxPdo
minOcc	1	1
maxOcc	1	1
Contents	sophisticated	sophisticated
Description	Input process image See chapter RxPdo [▶ 68].	Output process image See chapter TxPdo [▶ 69].

As XML code, this looks as follows, for example:

```
<TargetSystemProcessImage>
  <TxPdo BitSize="48" OrderId="1" Name="FSoE Master Connection 1 Master Message">
  </TxPdo>
  <TxPdo BitSize="48" OrderId="2" Name="FSoE Master Connection 2 Master Message">
  </TxPdo>
  ...
  <RxPdo BitSize="48" OrderId="9" Name="FSoE Master Connection 1 Slave Message">
  </RxPdo>
  <RxPdo BitSize="56" OrderId="10" Name="FSoE Master Connection 2 Slave Message">
  </RxPdo>
  ...
</TargetSystemProcessImage>
```

7.1.2.5.1 RxPdo

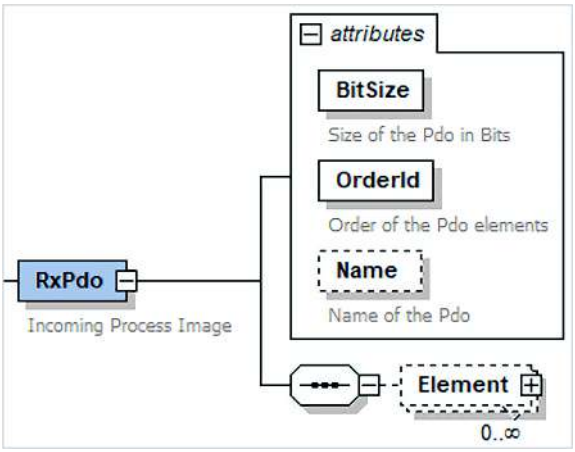


Fig. 40: RxPdo attributes

Attributes	BitSize	OrderId	Name
Data type	xs:int	xs:int	xs:string
Use	required	required	optional
Description	Size of the process image in bits	Order of the process image elements	Name of the process image

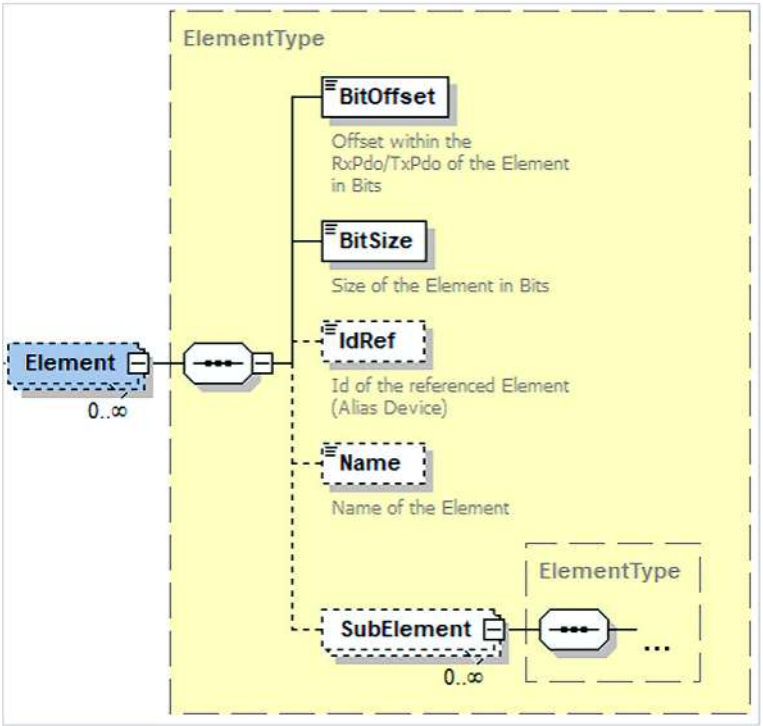


Fig. 41: RxPdo element

For a valid XML file, the order of the elements must be strictly adhered to.

Element	Element
minOcc	0
maxOcc	unlimited
Data type	ElementType
Contents	sophisticated
Description	Process image element

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	BitOffset	BitSize	IdRef	Name	SubElement
minOcc	1	1	0	0	0
maxOcc	1	1	1	1	unlimited
Data type	xs:int	xs:int	xs:string	xs:string	ElementType
Contents	simple	simple	simple	simple	sophisticated
Description	Element offset in the process image in bits	Element size in bits	ID of the referenced element (alias device)	Name of the element	Sub-element

As XML code, this looks as follows, for example:

```
<RxPdo BitSize="48" OrderId="9" Name="FSoE Master Connection 1 Slave Message">
  <Element>
    <BitOffset>0</BitOffset>
    <BitSize>48</BitSize>
    <IdRef>g1_a3</IdRef>
    <Name>Message_3</Name>
    <SubElement>
      <BitOffset>0</BitOffset>
      <BitSize>8</BitSize>
      <Name>FSoE CMD</Name>
    </SubElement>
    ...
  </Element>
</RxPdo>
```

7.1.2.5.2 TxPdo

The output process image "TxPdo" corresponds in its attributes and elements to the input process image "RxPdo". For more information, see the chapter RxPdo [▶ 68].

As XML code, this looks as follows, for example:

```
<TxPdo BitSize="48" OrderId="1" Name="FSoE Master Connection 1 Master Message">
  <Element>
    <BitOffset>0</BitOffset>
    <BitSize>48</BitSize>
    <IdRef>g1_a3</IdRef>
    <Name>Message_3</Name>
    <SubElement>
      <BitOffset>0</BitOffset>
      <BitSize>8</BitSize>
      <Name>FSoE CMD</Name>
    </SubElement>
    ...
  </Element>
</TxPdo>
```

7.1.3 Multi Settings

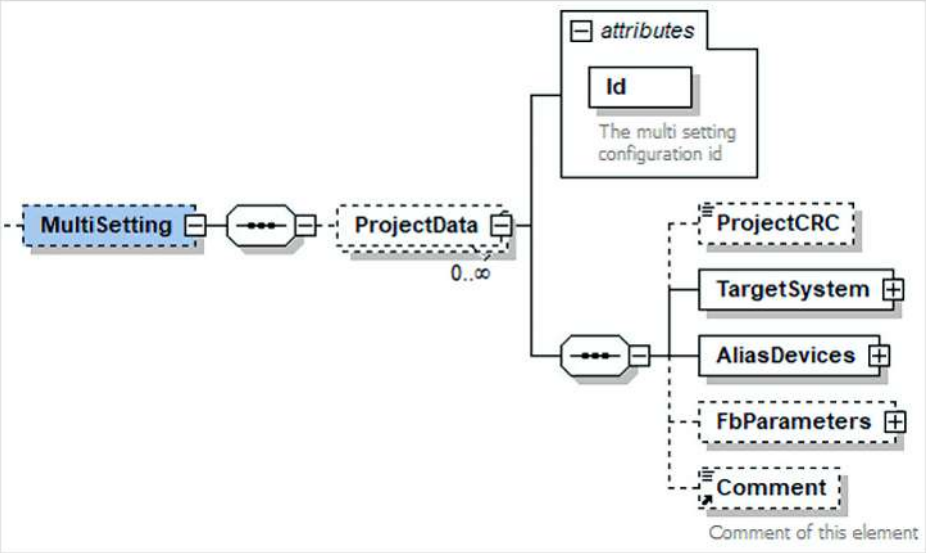


Fig. 42: Multisettings

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	ProjectData
minOcc	0
maxOcc	unlimited
Contents	sophisticated
Description	Project data

Attributes	Id
Data type	xs:ID
Use	required
Description	ID of the Multi Settings See chapter ID rules [▶ 75].

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	ProjectCRC	TargetSystem	AiasDevices	FbParameters	Comment
minOcc	0	1	1	0	0
maxOcc	1	1	1	1	1
Data type	HexDecUInt32Value	/	/	/	xs:string
Contents	simple	sophisticated	sophisticated	sophisticated	simple
Description	CRC of the project	Target System See chapter TargetSystem [▶ 71].	Alias device See chapter AliasDevices [▶ 72].	Function block parameters See chapter FbParameters [▶ 74].	Comment of this element

As XML code, this looks as follows, for example:

```
<MultiSetting>
  <ProjectData Id="ms_1">
    <ProjectCRC>0xCC9E</ProjectCRC>
    <TargetSystem BaseId="t1">
    </TargetSystem>
    <AliasDevices>
```

```
    </AliasDevices>
    <FbParameters>
    </FbParameters>
    <Comment>
    </Comment>
  </ProjectData>
</MultiSetting>
```

7.1.3.1 TargetSystem

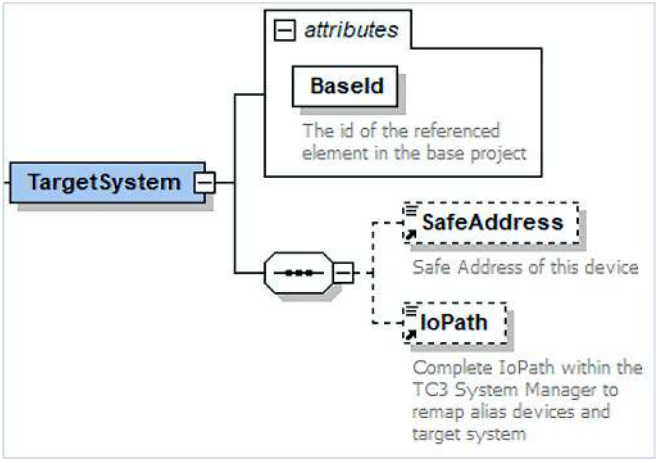


Fig. 43: TargetSystem

Attributes	BaseId
Data type	xs:IDREF
Use	required
Description	ID of the reference element in the base project See TargetSystemConfiguration.Id attribute in chapter TargetSystemConfiguration [▶ 25].

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	SafeAddress	IoPath
minOcc	0	0
maxOcc	1	1
Data type	xs:int	xs:string
Contents	simple	simple
Description	Safe address of the device	Complete I/O path within the TwinCAT 3 System Manager to remap alias devices and target systems

As XML code, this looks as follows, for example:

```
<TargetSystem BaseId="t1">
  <SafeAddress>1</SafeAddress>
  <IoPath>TIID^Device 2 (EtherCAT)^Term 30 (AX8620-0000-0104)^Drive 33 (AX8206-0210-0104)</IoPath>
</TargetSystem>
```

7.1.3.2 AliasDevices

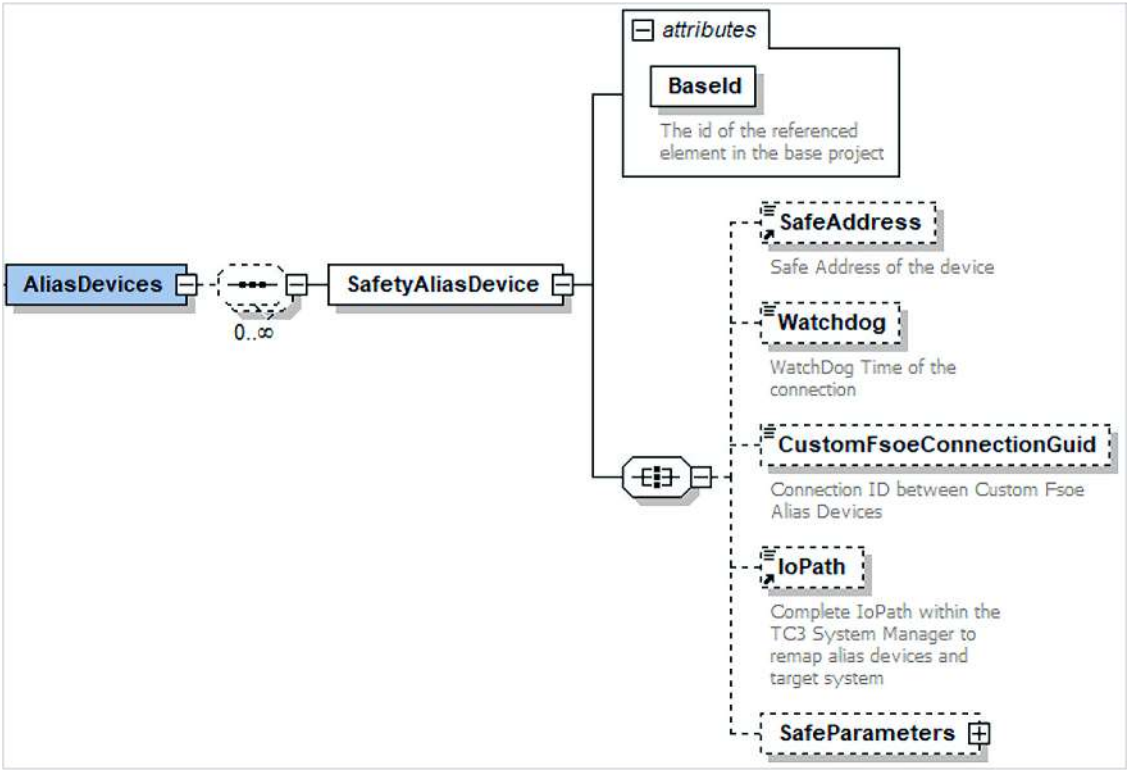


Fig. 44: AliasDevices

For a valid XML file, the order of the elements must be strictly adhered to.

Element	SafetyAliasDevice
minOcc	1
maxOcc	1
Contents	sophisticated
Description	Secure alias device

Attribute	BaseId
Data type	xs:IDREF
Use	required
Description	ID of the reference element in the base project See SafetyAliasDevice.Id attribute in chapter SafetyAliasDevice [▶ 57].

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	SafeAddress	Watchdog	CustomFsoe-ConnectionGuid	IoPath	SafeParameters
minOcc	0	0	0	0	0
maxOcc	1	1	1	1	1
Data type	xs:int	xs:unsignedShort	xs:string	xs:string	/
Contents	simple	simple	simple	simple	sophisticated
Description	Safe address of the device	Watchdog time of the connection	Connection ID between user-defined FSoE alias devices	Complete I/O path within the TwinCAT 3 System Manager to remap alias devices and target systems	Safe parameters of the device See chapter SafeParameters [▶ 73].

As XML code, this looks as follows, for example:

```
<AliasDevices>
  <SafetyAliasDevice BaseId="g1_a3">
    <SafeAddress>123</SafeAddress>
    <Watchdog>100</Watchdog>
    <CustomFsoeConnectionGuid>44ec90a0-1c84-4e8a-8a28-20811c35db74</CustomFsoeConnectionGuid>
  </SafetyAliasDevice>
  <SafetyAliasDevice BaseId="g1_a6">
    <SafeAddress>23</SafeAddress>
    <Watchdog>100</Watchdog>
    <IoPath>TIID^Device 2 (EtherCAT)^Term 15 (EK1101)^Term 1 (EL1918)^Module 1 (FSOE)</IoPath>
    <SafeParameters>
    </SafeParameters>
  </SafetyAliasDevice>
</AliasDevices>
```

7.1.3.2.1 SafeParameters

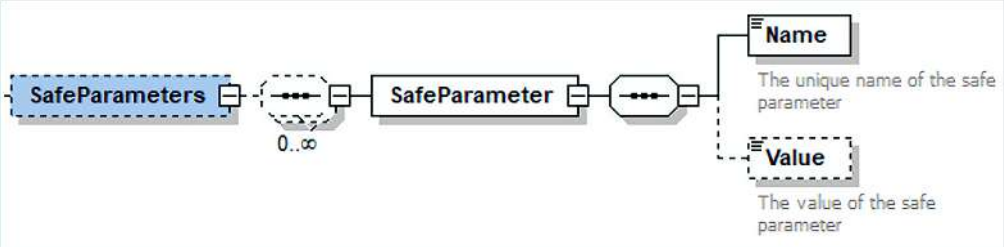


Fig. 45: SafeParameters

For a valid XML file, the order of the elements must be strictly adhered to.

Element	Safe Parameter
minOcc	1
maxOcc	1
Contents	sophisticated
Description	Safe parameter of the device

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Name	Value
minOcc	1	0
maxOcc	1	1
Data type	xs:string	xs:string
Contents	simple	simple
Description	Unique name of the safe parameter	Value of the safe parameter

As XML code, this looks as follows, for example:

```
<SafeParameters>
  <SafeParameter>
    <Name>32768_1_8_16</Name>
    <Value>0</Value>
  </SafeParameter>
</SafeParameters>
```

7.1.3.3 FbParameters

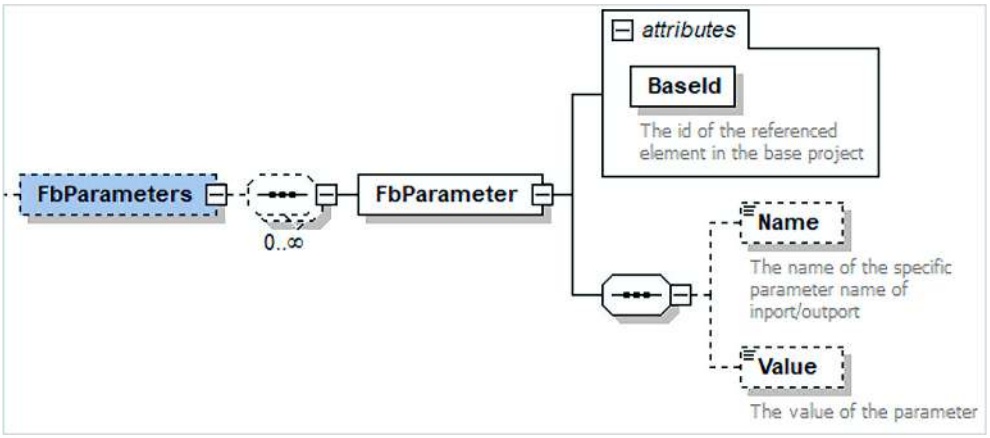


Fig. 46: FbParameters

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	FbParameter
minOcc	1
maxOcc	1
Contents	sophisticated
Description	Function block parameters

Attributes	BaseId
Data type	xs:string
Use	required
Description	ID of the reference element in the base project See • FunctionBlock.Id attribute in chapter FunctionBlock [▶ 35], • FunctionBlock.Outports.Element.Id attribute in chapter Outports [▶ 41], • FunctionBlock.Inports.Element.Id attribute in chapter Inports [▶ 43], • UserFunctionBlock.ParameterPorts.Element.Id attribute in chapter ParameterPorts [▶ 46].

For a valid XML file, the order of the elements must be strictly adhered to.

Elements	Name	Value
minOcc	0	0
maxOcc	1	1
Data type	xs:string	xs:string
Contents	simple	simple
Description	Name of the specific parameter at the inport or outport	Value of the parameter

7.2 ID rules

ID designation

The ID rules listed are recommendations. You can freely name the IDs. Pay attention to clarity.

The TwinSAFE Editor uses the following convention for generating values for the "Id" attribute:

ID	Format	Link
Unique identification number of the reference	r1, r2, …	See chapter References [▶ 24].
Identification number of the target system	t1, t2, …	See chapter TargetSystemConfiguration [▶ 25].
Identification number of an element in the output process image of the target system	TargetSystemConfiguration.Id + "_o1", TargetSystemConfiguration.Id + "_o2", ... Example: t1_o1	See chapter TxPdo [▶ 29].
Identification number of an element in the input process image of the target system	TargetSystemConfiguration.Id + "_i1", TargetSystemConfiguration.Id + "_i2", ... Example: t1_i1	See chapter RxPdo [▶ 31].
Identification number of a user function block definition	u1, u2, ...	See chapter UserFbs [▶ 32].
Identification number of a network	Within a user FB definition: UserFB.Id + "_n1", UserFB.Id + "_n2", ... Example: u1_n1 Within a group: TwinSAFEGroup.Id + "_n1", TwinSAFEGroup.Id + "_n2", ... Example: g1_n1	See chapter Network [▶ 34].
Identification number of a function block	Network.Id + "_f1", Network.Id + "_f2", ... Examples: g1_n1_f1, u1_n1_f1	See chapter FunctionBlock [▶ 35].
Identification number of an element in the output process image of a function block	FunctionBlock.Id + "_o1", FunctionBlock.Id + "_o2", ... Examples: g1_n1_f1_o1, u1_n1_f1_o1	See chapter Outports [▶ 41].
Identification number of an element in the input process image of a function block	FunctionBlock.Id + "_i1", FunctionBlock.Id + "_i2", ... Examples: g1_n1_f1_i1, u1_n1_f1_i1	See chapter Inports [▶ 43].
Identification number of a user function block	Network.Id + "_u1", Network.Id + "_u2", ... Examples: g1_n1_u1, u1_n1_u1	See chapter UserFunctionBlock [▶ 43].
Identification number of a parameter port	UserFB.Id + "_p1", UserFB.Id + "_p2", ... Examples: g1_n1_u1_p1, u1_n1_u1_p1	See chapter ParameterPorts [▶ 46].
Identification number of an element in the input process image of a user function block	UserFB.Id + "_i1", UserFB.Id + "_i2", ... Examples: g1_n1_u1_i1, u1_n1_u1_i1	See chapter Inports [▶ 47].

ID	Format	Link
Identification number of an element in the output process image of a user function block	UserFB.Id + "_o1", UserFB.Id + "_o2", ... Examples: g1_n1_u1_o1, u1_n1_u1_o1	See chapter Outports [▶ 48].
Identification number of a TwinSAFE group	g1, g2, ...	See chapter TwinSAFEGroups [▶ 51].
Identification number of an element in the output process image of a TwinSAFE group	TwinSAFEGroup.Id + "_o1", TwinSAFEGroup.Id + "_o2", ... Example: g1_o1	See chapter GroupOutputs [▶ 54].
Identification number of an element in the input process image of a TwinSAFE group	TwinSAFEGroup.Id + "_i1", TwinSAFEGroup.Id + "_i2", ... Example: g1_i1	See chapter GroupInputs [▶ 55].
Identification number of a safety alias device	TwinSAFEGroup.Id + "_a1", TwinSAFEGroup.Id + "_a2", ... Example: g1_a1	See chapter SafetyAliasDevice [▶ 57].
Identification number of DynamicPortionSafeAddress	SafetyAliasDevice.Id + "_d" Example: g1_a1_d	See chapter DynamicPortionSafeAddress [▶ 61]. See chapter SafetyAliasDevice [▶ 57].
Identification number of an element in the input process image of a safety alias device	SafetyAliasDevice.Id + "_i1", SafetyAliasDevice.Id + "_i2", ... If ConnectionType = Local: SafetyAliasDevice.Id + "_li1", SafetyAliasDevice.Id + "_li2", ... Examples: g1_a1_i1, g1_a1_li1	See chapter RxPdo [▶ 62].
Identification number of an element in the output process image of a safety alias device	SafetyAliasDevice.Id + "_o1", SafetyAliasDevice.Id + "_o2", ... If ConnectionType = Local: SafetyAliasDevice.Id + "_lo1", SafetyAliasDevice.Id + "_lo2", ... Examples: g1_a1_o1, g1_a1_lo1	See chapter TxPdo [▶ 62].
Identification number of ComErrAck	SafetyAliasDevice.Id + "_c" Example: g1_a1_c	See chapter ComErrAck [▶ 62]. See chapter SafetyAliasDevice [▶ 57].
Identification number of a standard alias device	TwinSAFEGroup.Id + "_a1", TwinSAFEGroup.Id + "_a2", ... Example: g1_a1	See chapter StandardAliasDevice [▶ 63].
Identification number of an element in the output process image of a standard alias device	StandardAliasDevice.Id + "_o1", StandardAliasDevice.Id + "_o2", ... Example: g1_a1_o1	See chapter TxPdo [▶ 64].
Identification number of an element in the input process image of a standard alias device	StandardAliasDevice.Id + "_i1", StandardAliasDevice.Id + "_i2", ... Example: g1_a1_i1	See chapter RxPdo [▶ 65].
Identification number of a derived port	Inports.Element.Id or Outports.Element.Id or ParameterPorts.Element.Id Examples: u2_n1_u2_i1, g1_n1_u3_o3, g1_n2_u6_p1	See chapter UserFbPortRelations [▶ 65].
I of the Multi Settings	ms_1, ms_2, ...	See chapter Multi Settings [▶ 70].

7.3 FBTypes

This appendix contains types for the unique identification of the function blocks. See chapter FunctionBlock [▶ 35].

Name	Type
FBAnd	34
FBOr	35
FBDecouple	36
FBOpmode	37
FBRs	41
FBSr	42
FBTof	43
FBTon	44
FBTon2	4402
FBEdm	39
FBEstop	32
FBMon	33
FBTwohand	38
FBMuting	40
FBConnShutdown	45
FBXor	46
FBAdd	65
FBSub	66
FBMul	67
FBDiv	68
FBCounter	69
FBLimit	70
FBCompare	64
FBScaling	71
FBCamMonitor	72
FBLoadSensing	74
FBSpeed	73
FBViolationCnt	75
FBSLI	76
FBSLI2	7602
FBEnvelope	77
FBSLP	78
FBSBT	79
FBAdvPosMon	80

7.4 PortIds

In this chapter you will find the IDs associated with the ports for unique identification

- of a TwinSAFE group (see chapter TwinSAFEGroups [▶ 51]),
- of a function block (see chapter UserFbs [▶ 32]),
- of a group port (see chapter GroupOutputs [▶ 54]).

The port ID for a function block port consists of

- (A) an identifier that indicates whether it is an input (0x01) or output (0x02) and
- (B) a port identifier.

These are combined by (in pseudocode):

Group port ID = (A << 24 | B << 16)

Alias	PortId	
	Hexadecimal	Decimal
Inports		
RunStop	0x01010000	16842752
ErrAck	0x01020000	16908288
ModuleFault	0x01030000	16973824
Outports		
FbErr	0x02010000	33619968
ComErr	0x02020000	33685504
OutErr	0x02030000	33751040
OtherErr	0x02040000	33816576
ComStartup	0x02050000	33882112
FbDeactive	0x02060000	33947648
FbRun	0x02070000	34078720
InRun	0x02080000	34078720

The port ID for a function block port consists of

- (A) an identifier that indicates whether it is an input (0x01) or output (0x02),
- (B) a port identifier and
- (C) of the SubType ID of the function block.

These are combined by (in pseudocode):

Port ID = (A << 24 | B << 16 | C)

7.4.1 FBAnd (SubType 0x22)

PortName	PortId	
	Hexadecimal	Decimal
AndIn1	0x01080022	17301538
AndIn2	0x01090022	17367074
AndIn3	0x010a0022	17432610
AndIn4	0x010b0022	17498146
AndIn5	0x010c0022	17563682
AndIn6	0x010d0022	17629218
AndIn7	0x010e0022	17694754
AndIn8	0x010f0022	17760290
AndOut	0x02010022	33620002

7.4.2 FBOr (SubType 0x23)

PortName	PortId	
	Hexadecimal	Decimal
OrIn1	0x01080023	17301539
OrIn2	0x01090023	17367075
OrIn3	0x010a0023	17432611
OrIn4	0x010b0023	17498147
OrIn5	0x010c0023	17563683
OrIn6	0x010d0023	17629219
OrIn7	0x010e0023	17694755
OrIn8	0x010f0023	17760291
OrOut	0x02010023	33620003

7.4.3 FBDecouple (SubType 0x24)

PortName	PortId	
	Hexadecimal	Decimal
DecIn1	0x01080024	17301540
DecIn2	0x01090024	17367076
DecIn3	0x010a0024	17432612
DecIn4	0x010b0024	17498148
DecIn5	0x010c0024	17563684
DecIn6	0x010d0024	17629220
DecIn7	0x010e0024	17694756
DecIn8	0x010f0024	17760292
DecOut1	0x02080024	34078756
DecOut2	0x02090024	34144292
DecOut3	0x020a0024	34209828
DecOut4	0x020b0024	34275364
DecOut5	0x020c0024	34340900
DecOut6	0x020d0024	34406436
DecOut7	0x020e0024	34471972
DecOut8	0x020f0024	34537508

7.4.4 FBOpmode (SubType 0x25)

PortName	PortId	
	Hexadecimal	Decimal
Restart	0x01000025	16777253
OpIn1	0x01080025	17301541
OpIn2	0x01090025	17367077
OpIn3	0x010a0025	17432613
OpIn4	0x010b0025	17498149
OpIn5	0x010c0025	17563685
OpIn6	0x010d0025	17629221
OpIn7	0x010e0025	17694757
OpIn8	0x010f0025	17760293
Error	0x02000025	33554469
OpOut1	0x02080025	34078757

PortName	PortId	
	Hexadecimal	Decimal
OpOut2	0x02090025	34144293
OpOut3	0x020a0025	34209829
OpOut4	0x020b0025	34275365
OpOut5	0x020c0025	34340901
OpOut6	0x020d0025	34406437
OpOut7	0x020e0025	34471973
OpOut8	0x020f0025	34537509

7.4.5 FBRs (SubType 0x29)

PortName	PortId	
	Hexadecimal	Decimal
Reset	0x01080029	17301545
Set	0x01090029	17367081
RsOut	0x02000029	33554473

7.4.6 FBSr (SubType 0x2a)

PortName	PortId	
	Hexadecimal	Decimal
Set	0x0109002a	17367082
Reset	0x0108002a	17301546
SrOut	0x0200002a	33554474

7.4.7 FBTof (SubType 0x2b)

PortName	PortId	
	Hexadecimal	Decimal
TofIn	0x0108002b	17301547
TofOut	0x0201002b	33620011

7.4.8 FBTone (SubType 0x2c)

PortName	PortId	
	Hexadecimal	Decimal
TonIn	0x0108002c	17301548
TonOut	0x0201002c	33620012

7.4.9 FBTon2 (SubType 0x1132)

PortName	PortId	
	Hexadecimal	Decimal
TonIn	0x01081132	17305906
Enable	0x01091132	17371442
TonOut	0x02011132	33624370

7.4.10 FBEdm (SubType 0x27)

PortName	PortId	
	Hexadecimal	Decimal
Mon1	0x01080027	17301543
Mon2	0x01090027	17367079
Error	0x02000027	33554471

7.4.11 FBEstop (SubType 0x20)

PortName	PortId	
	Hexadecimal	Decimal
Restart	0x01000020	16777248
EDM1	0x01010020	16842784
EDM2	0x01020020	16908320
EStopIn1	0x01080020	17301536
EStopIn2	0x01090020	17367072
EStopIn3	0x010a0020	17432608
EStopIn4	0x010b0020	17498144
EStopIn5	0x010c0020	17563680
EStopIn6	0x010d0020	17629216
EStopIn7	0x010e0020	17694752
EStopIn8	0x010f0020	17760288
Error	0x02000020	33554464
EStopOut	0x02010020	33620000
EStopDelOut	0x02020020	33685536

7.4.12 FBMon (SubType 0x21)

PortName	PortId	
	Hexadecimal	Decimal
Restart	0x01000021	16777249
EDM1	0x01010021	16842785
EDM2	0x01020021	16908321
MonIn1	0x01080021	17301537
MonIn2	0x01090021	17367073
MonIn3	0x010a0021	17432609
MonIn4	0x010b0021	17498145
Secure1	0x010c0021	17563681
Secure2	0x010d0021	17629217
Error	0x02000021	33554465
MonOut	0x02010021	33620001
MonDelOut	0x02020021	33685537

7.4.13 FBTwohand (SubType 0x26)

PortName	PortId	
	Hexadecimal	Decimal
TwoHand1	0x01080026	17301542
TwoHand2	0x01090026	17367078

PortName	PortId	
	Hexadecimal	Decimal
TwoHand3	0x010a0026	17432614
TwoHand4	0x010b0026	17498150
Error	0x02000026	33554470
TwoHandOut	0x02010026	33620006

7.4.14 FBMuting (SubType 0x28)

PortName	PortId	
	Hexadecimal	Decimal
Enable	0x01000028	16777256
EDM1	0x01010028	16842792
EDM2	0x01020028	16908328
Muting1	0x01080028	17301544
Muting2	0x01090028	17367080
OSSDIn1	0x010a0028	17432616
OSSDIn2	0x010b0028	17498152
Muting3	0x010c0028	17563688
Muting4	0x010d0028	17629224
Error	0x02000028	33554472
MutingActive	0x02010028	33620008
MuteOut	0x02020028	33685544
MuteDelOut	0x02030028	33751080

7.4.15 FBConnShutdown (SubType 0x2d)

PortName	PortId	
	Hexadecimal	Decimal
Deactivate1	0x0108002d	17301549
Deactivate2	0x0109002d	17367085
Error	0x0200002d	33554477
Deactivated	0x0201002d	33620013

7.4.16 FBXor (SubType 0x2e)

PortName	PortId	
	Hexadecimal	Decimal
Xor1In1	0x0100002e	16777262
Xor2In1	0x0108002e	17301550
Xor1In2	0x0101002e	16842798
Xor2In2	0x0109002e	17367086
Xor1In3	0x0102002e	16908334
Xor2In3	0x010a002e	17432622
Xor1In4	0x0103002e	16973870
Xor2In4	0x010b002e	17498158
Xor1In5	0x0104002e	17039406
Xor2In5	0x010c002e	17563694
Xor1In6	0x0105002e	17104942
Xor2In6	0x010d002e	17629230

PortName	PortId	
	Hexadecimal	Decimal
Xor1In7	0x0106002e	17170478
Xor2In7	0x010e002e	17694766
Xor1In8	0x0107002e	17236014
Xor2In8	0x010f002e	17760302
XorOut1	0x0208002e	34078766
XorOut2	0x0209002e	34144302
XorOut3	0x020a002e	34209838
XorOut4	0x020b002e	34275374
XorOut5	0x020c002e	34340910
XorOut6	0x020d002e	34406446
XorOut7	0x020e002e	34471982
XorOut8	0x020f002e	34537518

7.4.17 FBAdd (SubType 0x41)

PortName	PortId	
	Hexadecimal	Decimal
AnalogIn1	0x01100041	17825857
AnalogIn2	0x01110041	17891393
Error	0x02000041	33554497
AnalogOut	0x02100041	34603073

7.4.18 FBSub (SubType 0x42)

PortName	PortId	
	Hexadecimal	Decimal
AnalogIn1	0x01100042	17825858
AnalogIn2	0x01110042	17891394
Error	0x02000042	33554498
AnalogOut	0x02100042	34603074

7.4.19 FBMul (SubType 0x43)

PortName	PortId	
	Hexadecimal	Decimal
AnalogIn1	0x01100043	17825859
AnalogIn2	0x01110043	17891395
Error	0x02000043	33554499
AnalogOut	0x02100043	34603075

7.4.20 FBDiv (SubType 0x44)

PortName	PortId	
	Hexadecimal	Decimal
AnalogIn1	0x01100044	17825860
AnalogIn2	0x01110044	17891396
Error	0x02000044	33554500
AnalogOut	0x02100044	34603076

7.4.21 FBCounter (SubType 0x45)

PortName	PortId	
	Hexadecimal	Decimal
Reset	0x01000045	16777285
CountUp	0x01010045	16842821
CountDown	0x01020045	16908357
Error	0x02000045	33554501
CounterOut	0x02010045	33620037
CounterZero	0x02020045	33685573
ActValue	0x02100045	34603077

7.4.22 FBLimit (SubType 0x46)

PortName	PortId	
	Hexadecimal	Decimal
AnalogIn	0x01100046	17825862
MinValue	0x01110046	17891398
MaxValue	0x01120046	17956934
Error	0x02000046	33554502
InLimit	0x02010046	33620038
BelowMin	0x02020046	33685574
AboveMax	0x02030046	33751110

7.4.23 FBCompare (SubType 0x40)

PortName	PortId	
	Hexadecimal	Decimal
CompIn1	0x01100040	17825856
CompIn2	0x01110040	17891392
CompIn3	0x01120040	17956928
CompIn4	0x01130040	18022464
CompIn5	0x01140040	18088000
Error	0x02000040	33554496
CompOut	0x02100040	34603072

7.4.24 FBScale (SubType 0x47)

PortName	PortId	
	Hexadecimal	Decimal
AnalogIn	0x01100047	17825863
Error	0x02000047	33554503
AnalogOut	0x02100047	34603079
StuckAtError	0x02010047	33620039

7.4.25 FBCamMonitor (SubType 0x48)

PortName	PortId	
	Hexadecimal	Decimal
Automatic	0x01000048	16777288

PortName	PortId	
	Hexadecimal	Decimal
BackwardsMove	0x01010048	16842824
Reset	0x01020048	16908360
PressStarted	0x01030048	16973896
OverrunCAM	0x01040048	17039432
UpwardCAM	0x01050048	17104968
SettingMode	0x01060048	17170504
Position	0x01100048	17825864
TDC1-Lower Limit	0x01110048	17891400
TDC1-Upper Limit	0x01120048	17956936
TDC2-Lower Limit	0x01130048	18022472
TDC2-Upper Limit	0x01140048	18088008
BDC-Lower Limit	0x01150048	18153544
BDC-Upper Limit	0x01160048	18219080
OverrunMax	0x01170048	18284616
Error	0x02000048	02000048
CAMMonOk	0x02010048	33620040
UpwardsMove	0x02020048	33685576
TDC	0x02030048	33751112
BDC	0x02040048	33816648
OverrunTDC	0x02100048	34603080
OverrunStop	0x02110048	34668616

7.4.26 FBLoadSensing (SubType 0x4a)

PortName	PortId	
	Hexadecimal	Decimal
Inactive	0x0106004a	17170506
AnalogInX	0x0110004a	17825866
AnalogInY	0x0111004a	17891402
Valid	0x0201004a	33620042
Warning	0x0202004a	33685578

7.4.27 FBSpeed (SubType 0x49)

PortName	PortId	
	Hexadecimal	Decimal
Position	0x01100049	17825865
Error	0x02000049	33554505
Speed	0x02100049	34603081

7.4.28 FBViolationCnt (SubType 0x4b)

PortName	PortId	
	Hexadecimal	Decimal
Input	0x0100004b	16777291
Enable	0x0101004b	16842827
InputOK	0x0201004b	33620043
ActViolationCnt	0x0210004b	34603083

7.4.29 FBSLI (SubType 0x4c)

PortName	PortId	
	Hexadecimal	Decimal
SLI	0x0100004c	16777292
Position	0x0110004c	17825868
Error	0x0200004c	33554508
SLIActive	0x0201004c	33620044
PositionDiff	0x0210004c	34603084

7.4.30 FBSLI2 (SubType 0x1db2)

PortName	PortId	
	Hexadecimal	Decimal
SLI	0x01001db2	16784818
Position	0x01101db2	17833394
Error	0x02001db2	33562034
SLIActive	0x02011db2	33627570
BelowMin	0x02021db2	33693106
AboveMax	0x02031db2	33758642
PositionDiff	0x02101db2	34610610

7.4.31 FBEnvelope (SubType 0x4d)

PortName	PortId	
	Hexadecimal	Decimal
SafeFunction	0x0100004d	16777293
InValue	0x0110004d	17825869
Error	0x0200004d	33554509
SafeFunctionOut	0x0201004d	33620045

7.4.32 FBSLP (SubType 0x4e)

PortName	PortId	
	Hexadecimal	Decimal
SafePosValid	0x0100004e	16777294
SingleturnPos	0x0110004e	17825870
MultiturnPos	0x0111004e	17891406
SingleturnPosLL	0x0112004e	17956942
MultiturnPosLL	0x0113004e	18022478
SingleturnPosUL	0x0114004e	18088014
MultiturnPosUL	0x0115004e	18153550
Error	0x0200004e	33554510
PositionInLimit	0x0202004e	33685582
BelowMin	0x0203004e	33751118
AboveMax	0x0204004e	33816654

7.4.33 FBSBT (SubType 0x4f)

PortName	PortId	
	Hexadecimal	Decimal
SBT	0x0100004f	16777295
SBCIn	0x0101004f	16842831
SingleturnPos	0x0110004f	17825871
ActCurrentIq	0x0111004f	17891407
MinCurrentIq	0x0112004f	17956943
TestError	0x0201004f	33620047
BrakeValid	0x0202004f	33685583
RemainingTime	0x0210004f	34603087

7.4.34 FBAdvPosMon (SubType 0x50)

PortName	PortId	
	Hexadecimal	Decimal
Restart	0x01000050	16777296
Enable	0x01080050	17301584
Error	0x02000050	33554512
Active	0x02010050	33620048
Wait	0x02020050	33685584
OverRange	0x02030050	33751120

7.5 ConnectionType

The ConnectionType contains information about the safe communication protocol of the SafetyAliasDevice and the role of the communication partner in this project.

ConnectionType	Communication protocol	Role of the communication partner
FSoEMaster	FSoE See document [5] in chapter References [▶ 7].	Master
FSoESlave	FSoE	Slave
TwinSafeScCrc1master	TwinSAFE SC with the generator polynomial 0x7B0F See document [5] in chapter References [▶ 7].	Master
TwinSafeScCrc2master	TwinSAFE SC with the generator polynomial 0x571F	Master
TwinSafeScCrc3master	TwinSAFE SC with the generator polynomial 0x1F95	Master
TwinSafeScCrc4master	TwinSAFE SC with the generator polynomial 0x53F1	Master
TwinSafeScCrc5master	TwinSAFE SC with the generator polynomial 0xF1D5	Master
TwinSafeScCrc6master	TwinSAFE SC with the generator polynomial 0x663B	Master
TwinSafeScCrc7master	TwinSAFE SC with the generator polynomial 0xB8CD	Master
TwinSafeScCrc8master	TwinSAFE SC with the generator polynomial 0xE1BD	Master
TwinSafeScCustomCrcMaster	TwinSAFE SC with a self-selected generator polynomial See the "CustomCRC" element in chapter SafetyAliasDevice [▶ 57].	Master
PROFIsafeMaster	PROFIsafe See document [5] in chapter References [▶ 7].	Master
PROFIsafeSlave	PROFIsafe	Slave
Local	This setting indicates that this SafetyAliasDevice is part of the TargetSystemConfiguration. See chapter TargetSystemConfiguration [▶ 25].	

7.5.1 ConnectionType in TE9000

You can find the ConnectionType in the TwinCAT Safety Editor TE9000 via the "Connection" and the "Linking" tab of a SafetyAliasDevice.

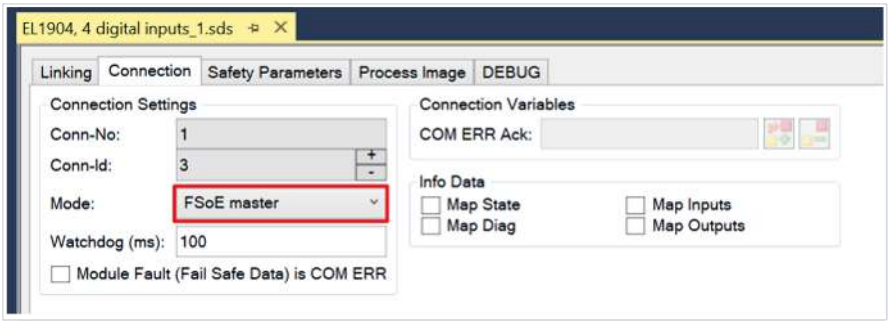


Fig. 47: ConnectionType EL1904

In the figure you can see that the SafetyAliasDevice EL1904 is defined as "FSoE master".

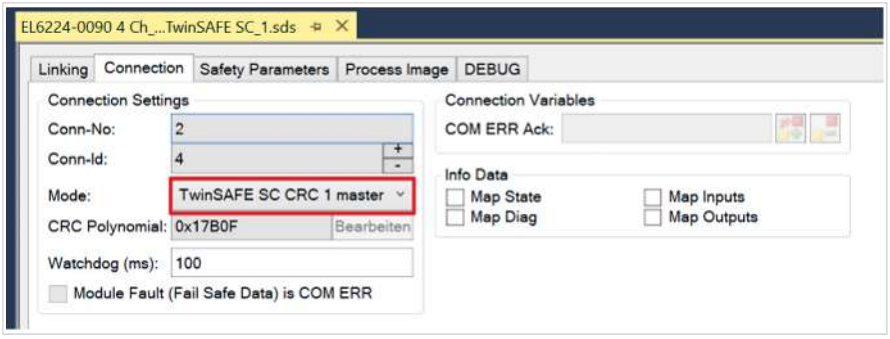


Fig. 48: ConnectionType TwinSAFE SC

For TwinSAFE SC, "TwinSAFE SC CRC 1 master" is selected as the mode.

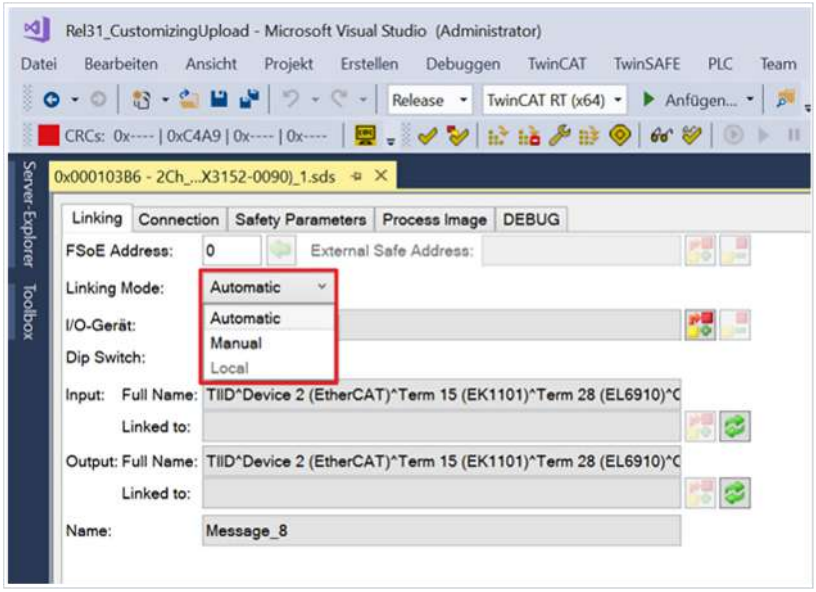


Fig. 49: Linking mode

You can find the settings for linking mode via the "Linking" tab.

7.6 LogicType

In TwinCAT 3, you have the option of setting the type in the CustomFSoEConnections. The type corresponds to the LogicType. This option does not exist for all other SafetyAliasDevices.

This setting is used to announce the communication partner. You can set the following values:

- None
- KL6904

- EL6900EL6930 (if the communication partner is an EL6900 or EL6930)
- EL6910EJ6910 (if the communication partner is an EL6910 or EJ6910)
- All target systems named in TargetSystemConfiguration/SubType except EL6900, EL6930, EL6910 and EJ6910

Depending on the selection, additional backup & restore functions are enabled or disabled. For further information, please refer to the chapter BackupRestore [▶ 28].

If "FsoEMaster" is set as the ConnectionType for the SafetyAliasDevice and the communication partner is a device from the possible device list, the SafeParameters configuration is automatically created for this communication. For more information about ConnectionType and SafetyParameters, see the chapters SafetyAliasDevice [▶ 57] and SafetyParameters [▶ 61].

For the EL6900 and EL6930 target systems, the LogicType is included in the project CRC for reasons of compatibility with TwinCAT 2.

7.6.1 LogicType in TE9000

You add the SafetyAliasDevice "Custom FSoE Connection" via the "Add" dialog:

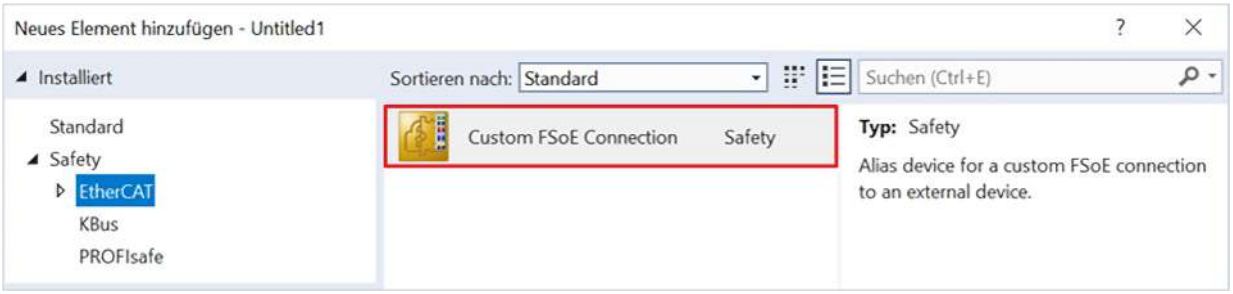


Fig. 50: Adding element

The LogicType setting can be found on the "Connection" tab of the SafetyAliasDevice:

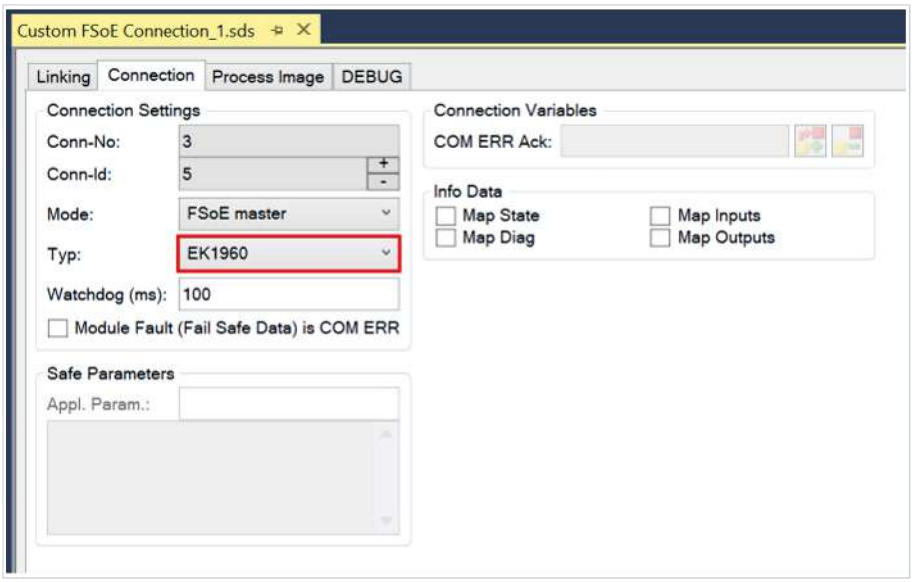


Fig. 51: LogicType

8 Generate compatible safety project

This chapter describes the creation of a TwinCAT 3-compatible safety project outside the TwinCAT 3 development environment as an example.

You can use the information from the previous chapters for this. As a rule, you do not need all the elements from the XSD file.

Information

You do not need to create the TargetSystemProcessImage element. The TwinSAFE Editor automatically creates the process image for the base project after importing the XML file.

Due to the complexity of the UserFunctionBlock, UserFbPortRelations and MultiSettings areas, automatic creation of a safety project with these elements is not recommended by Beckhoff and may not be possible.

8.1 Sample EStop function

This chapter describes how to create a safety project using the example "2.3 ESTOP function variant 3 (category 4, PL e)" from document [5] in chapter References [▶ 7].

First create an XML file with the name "SafetyProject_Estop_Function.xml". The further procedure is based on the structure of the XSD file. This can be found in this and the following chapters.

TwinCATExport

You need the TwinCATExport attribute [▶ 21] "Version" for this element. The version depends on the TwinCAT version used and the TE9000 version. See chapters System requirements [▶ 15] and System limits [▶ 15] for more information.

In this sample, fix the version to 0.31.

```
<TwinCATExport Version="0.31">
</TwinCATExport>
```

8.1.1 BaseConfiguration

The basic configuration contains the complete description of the safety project.

GeneralInformation

Assign the project name here. All other details are optional.

As you are creating a new project, the project CRC is not yet known and therefore cannot be specified.

```
<GeneralInformation>
  <ProjectName>EstopFunction</ProjectName>
  <Author>Person</Author>
  <InternalProjectName>EstopFunction</InternalProjectName>
</GeneralInformation>
```

References

No reference is used for this sample.

TargetSystemConfiguration

In this sample, the TwinSAFE Logic EL6900 is used as the target system. To do this, define the attributes "Id", "Type" and "SubType" as well as the elements "SafeAddress" and "VersionNumber".

All other elements and attributes are optional. The Id, the Type and the SubType can be found in the previous chapters ID rules [▶ 75] and TargetSystemConfiguration [▶ 25].
For SafeAddress and VersionNumber, first select any valid numerical values.

```
<TargetSystemConfiguration Id="t1" Type="Hardware" SubType="EL6900">
  <SafeAddress>1</SafeAddress>
  <VersionNumber>1</VersionNumber>
</TargetSystemConfiguration>
```

ApplicationConfiguration

As the configuration of the application is more extensive, this element is described in more detail in the next chapter.

8.1.1.1 ApplicationConfiguration

Here you define the essential logic.

```
<ApplicationConfiguration>
</ApplicationConfiguration>
```

UserFbs

No user function blocks are used for this sample.

TwinSAFEGroups

Every safety project always requires at least one TwinSAFE group. One TwinSAFE group is sufficient for this sample with two function blocks.

As the configuration of the TwinSAFE group is more extensive, this element is described in more detail in the next chapter.

8.1.1.1.1 TwinSAFEGroup

The value for the "Id" attribute can be found in the chapter ID rules [▶ 75]. The OrderId usually starts at "0" and increases by the value "1" for each TwinSAFE group.

```
<TwinSAFEGroups>
  <TwinSAFEGroup Id="g1" OrderId="0">
  </TwinSAFEGroup>
</TwinSAFEGroups>
```

8.1.1.1.1.1 GroupInputs

In this sample, the use of the ErrAck group input port and the RunStop input is specified for the EL6900 target system.

The values for the "Id" attribute and the "PortId" and "Alias" elements can be found in the chapters ID rules [▶ 75] and PortIds [▶ 78]. The "Alias" element is optional.

```
<GroupInputs>
  <Element Id="g1_i1">
    <PortId>16842752</PortId>
  <Alias>RunStop</Alias>
  </Element>
  <Element Id="g1_i2">
    <PortId>16908288</PortId>
    <Alias>ErrAck</Alias>
  </Element>
</GroupInputs>
```

8.1.1.1.1.2 AliasDevices

In this sample, you are using the EL1904 and EL2904 TwinSAFE I/O devices. These devices are described in the TwinSAFE Editor as SafetyAliasDevices. TwinCAT 3 interprets all FSoE, TwinSAFE SC or PROFIsafe devices as SafetyAliasDevices.

You also need 4 StandardAliasDevices.

- 1 x StandardAliasDevice for the group input port ErrAck

- 1 x StandardAliasDevice for the RunStop group input
- 2 x StandardAliasDevice for linking to the 1st and 2nd input of an EL1004

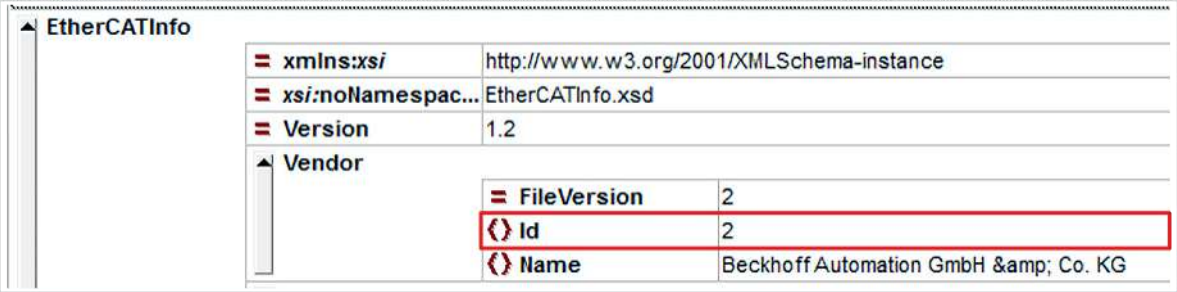
You set the values for the Id attributes using ID rules [▶ 75].

```
<AliasDevices>
  <SafetyAliasDevice Id="g1_a1">
  </SafetyAliasDevice>
  <SafetyAliasDevice Id="g1_a2">
  </SafetyAliasDevice>
  <StandardAliasDevice Id="g1_a3">
  </StandardAliasDevice>
  <StandardAliasDevice Id="g1_a4">
  </StandardAliasDevice>
  <StandardAliasDevice Id="g1_a5">
  </StandardAliasDevice>
  <StandardAliasDevice Id="g1_a6">
  </StandardAliasDevice>
</AliasDevices>
```

8.1.1.1.1.2.1 SafetyAliasDevice Id="g1_a1"

This SafetyAliasDevice is described by the EL1904. As the EL1904 is an EtherCAT device, you can take the VendorId from the ESI file. See document [2] at References [▶ 7].

VendorId (EtherCATInfo.Vendor.Id):



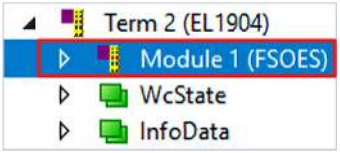
For Beckhoff EtherCAT devices, the VendorId is always 2.

Type and SubType

You can read the Type and SubType from the TwinCAT System Manager if they are not yet known. You can choose between 2 procedures.

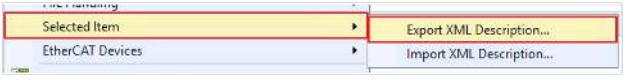
First procedure: Export module information:

Proceed as follows:



1. Select the EL1904 safety module in TwinCAT 3

2. Select "TwinCAT" in the menu bar



3. Click on "Export XML description..." via the "Select element" button

4. Read the Type from the ItemType and the SubType from the ItemSubType from the XML file

```
<TreeItem>
  <ItemName>Module 1 (FSOES)</ItemName>
  <PathName>TIID^Device 2 (EtherCAT)^Term 2 (EL1904)^Module 1 (SFOES)</PathName>
```

```
  <ItemType>60</ItemType>
  <ItemId>50921473</ItemId>
  <ObjectId>#x03090001</ObjectId>
  <ItemSubType>190</ItemSubType>
  <ChildCount>0</ChildCount>
  <Disabled>false</Disabled>
  <TreeImageId>118</TreeImageId>
```

Second procedure: Export safety project

Another way to find out Type and SubType is to create a TwinCAT 3 Safety project with the corresponding SafetyAliasDevice and export it as TwinCAT 3 Xml format.

Take the Type and SubType from the exported XML file.

Define elements

Here you define the following elements:

- AliasName
  - File name of the AliasDevice EL1904
- ConnectionId
  - Identification of the EL1904 as a communication device
- SafeAddress
  - In this sample, address "2" is selected for the SafetyAliasDevice, as address "1" has already been assigned to the EL6900.
- Watchdog
  - In this sample, the typical value "100 [ms]" is selected.

```
<SafetyAliasDevice Id="g1_a1">
  <VendorId>2</VendorId>
  <Type>60</Type>
  <SubType>190</SubType>
  <AliasName>EL1904</AliasName>
  <ConnectionId>1</ConnectionId>
  <SafeAddress>2</SafeAddress>
  <Watchdog>100</Watchdog>
</SafetyAliasDevice >
```

8.1.1.1.1.2.1.1 SafetyParameter

The emergency stop button is wired to the 3rd and 4th input of the EL1904. In addition, the testing of the inputs must be activated in order to increase the diagnostic coverage. You make this setting in the safety parameters.

To do this, open the ESI file (Beckhoff Elx9xx.xml) in which the EL1904 module is defined and navigate to the "SafetyParaMapping" element.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<EtherCATInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="EtherCATInfo.xsd" Version="1.2">
  <Vendor FileVersion="2">
  <Descriptions>
    <Groups>
    <Devices>
    <Modules>
      <Module Crc32="#xc1f11b2b">
        <Type ModuleIdent="190" ModuleClass="190">FSOES</Type>
        <Name>EL1904 FSoE Message</Name>
        <RxPdo Fixed="true" Sm="2">
        <TxPdo Fixed="true" Sm="3">
        <SafetyParaMapping Fixed="true">
        <Profile>
      </Module>
```

List the parameters in the order in which they are listed in the ESI file. Exceptions to this are the first three standard parameters and parameters without a name, which you do not need to worry about.

	Index	SubIndex	BitLen	Name	DataType
1	#x9001	7	16	LengthOfCommParameter	UINT
2	#x9001	4	16	WatchdogTime	UINT
3	#x9001	8	16	LengthOfApplParameter	UINT
4	#x8000	1	8	Operating Mode	USINT
5	#x8001	1	1	Sensor test Channel 1 active	BOOL
6	#x8001	2	1	Sensor test Channel 2 active	BOOL
7	#x8001	3	1	Sensor test Channel 3 active	BOOL
8	#x8001	4	1	Sensor test Channel 4 active	BOOL
9	#x0000	0	4		
10	#x8002	1	2	Logic of Channel 1 and 2	BIT2
11	#x8002	3	2	Logic of Channel 3 and 4	BIT2
12	#x0000	0	12		
13	#x10E0	1	16	Store Code	UINT
14	#x10E0	2	16	Project CRC	UINT

If you stick to the order of the ESI file, you can omit the "OrderId" attribute. In this sample, the parameter names are used for better assignment of the parameters to the values. These are not evaluated during import.

```
<SafetyParameters>
  <SafetyParameter Name="Operating Mode">0</SafetyParameter>
  <SafetyParameter Name="Sensor test Channel 1 active">0</SafetyParameter>
  <SafetyParameter Name="Sensor test Channel 2 active">0</SafetyParameter>
  <SafetyParameter Name="Sensor test Channel 3 active">1</SafetyParameter>
  <SafetyParameter Name="Sensor test Channel 4 active">1</SafetyParameter>
  <SafetyParameter Name="Logic of Channel 1 and 2">0</SafetyParameter>
  <SafetyParameter Name="Logic of Channel 3 and 4">0</SafetyParameter>
  <SafetyParameter Name="Store Code">0</SafetyParameter>
  <SafetyParameter Name="Project CRC">0</SafetyParameter>
</SafetyParameters>
```

8.1.1.1.1.2.1.2 RxPdo

The EL1904 TwinSAFE Terminal has 4 safe digital inputs. You must incorporate these into the "RxPdo" area.

First, turn to the 1st input. The value for the Id attribute can be found in the chapter ID rules [▶ 75]. You must also define the elements "Name", "Type" and "BitOffset", as well as the "Index" and "SubIndex", as the EL1904 is an ESI device.

Open the ESI file (Beckhoff ELx9xx.xml) and navigate to the "TxPdo" element.

	Index	SubIndex	BitLen	Name	DataType
1	#x6000	1	8	FSoE Slave CMD	USINT
2	#x6001	1	1	InputChannel1	BOOL
3	#x6001	2	1	InputChannel2	BOOL
4	#x6001	3	1	InputChannel3	BOOL
5	#x6001	4	1	InputChannel4	BOOL
6	#x0000	0	4		
7	#x6000	3	16	FSoE Slave CRC_0	UINT
8	#x6000	2	16	FSoE Slave ConnID	UINT

Here you will find the name, index and subindex and you can calculate the BitOffset from the BitLen. You select the type after viewing the BitLen of the DataType from the ESI file and the list of possible types in the XSD file. For the possible types, see the chapter PortIds [▶ 78].

The XML description for the first input then looks as follows:

```
<Element Id="g1_a1_i1">
  <Name>InputChannel1</Name>
  <Type>BIT</Type>
  <BitOffset>8</BitOffset>
  <Index>24577</Index>
  <SubIndex>1</SubIndex>
</Element>
```

Use the same principle to define the other inputs of the EL1904.

```
<Element Id="g1_a1_i2">
  <Name>InputChannel2</Name>
  <Type>BIT</Type>
  <BitOffset>9</BitOffset>
  <Index>24577</Index>
  <SubIndex>2</SubIndex>
</Element>
<Element Id="g1_a1_i3">
  <Name>InputChannel3</Name>
  <Type>BIT</Type>
  <BitOffset>10</BitOffset>
  <Index>24577</Index>
  <SubIndex>3</SubIndex>
</Element>
<Element Id="g1_a1_i4">
  <Name>InputChannel4</Name>
  <Type>BIT</Type>
  <BitOffset>11</BitOffset>
  <Index>24577</Index>
  <SubIndex>4</SubIndex>
</Element>
```

Finally, the BitSize for the "RxPdo" element is specified. To do this, add up all BitLen.

```
<RxPdo BitSize="48">
  <Element Id="g1_a1_i1">
  <Element Id="g1_a1_i2">
  <Element Id="g1_a1_i3">
  <Element Id="g1_a1_i4">
</RxPdo>
```

8.1.1.1.1.2.2 SafetyAliasDevice Id="g1_a2"

This SafetyAliasDevice is described by the EL2904. As the EL2904 is an EtherCAT device, proceed in the same way as for the EL1904 (g1_a1).

In this sample, you must activate the current measurement and the testing of the outputs. You make these settings in the "SafetyParameters" element.

The configuration for this StandardAliasDevice is as follows:

```
<SafetyAliasDevice Id="g1_a2">
  <VendorId>2</VendorId>
  <Type>60</Type>
  <SubType>290</SubType>
  <AliasName>EL2904</AliasName>
  <ConnectionId>2</ConnectionId>
  <SafeAddress>3</SafeAddress>
  <Watchdog>100</Watchdog>
  <SafetyParameters>
    <SafetyParameter Name="Standard outputs active">0</SafetyParameter>
    <SafetyParameter Name="Current measurement active">1</SafetyParameter>
    <SafetyParameter Name="Testing of outputs active">1</SafetyParameter>
    <SafetyParameter Name="Error acknowledge active">0</SafetyParameter>
    <SafetyParameter Name="Store Code">0</SafetyParameter>
    <SafetyParameter Name="Project CRC">0</SafetyParameter>
  </SafetyParameters>
  <TxPdo BitSize="48">
    <Element Id="g1_a2_o1"> <Name>OutputChannel1</Name>
      <Type>BIT</Type>
      <BitOffset>8</BitOffset>
```

```
      <Index>28673</Index>
      <SubIndex>1</SubIndex>
    </Element>
    <Element Id="g1_a2_o2">
      <Name>OutputChannel2</Name>
      <Type>BIT</Type>
      <BitOffset>9</BitOffset>
      <Index>28673</Index>
      <SubIndex>2</SubIndex>
    </Element>
    <Element Id="g1_a2_o3">
      <Name>OutputChannel3</Name>
      <Type>BIT</Type>
      <BitOffset>10</BitOffset>
      <Index>28673</Index>
      <SubIndex>3</SubIndex>
    </Element>
    <Element Id="g1_a2_o4">
      <Name>OutputChannel4</Name>
      <Type>BIT</Type>
      <BitOffset>11</BitOffset>
      <Index>28673</Index>
      <SubIndex>4</SubIndex>
    </Element>
  </TxPdo>
</SafetyAliasDevice>
```

8.1.1.1.1.2.3 StandardAliasDevice Id="g1_a3"

You link this StandardAliasDevice to the "ErrAck" group input. You need the AliasName for this. After importing, the AliasName represents the file name of the AliasDevice.

Also adjust the process image. Currently, all StandardAliasDevices are designed so that there can only be one Pdo element. As you need a Boolean input in this example, define this StandardAliasDevice similarly to the SafetyAliasDevice.

The configuration for this StandardAliasDevice is as follows:

```
<StandardAliasDevice Id="g1_a3">
 <AliasName>ErrAck</AliasName>
  <RxPdo BitSize="1">
    <Element Id="g1_a3_i1">
      <Name>In</Name>
      <Type>BIT</Type>
      <BitOffset>0</BitOffset>
    </Element>
  </RxPdo>
</StandardAliasDevice>
```

8.1.1.1.1.2.4 StandardAliasDevice Id="g1_a4"

You link this StandardAliasDevice to the "RunStop" group input. Proceed as for the StandardAliasDevice "g1_a3".

```
<StandardAliasDevice Id="g1_a4">
 <AliasName>Run</AliasName>
  <RxPdo BitSize="1">
    <Element Id="g1_a4_i1">
      <Name>In</Name>
      <Type>BIT</Type>
      <BitOffset>0</BitOffset>
    </Element>
  </RxPdo>
</StandardAliasDevice>
```

8.1.1.1.1.2.5 StandardAliasDevice Id="g1_a5"

Link this StandardAliasDevice to the 1st input of the EL1004. Proceed as for the StandardAliasDevice "g1_a3". Link this signal to the "Restart" input of the "FBEstop" function block.

```
<StandardAliasDevice Id="g1_a5">
 <AliasName>Restart</AliasName>
  <RxPdo BitSize="1">
    <Element Id="g1_a5_i1">
```

```
      <Name>In</Name>
      <Type>BIT</Type>
      <BitOffset>0</BitOffset>
    </Element>
  </RxPdo>
</StandardAliasDevice>
```

8.1.1.1.1.2.6 StandardAliasDevice Id="g1_a6"

Link this StandardAliasDevice to the 2nd input of the EL1004. This input is wired to the feedback signal of contactors K1 and K2. Proceed as for the StandardAliasDevice "g1_a3".

```
<StandardAliasDevice Id="g1_a6">
 <AliasName>Feedback</AliasName>
  <RxPdo BitSize="1">
    <Element Id="g1_a6_i1">
      <Name>In</Name>
      <Type>BIT</Type>
      <BitOffset>0</BitOffset>
    </Element>
  </RxPdo>
</StandardAliasDevice>
```

8.1.1.1.1.3 Application

You configure the function blocks in this element. To do this, first define the network, as you can only configure function blocks within networks. As only 2 function blocks are required in this application, one network is sufficient.

The configuration of the Id can be found in the chapter ID rules [▶ 75]. You need the 2 function blocks "FBEstop" and "FBEdm". The description of the function blocks can be found in the function block documentation. This is document [4] in chapter References [▶ 7].

```
<Application>
  <Network Id="g1_n1" OrderId="1">
    <FunctionBlock Id="g1_n1_f1" OrderId="1">
    </FunctionBlock>
    <FunctionBlock Id="g1_n1_f2" OrderId="2">
    </FunctionBlock>
  </Network>
</Application>
```

8.1.1.1.1.3.1 FBEstop

Specify the type, instance name and the inputs and outputs for a function block. You must configure at least one input and one output for each function block. The function block type can be found in the chapter FBTypes [▶ 77]. If you require additional parameters whose values differ from the default values, enter these parameters here as well. In this sample, you only use the default values of the function block.

```
<FunctionBlock Id="g1_n1_f1" OrderId="1">
  <Type>32</Type>
  <Name>FBEstop1</Name>
</FunctionBlock>
```

8.1.1.1.1.3.1.1 OutPorts

The FBEstop has 3 outputs. Specify the Id and PortId for each of these outputs. The ID can be found in chapter ID rules [▶ 75]. The PortId can be found in the chapter PortIds [▶ 78]. For better readability, you can also specify the PortName. You also use the configuration of the ports for the Mappings element.

```
<Outports>
  <Element Id="g1_n1_f1_o1">
    <PortId>33554464</PortId>
    <PortName>Error</PortName>
  </Element>
  <Element Id="g1_n1_f1_o2">
    <PortId>33620000</PortId>
    <PortName>EStopOut</PortName>
  </Element>
  <Element Id="g1_n1_f1_o3">
    <PortId>33685536</PortId>
```

```
    <PortName>EStopDelOut</PortName>
  </Element>
</Outports>
```

8.1.1.1.1.3.1.2 InPorts

The FBEstop has 11 inputs. Configure these according to the same principle as the OutPorts. To increase the diagnostic coverage, set a discrepancy time for the "EStopIn1" and "EstopIn2" inputs. For the InPorts for which you configure a mapping later, set the "Active" element.

```
<Inports>
  <Element Id="g1_n1_f1_i1">
    <PortId>16777248</PortId>
    <PortName>Restart</PortName>
    <Active>true</Active>
  </Element>
  <Element Id="g1_n1_f1_i10">
    <PortId>16842784</PortId>
    <PortName>EDM1</PortName>
    <Active>true</Active>
  </Element>
  <Element Id="g1_n1_f1_i11">
    <PortId>16908320</PortId>
    <PortName>EDM2</PortName>
  </Element>
  <Element Id="g1_n1_f1_i2">
    <PortId>17301536</PortId>
    <PortName>EStopIn1</PortName>
    <DiscTime>200</DiscTime>
    <Active>true</Active> </Element>
  <Element Id="g1_n1_f1_i3">
    <PortId>17367072</PortId>
    <PortName>EStopIn2</PortName>
    <Active>true</Active>
  </Element>
  <Element Id="g1_n1_f1_i4">
    <PortId>17432608</PortId>
    <PortName>EStopIn3</PortName>
  </Element>
  <Element Id="g1_n1_f1_i5">
    <PortId>17498144</PortId>
    <PortName>EStopIn4</PortName>
  </Element>
  <Element Id="g1_n1_f1_i6">
    <PortId>17563680</PortId>
    <PortName>EStopIn5</PortName>
  </Element>
  <Element Id="g1_n1_f1_i7">
    <PortId>17629216</PortId>
    <PortName>EStopIn6</PortName>
  </Element>
  <Element Id="g1_n1_f1_i8">
    <PortId>17694752</PortId>
    <PortName>EStopIn7</PortName>
  </Element>
  <Element Id="g1_n1_f1_i9">
    <PortId>17760288</PortId>
    <PortName>EStopIn8</PortName>
  </Element>
</Inports>
```

8.1.1.1.1.3.2 FBEdm

Proceed in a similar way for this function block. Use the "SwitchOffTime" and "SwitchOnTime" parameters. Please refer to the chapter FunctionBlock [▶ 35].

```
<FunctionBlock Id="g1_n1_f2" OrderId="2">
  <Type>39</Type>
  <Name>FBEdm1</Name>
  <SwitchOffTime>1500</SwitchOffTime>
  <SwitchOnTime>1000</SwitchOnTime>
  <Outports>
    <Element Id="g1_n1_f2_o1">
      <PortId>33554471</PortId>
      <PortName>Error</PortName>
    </Element>
```

```
  </Outports>
  <Inports>
    <Element Id="g1_n1_f2_i1">
      <PortId>17301543</PortId>
      <PortName>Mon1</PortName>
      <Active>true</Active>
    </Element>
    <Element Id="g1_n1_f2_i2">
      <PortId>17367079</PortId>
      <PortName>Mon2</PortName>
      <Active>true</Active>
    </Element>
  </Inports>
</FunctionBlock>
```

8.1.1.1.2 Mappings

In this element, you make the mappings between the inputs and outputs of the groups, function blocks and AliasDevices. To do this, enter the SourceId and the TargetId. These are IDs that you have previously defined. If it is a mapping to a port of a TwinSAFE group or a function block, enter the variable name.

```
<Mappings>
</Mappings>
```

You have already defined the "Run" (g1_i2) and "ErrAck" (g1_i1) group inputs relevant to this sample in the element GroupInputs [▶ 92]. You map these to the inputs of StandardAliasDevice Id="g1_a3" [▶ 97] and StandardAliasDevice Id="g1_a4" [▶ 97].

```
<Mapping TargetId="g1_i1" SourceId="g1_a3_i1" LocalVarName="GroupPort_RunStop"/>
<Mapping TargetId="g1_i2" SourceId="g1_a4_i1" LocalVarName="GroupPort_ErrAck"/>
```

Map the "Restart" input of FBEstop [▶ 98] with the input of StandardAliasDevice Id="g1_a5" [▶ 97].

```
<Mapping TargetId="g1_n1_f1_i1" SourceId="g1_a5_i1" LocalVarName="Restart"/>
```

Map the "EStopIn1" input from FBEstop [▶ 98] to the 3rd input of SafetyAliasDevice Id="g1_a1" [▶ 93]. Map the "EstopIn2" input from FBEstop [▶ 98] to the 4th input of SafetyAliasDevice Id="g1_a1" [▶ 93].

```
<Mapping TargetId="g1_n1_f1_i2" SourceId="g1_a1_i3" LocalVarName="S1_1"/>
<Mapping TargetId="g1_n1_f1_i3" SourceId="g1_a1_i4" LocalVarName="S1_2"/>
```

Connect the feedback signal to the "EDM1" input of the FBEstop [▶ 98] and the "Mon2" input of the FBEdm [▶ 99]. Map this signal StandardAliasDevice Id="g1_a6" [▶ 98].

```
<Mapping TargetId="g1_n1_f1_i10" SourceId="g1_a6_i1" LocalVarName="Feedback"/>
<Mapping TargetId="g1_n1_f2_i2" SourceId="g1_a6_i1" LocalVarName="Feedback"/>
```

Map the "EStopOut" output of the FBEstop [▶ 98] to the 1st input of the SafetyAliasDevice Id="g1_a2" [▶ 96].

```
<Mapping TargetId="g1_a2_o1" SourceId="g1_n1_f1_o2" LocalVarName="Output"/>
```

Define a line mapping between the "EStopOut" output of the FBEstop [▶ 98] and the "Mon1" input of the FBEdm [▶ 99].

```
<Mapping TargetId="g1_n1_f2_i1" SourceId="g1_n1_f1_o2"/>
```

8.2 XSD validation

After you have created the XML file, validate this XML with the XSD file with the same version number.

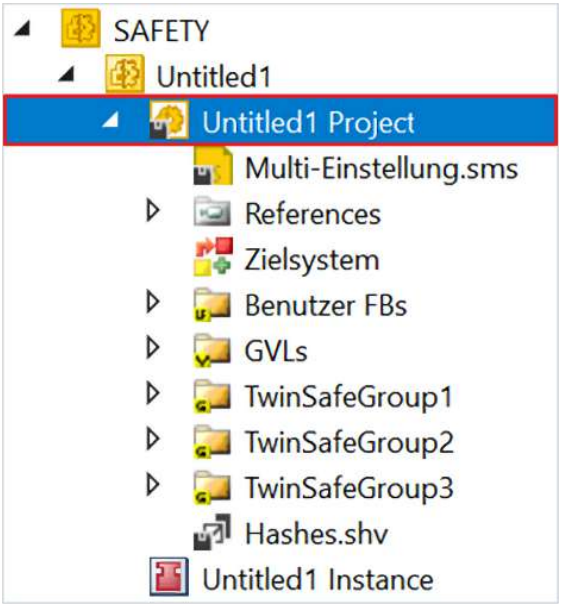
After successful validation, you can carry out the import.

9 Test Automation Interface

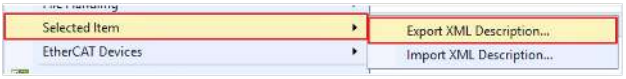
9.1 Manual export

You have the option of manually testing the export function of the Automation Interface.

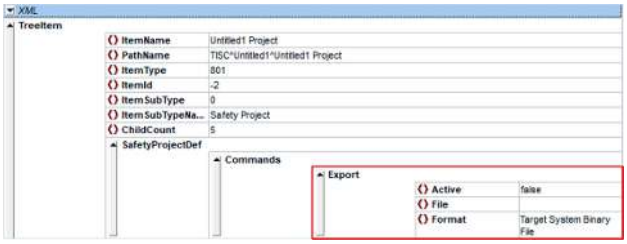
Proceed as follows:



1. Select safety project



2. Click on "TwinCAT" in the menu bar
3. Click on "Export XML description..." via the "Select element" button



4. Open XML file

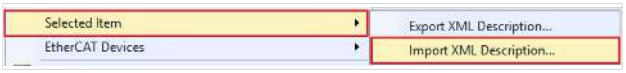
Under the "SafetyProjectDef" element, you will find all the commands that the TwinSAFE Editor provides for the Automation Interface. You need the "Export" command for the export function.

To activate the export function, you must next set 3 values:

5. Set element "Active" to "true"
6. Enter a file name in the "File" element
7. Select one of the following formats in the "Format" element:

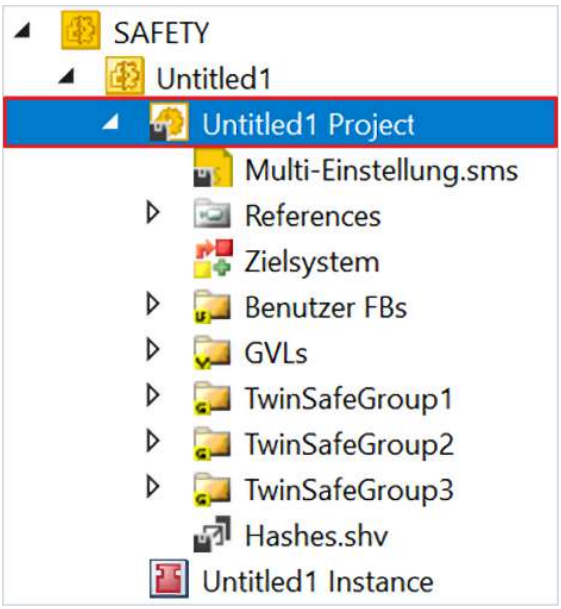
- TwinCAT 3 Xml Export Format
- TwinCAT 2 Xml Export Format
- Target System Binary File

8. Save changes in the file

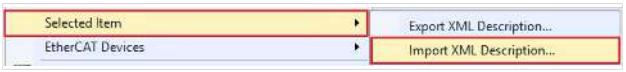


9. Select safety project
10. Click on "TwinCAT" in the menu bar
11. Click on "Insert XML description..." via the "Select element" button

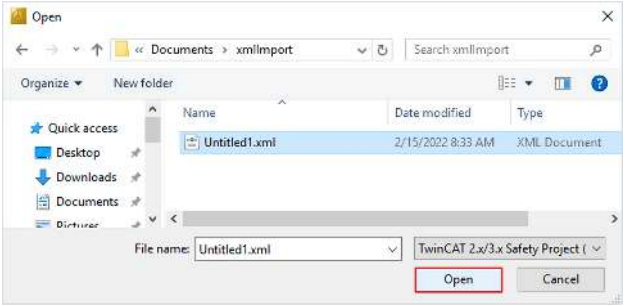
9.2 Manual import



1. Select safety project



2. Click on "TwinCAT" in the menu bar
3. Click on "Insert XML description..." via the "Select element" button



4. Select storage location
5. Select modified XML file

The modification of the XML file can be found in the chapter Manual export [▶ 101].

6. Confirm with “Open”

10 FMEDA

he following table contains the FMEDA for XML export and XML import. The *Error mode* column contains the error description, the *Effect* column describes the effect and the *Diagnosis* column describes how the error is detected or not detected.

⚠ CAUTION

FMEDA

The last column *Required measures by the user* of the following table indicates whether the user must take measures to rectify the error described under *Error mode* in a safe manner. These measures must be defined and implemented by the user in the form of process descriptions or software specifications, for example. The user shall bear full responsibility for demonstrating the accuracy and efficacy of these measures.

The XSD validation mentioned in this chapter is not performed automatically via the TwinCAT 3 Safety Editor. The user is responsible for validation.

FMEDA ID	Error mode	Effect	Diagnosis	Required measures by the user 
1	The TwinCAT 3 safety project contains errors.	The safety project cannot be exported, either manually or via the Automation Interface. The "Export XML" button is disabled in TwinCAT 3.	The errors can be displayed via the safety project verification in manual operation of TwinCAT 3.	Eliminate the specified errors from the safety project verification.
2	The safety project in XML format is not valid with regard to the XSD.	As long as the data is correct, the project is imported. Otherwise, the project is rejected.	Import errors are displayed in a dialog during manual operation of TwinCAT 3. Errors are not displayed when using the Automation Interface.	Before the xml import, make sure that the safety project to be imported is valid against the corresponding XSD. Eliminate the specified errors in the dialog.
3	The XSD validation of the XML safety project is successful, but there are typing errors in the PortIds, types or other elements, for example.	Elements can no longer be assigned correctly when importing. The import is canceled.	Import errors are displayed in a dialog during manual operation of TwinCAT 3. Errors are not displayed when using the Automation Interface.	Eliminate the specified errors in the dialog.
4	The XSD validation of the XML safety project is successful, but links are missing, for example, or safe addresses have been assigned twice.	The project is imported, but it is invalid in TE9000.	The errors can be displayed via the safety project verification in manual operation of TwinCAT 3.	Eliminate the specified errors from the safety project verification or, if necessary, the errors in the automatic generation of the XML.
5	The XSD validation of the XML safety project is successful, but paths are entered in the "IoPath" elements that are already used in the	The safety project is imported. The project is valid if there are no other errors, but the TwinCAT configuration cannot be activated.	In manual TwinCAT operation, a dialog indicates why the TwinCAT project cannot be activated.	Eliminate the specified errors from the TwinCAT dialog or, if necessary, the errors in the automatic generation of the XML.

FMEDA ID	Error mode	Effect	Diagnosis	Required measures by the user 
	TwinCAT project. For example, an input variable can be linked to several output variables.			
6	The XSD validation of the XML safety project is successful, but an IoPath is entered in the TargetSystemConfiguration that is already being used for an existing safety project in a TwinCAT project into which the XML is to be imported.	The safety project is imported. The project is valid (if there are no other errors), but the process image of the pre-existing project is overwritten by the imported safety project.	This case can be detected by testing.	Select a different TargetSystemConfiguration - IoPath, or check the IoPath before importing, or adjust the automatic generation of the XML.

No action is performed on a safety-critical device with the XML editor of choice! It is a purely offline function.

List of figures

More Information:
www.beckhoff.com/TE9000

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