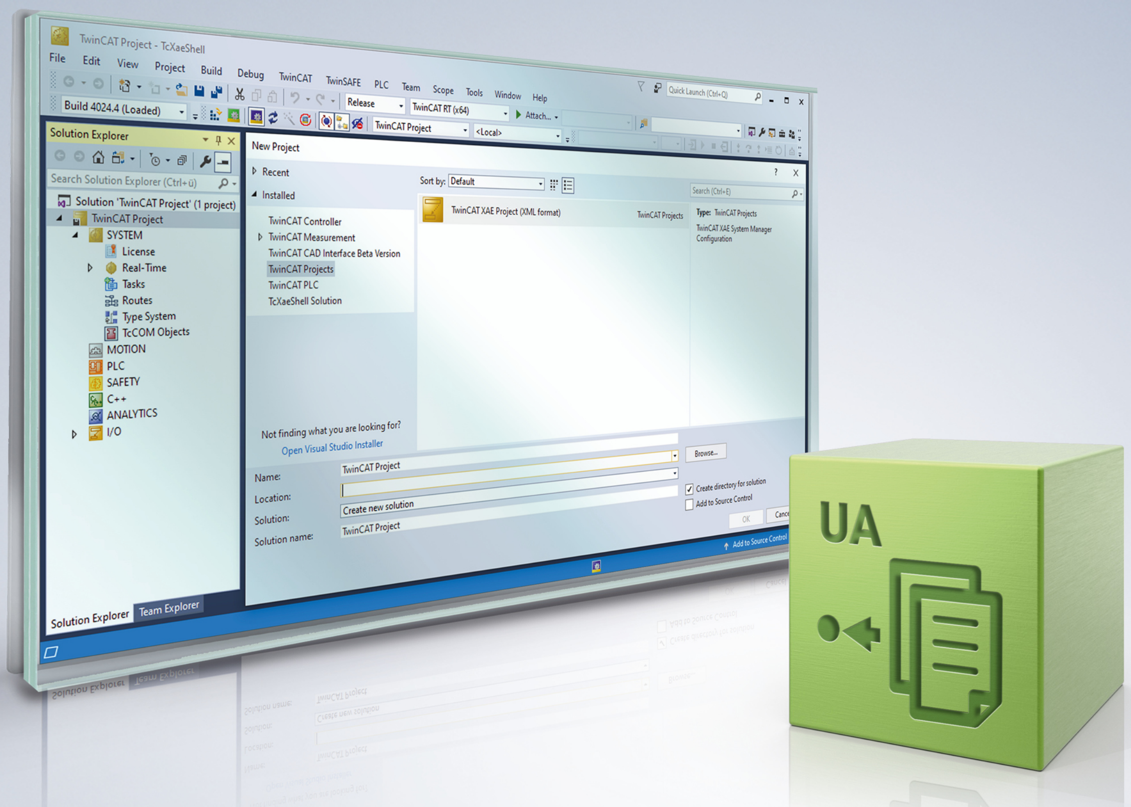
BECKHOFF New Automation Technology

Manual | EN

TE6100

TwinCAT 3 OPC UA Nodeset Editor



2024-08-14 | Version: 1.0.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

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

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ 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
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.

1.3 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.

1.4 Documentation output status

Version	Change
1.0.x	First publication

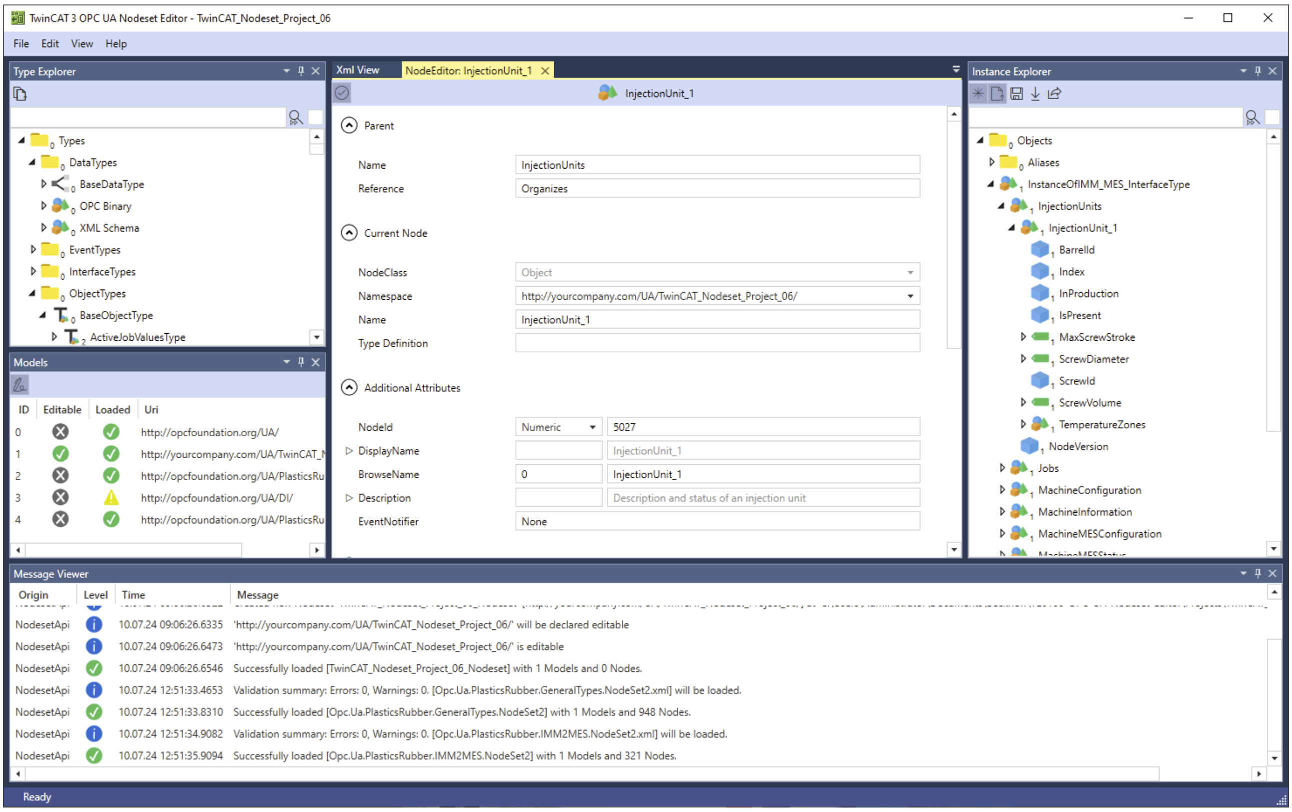
2 Overview

The TwinCAT OPC UA Nodeset Editor is an engineering tool for creating and editing OPC UA nodeset files, which are used in particular for companion specifications. With the help of the editor, you can map existing companion specifications or your own information models on your Beckhoff controller and link them with data points from the PLC.

When developing a machine application, there are often requirements for the structure of the OPC UA namespace of a machine to be mapped. In this case, it is necessary to establish conformity to a companion specification or a customer-specific information model so that the machine can be integrated as smoothly as possible into an existing and standardized environment.

The TwinCAT OPC UA Nodeset Editor is the graphical solution to meet this challenge. It allows:

- Loading nodeset files from companion specifications or custom information models.
- Creation of simple user-defined nodeset files.
- Convenient linking of objects from the controller with those from a loaded or created nodeset.
- Using the nodeset for new and existing machines using the TwinCAT OPC UA Server (TF6100).



Parent

Name	InjectionUnits
Reference	Organizes

Current Node

NodeClass	Object
Namespace	http://yourcompany.com/UA/TwinCAT_Nodeset_Project_06/
Name	InjectionUnit_1
Type Definition	

Additional Attributes

NodeId	Numeric	5027
DisplayName		InjectionUnit_1
BrowseName	0	InjectionUnit_1
Description		Description and status of an injection unit
EventNotifier	None	

ID	Editable	Loaded	Uri
0	☒	✓	http://opcfoundation.org/UA/
1	✓	✓	http://yourcompany.com/UA/TwinCAT_N
2	☒	✓	http://opcfoundation.org/UA/PlasticsRu
3	☒	⚠	http://opcfoundation.org/UA/DI/
4	☒	✓	http://opcfoundation.org/UA/PlasticsRu

```
Origin       Level  Time                     Message
NodesetApi   i      10.07.24 09:06:26.6335   'http://yourcompany.com/UA/TwinCAT_Nodeset_Project_06/' will be declared editable
NodesetApi   i      10.07.24 09:06:26.6473   'http://yourcompany.com/UA/TwinCAT_Nodeset_Project_06/' is editable
NodesetApi   ✓      10.07.24 09:06:26.6546   Successfully loaded [TwinCAT_Nodeset_Project_06_Nodeset] with 1 Models and 0 Nodes.
NodesetApi   i      10.07.24 12:51:33.4653   Validation summary: Errors: 0, Warnings: 0. [Opc.Ua.PlasticsRubber.GeneralTypes.NodeSet2.xml] will be loaded.
NodesetApi   ✓      10.07.24 12:51:33.8310   Successfully loaded [Opc.Ua.PlasticsRubber.GeneralTypes.NodeSet2] with 1 Models and 948 Nodes.
NodesetApi   i      10.07.24 12:51:34.9082   Validation summary: Errors: 0, Warnings: 0. [Opc.Ua.PlasticsRubber.IMM2MES.NodeSet2.xml] will be loaded.
NodesetApi   ✓      10.07.24 12:51:35.9094   Successfully loaded [Opc.Ua.PlasticsRubber.IMM2MES.NodeSet2] with 1 Models and 321 Nodes.
```

Use cases

The TwinCAT OPC UA Nodeset Editor supports the user in the implementation of requirements regarding companion specifications by providing a graphical user interface that enables the import of nodesets. However, even without requirements for a specific companion specification, the TwinCAT OPC UA Nodeset Editor may be a useful tool and assist in the following use cases.

Creation of custom server address space

Even before companion specifications became widespread, there were initial requests to make information from the PLC available on the TwinCAT OPC UA Server according to a defined structure. By default, the server uses the IEC61131-3 information model to provide variables from the PLC via OPC UA. However, sometimes this data representation on the server is not desired and there were requests for a "simple modeling option for folders and simple data types". This use case can be addressed with the help of this product.

Linking variables from existing nodesets

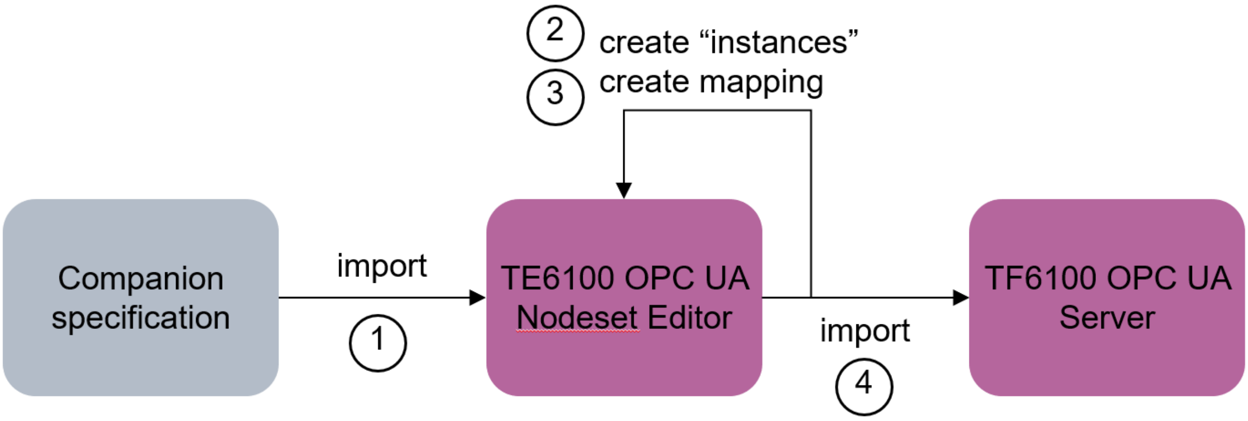
Nodesets that have been created with a third-party tool, such as the UA Modeler (https://www.unified-automation.com), can be imported into the TwinCAT OPC UA Server. However, since third-party tools do not have any variable information from a TwinCAT control program, the TwinCAT OPC UA Nodeset Editor can be used instead. The existing nodesets can be imported into the graphical user interface of the Nodeset Editor and linked there with PLC variables using the TwinCAT Target Browser. The resulting nodeset file can then be deployed to the TwinCAT OPC UA Server.

Creating instance nodesets from a companion specification

Importing companion specifications and creating instances from the data and object types within is required if a machine has to be equipped with a standardized interface that has been defined by the companion specification. The TwinCAT OPC UA Nodeset Editor enables the instantiation of such data and object types and the subsequent linking of the instances with variables from the TwinCAT PLC.

Workflow

The workflow when using the TwinCAT OPC UA Nodeset Editor to create a nodeset and the subsequent export to the TwinCAT OPC UA Server can be illustrated as follows.



In step (1) the nodeset file (the "companion specification") is imported into the TwinCAT OPC UA Nodeset Editor. This means that the data types defined in the file are available in the editor environment and can be used for project planning.

In step (2) and (3) the project engineering takes place, i.e. objects ("instances") are created from the data types and linked with variables from the PLC.

In step (4) the generated project is imported into the TwinCAT OPC UA Server so that it can provide the generated information model in its address space.

Also see about this

- Supported features [▸ 23]

3 Installation

3.1 System requirements

The following table describes the technical system requirements for the pure operation of the application.

Technical data	Description
Operating system	Windows 10, 11
Target platform	PC architecture (x86, x64)
Minimum TwinCAT version	TwinCAT 3.1 Build 4026
Required TwinCAT setup level	TwinCAT 3 XAE, XAR, ADS
Required TwinCAT license	TE6100 TC3 OPC UA Nodeset Editor L1 TE6101 TC3 OPC UA Nodeset Editor L2 (optional add-on license)

ℹ **TF6100 TC3 OPC UA Server**

This product works together with the TwinCAT OPC UA Server to run a created nodeset and display it on the server’s address space. Please make sure that the product TF6100 TC3 OPC UA Server is installed on your system.

3.2 Installation

If you are using TwinCAT 3.1 Build 4026 (and higher) on the Microsoft Windows operating system, you can install this function via the TwinCAT Package Manager, see Installation documentation. Normally you install the function via the corresponding workload. However, you can also install the packages contained in the workload individually. This documentation briefly describes the installation process via the workload.

Command line program TcPkg

You can use the TcPkg Command Line Interface (CLI) to display the available workloads on the system:

```
tcpkg list -t workload
```

You can use the following command to install the workload of this function.

```
tcpkg install TE6100.OpcUaNodesetEditor.XAE
```

TwinCAT Package Manager UI

You can use the User Interface (UI) to display all available workloads and install them if required.

To do this, follow the corresponding instructions in the interface.

Installation of the corresponding package or workload can be either done via TcPkg CLI or TcPkg UI.

ℹ **Unprepared TwinCAT restart can cause data loss**

The installation of this function may result in a TwinCAT restart.

Make sure that no critical TwinCAT applications are running on the system or shut them down in an orderly manner first.

3.3 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

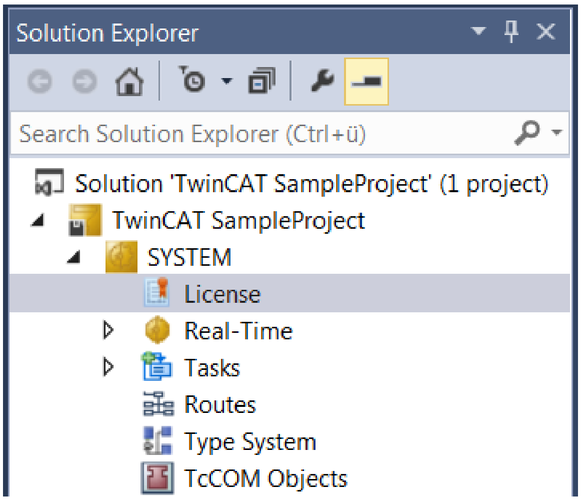
Licensing the full version of a TwinCAT 3 Function

A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

Licensing the 7-day test version of a TwinCAT 3 Function

A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

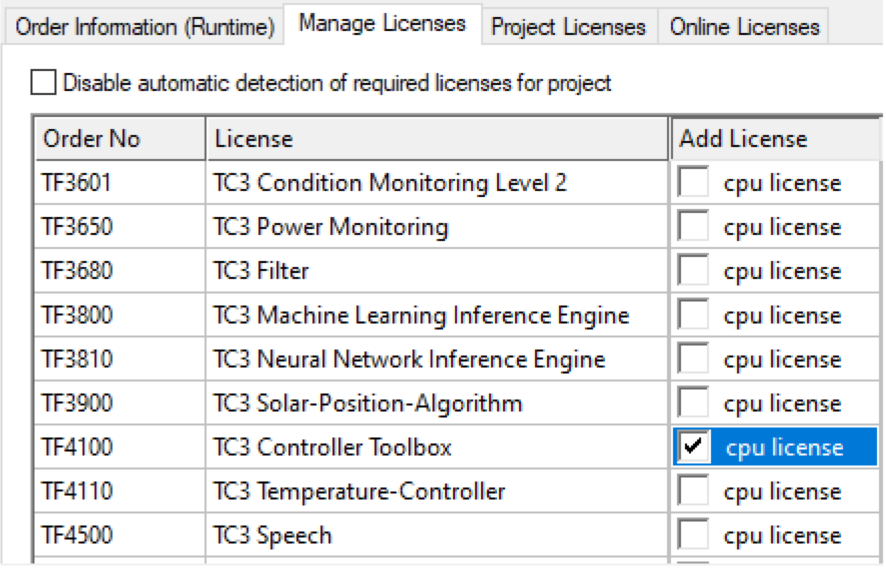
1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
   ⇨ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.



Solution 'TwinCAT SampleProject' (1 project)
- TwinCAT SampleProject
  - SYSTEM
    - License
    - Real-Time
    - Tasks
    - Routes
    - Type System
    - TcCOM Objects

   ⇨ The TwinCAT 3 license manager opens.

5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").

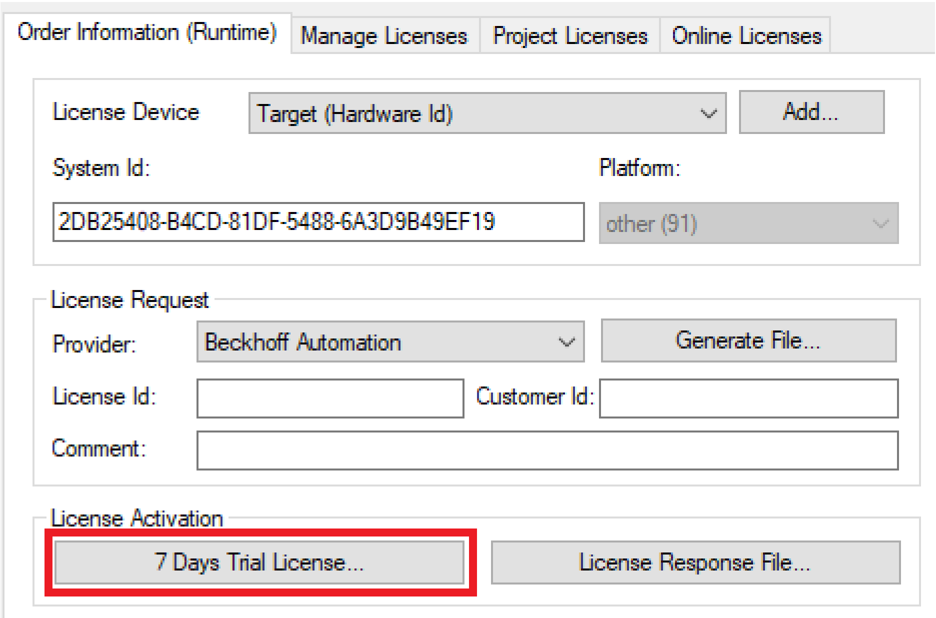


□ Disable automatic detection of required licenses for project

Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	□ cpu license
TF3650	TC3 Power Monitoring	□ cpu license
TF3680	TC3 Filter	□ cpu license
TF3800	TC3 Machine Learning Inference Engine	□ cpu license
TF3810	TC3 Neural Network Inference Engine	□ cpu license
TF3900	TC3 Solar-Position-Algorithm	□ cpu license
TF4100	TC3 Controller Toolbox	☑ cpu license
TF4110	TC3 Temperature-Controller	□ cpu license
TF4500	TC3 Speech	□ cpu license

6. Open the **Order Information (Runtime)** tab.
   ⇨ In the tabular overview of licenses, the previously selected license is displayed with the status “missing”**.**

7. Click **7-Day Trial License...** to activate the 7-day trial license.



License Device: Target (Hardware Id)

System Id: 2DB25408-B4CD-81DF-5488-6A3D9B49EF19

Platform: other (91)

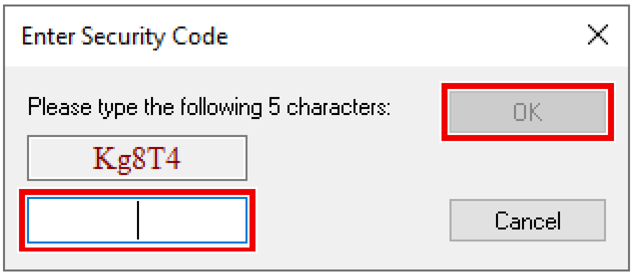
Provider: Beckhoff Automation

License Id:

Customer Id:

Comment:

⇨ A dialog box opens, prompting you to enter the security code displayed in the dialog.



Enter Security Code

Please type the following 5 characters:

Kg8T4

8. Enter the code exactly as it is displayed and confirm the entry.
9. Confirm the subsequent dialog, which indicates the successful activation.
   ⇨ In the tabular overview of licenses, the license status now indicates the expiry date of the license.
10. Restart the TwinCAT system.

⇨ The 7-day trial version is enabled.

4 Technical introduction

4.1 Quick Start

The following chapter will guide you through some quick start tutorials on how to use the TwinCAT OPC UA Nodeset Editor. Please make sure that you have followed the installation instructions and that your system meets the system requirements. The following quick start tutorials are available:

Tutorial	Description
Simple nodeset with base types [▶ 13]	Demonstrates how to create a nodeset with simple data type instances.
Simple companion specification [▶ 19]	Demonstrates how to create a nodeset with object instances from a companion specification.

4.1.1 Simple nodeset with base types

This documentation article is intended to get you started using this product quickly. After successful installation [▶ 10] and licensing [▶ 10], perform the following steps to create an initial nodeset.

OPC UA Base Nodeset

By default, the TwinCAT OPC UA Nodeset Editor automatically loads the OPC UA Base Nodeset, which defines the namespace http://opcfoundation.org/UA/.

TF6100 TC3 OPC UA Server

This product works together with the TwinCAT OPC UA Server to run a created nodeset and display it on the server’s address space. Please make sure that the product TF6100 TC3 OPC UA Server is installed on your system.

The following Quick Start Tutorial will guide you through the following steps:

- Preparing a PLC project
- Creating a new project
- Creating instances from the OPC UA Base Nodeset
- Linking the created instances with PLC variables
- Deploying the project to TwinCAT OPC UA Server

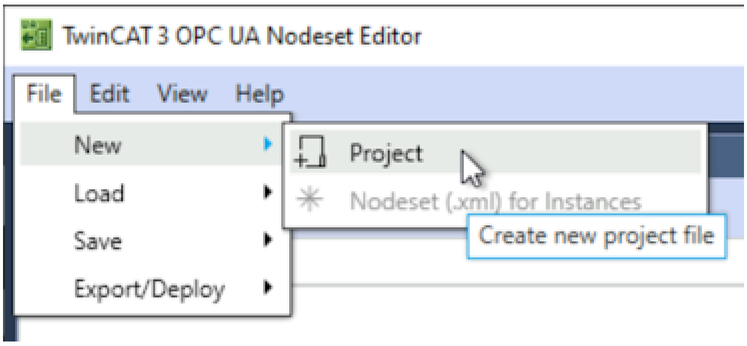
Preparing a PLC project

Create a new TwinCAT project and add a PLC project. Create the following data structure in the PLC project and instantiate it three times in the MAIN program.

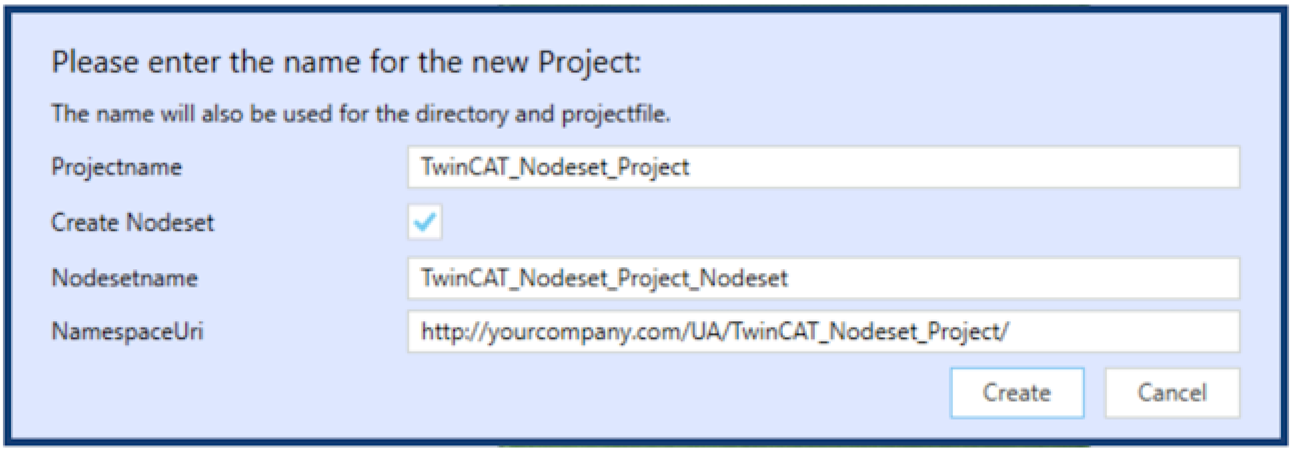
```
TYPE ST_MyStructure :
STRUCT
  MyBoolean : BOOL;
  MyInt16 : INT;
  MyFloat : REAL;
END_STRUCT
END_TYPE

PROGRAM MAIN
VAR
  MyStruct1 : ST_MyStructure;
  MyStruct2 : ST_MyStructure;
  MyStruct3 : ST_MyStructure;
END_VAR
```

Creating a new project

To create a new nodeset project, please open the menu **File** and click on **New \ Project**.



In the selection dialog that follows, you can specify the name and a target directory for the new project.


Please enter the name for the new Project:

The name will also be used for the directory and projectfile.

Projectname: TwinCAT_Nodeset_Project

Create Nodeset: ☑

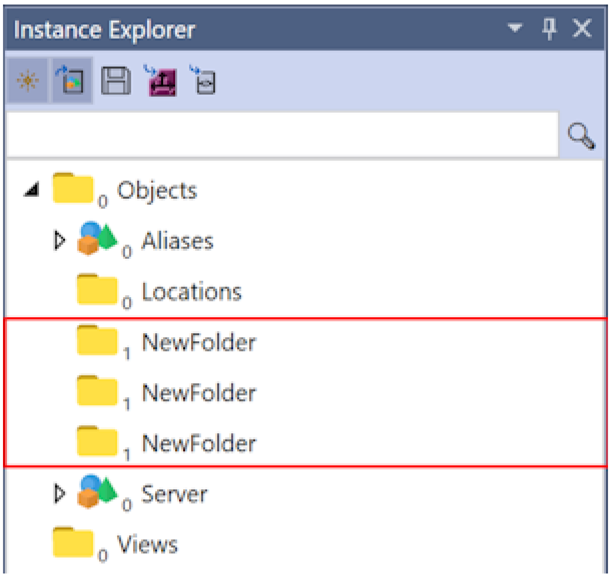
Nodesetname: TwinCAT_Nodeset_Project_Nodeset

NamespaceUri: http://yourcompany.com/UA/TwinCAT_Nodeset_Project/

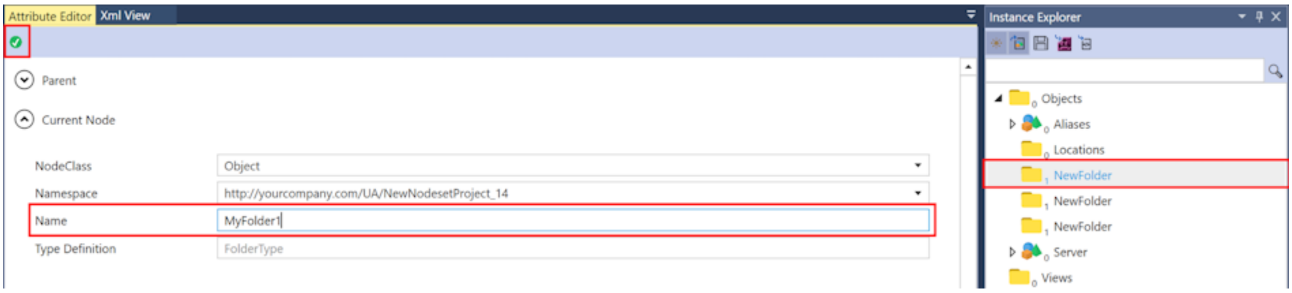
Creating Instances

In this tutorial, we want to create a number of instances with simple data types (Integer, Boolean, Float) and group them in three different folders.

Go to the Instance Explorer and add three folders by selecting the **Objects** node and choosing **Add Folder** from the context menu. The three folders will then be shown as follows:



Rename the folders by double-clicking on each folder and adjusting the **Name** attribute in the Attribute Explorer accordingly. Save this change via the **Save** button on the toolbar.



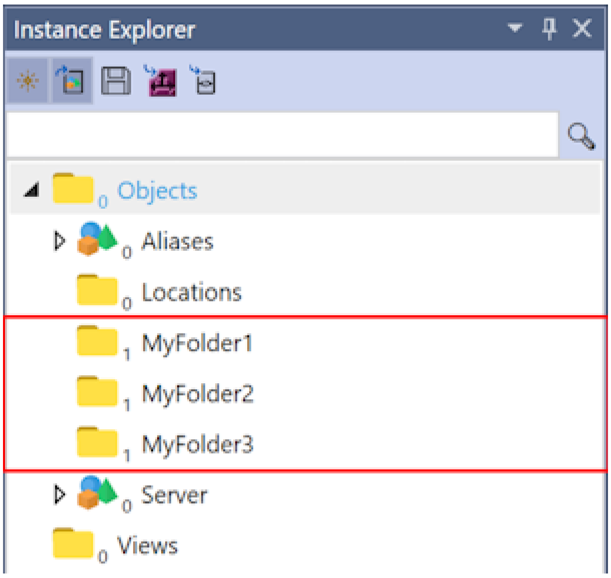
NodeClass: Object

Namespace: http://yourcompany.com/UA/NewNodesetProject_14

Name: MyFolder1

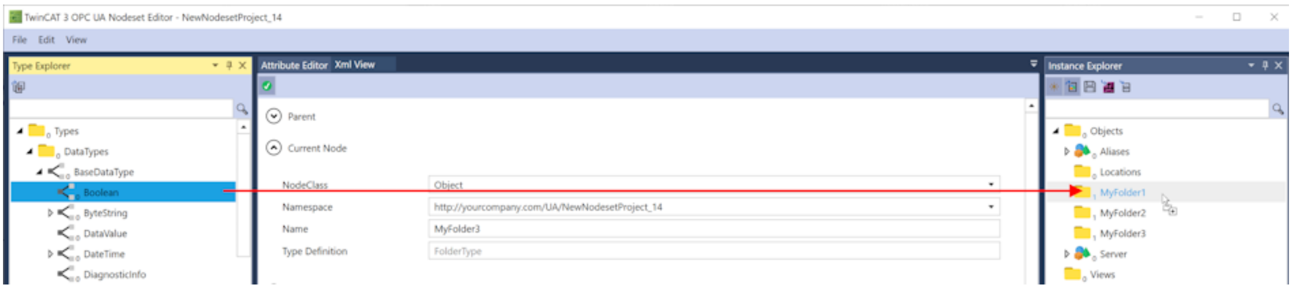
Type Definition: FolderType

Perform this procedure for all three folders so that the Instance Explorer looks like this:



In the next step we want to add some variables to the folders.

Creating new variable or object instances is usually done by a drag and drop operation from a data type on the Type Explorer. Select the first folder in the Instance Explorer and then navigate to the desired data type in the Type Explorer. Drag and drop the data type onto the selected folder in the Instance Explorer. As a result, a corresponding instance of this data type is created there. In our example, we use the Boolean data type.



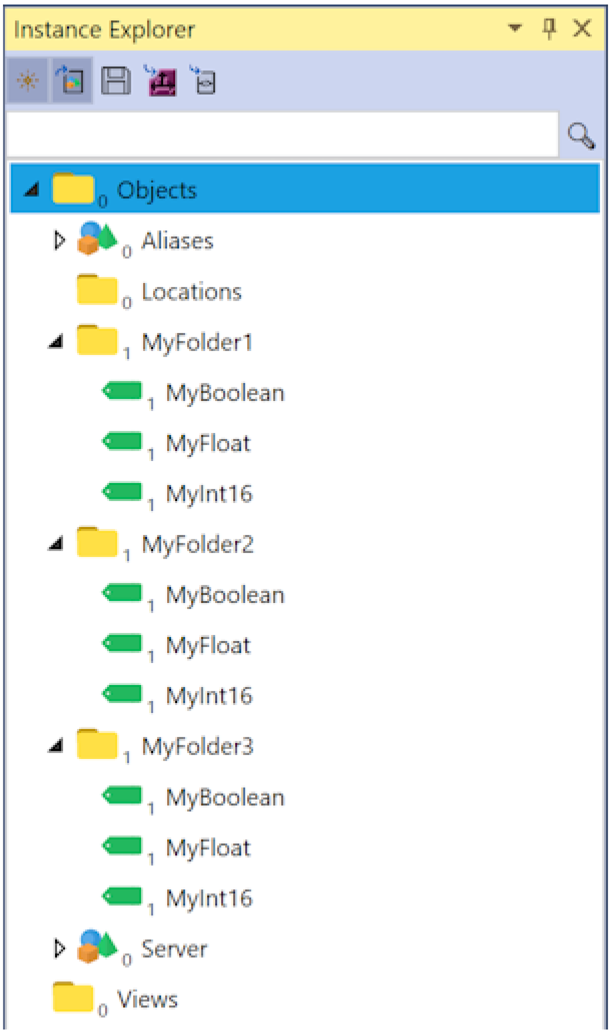
NodeClass: Object

Namespace: http://yourcompany.com/UA/NewNodesetProject_14

Name: MyFolder3

Type Definition: FolderType

We now repeat this step with two more data types, e.g. Int16 and Float, and perform this process for the other two folders as well. If necessary, you can adjust the name of the added variable by double-clicking on it and then renaming it in the Attribute Explorer.

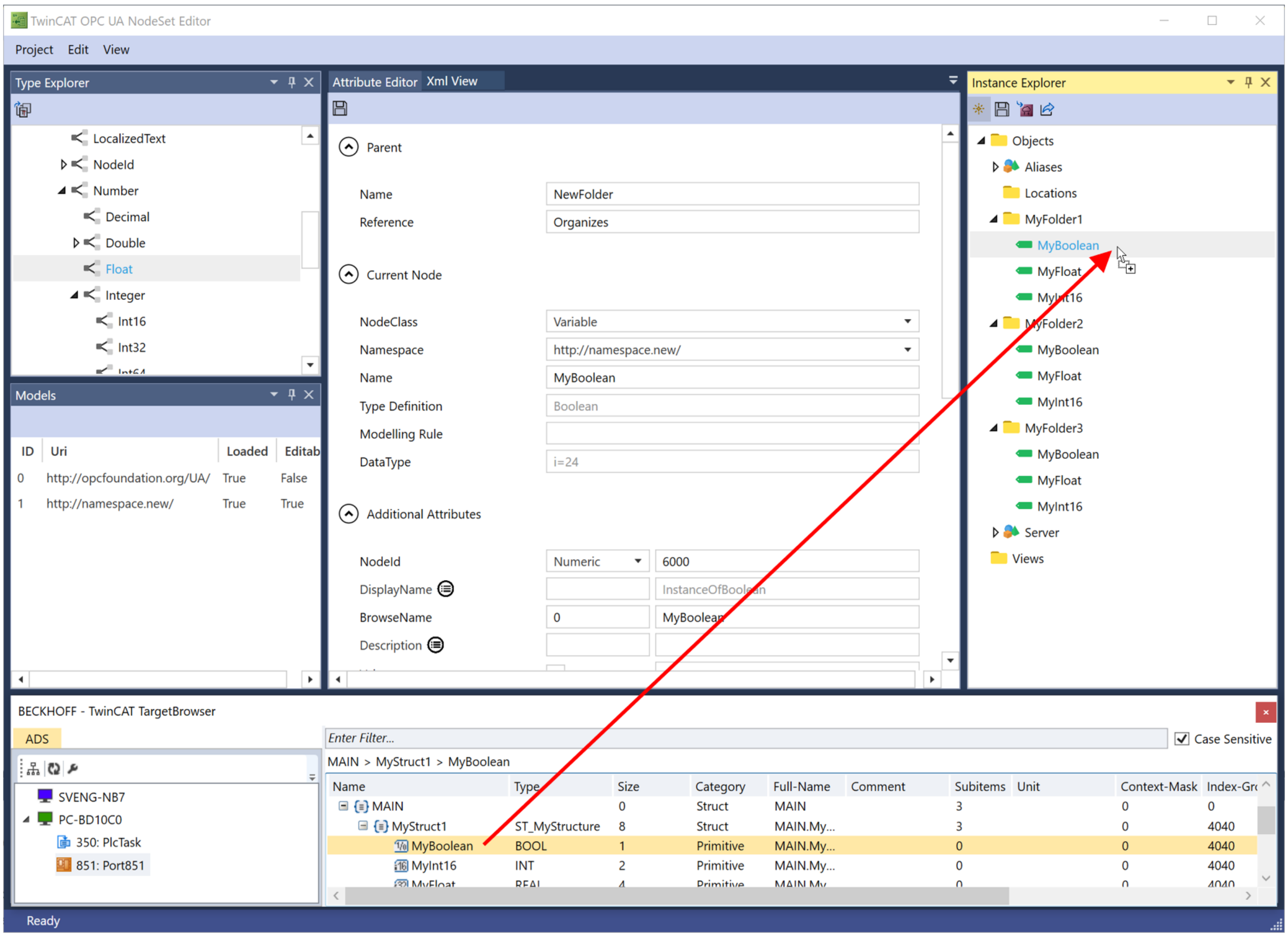


Linking with PLC variables

We want to link the now created instances with variables from our prepared PLC project. For this purpose, the TwinCAT Target Browser can be used, which you can open in the menu **View**.

Navigate to the prepared PLC project in the Target Browser and link the individual member variables of the ST_MyStructure data structure with the corresponding instances by dragging and dropping them onto the corresponding node in the Instance Explorer.

The link is then visually visible in the tree and can also be checked in the Attribute Explorer.



Parent

- Name: NewFolder
- Reference: Organizes

Current Node

- NodeClass: Variable
- Namespace: http://namespace.new/
- Name: MyBoolean
- Type Definition: Boolean
- Modelling Rule:
- DataType: i=24

Additional Attributes

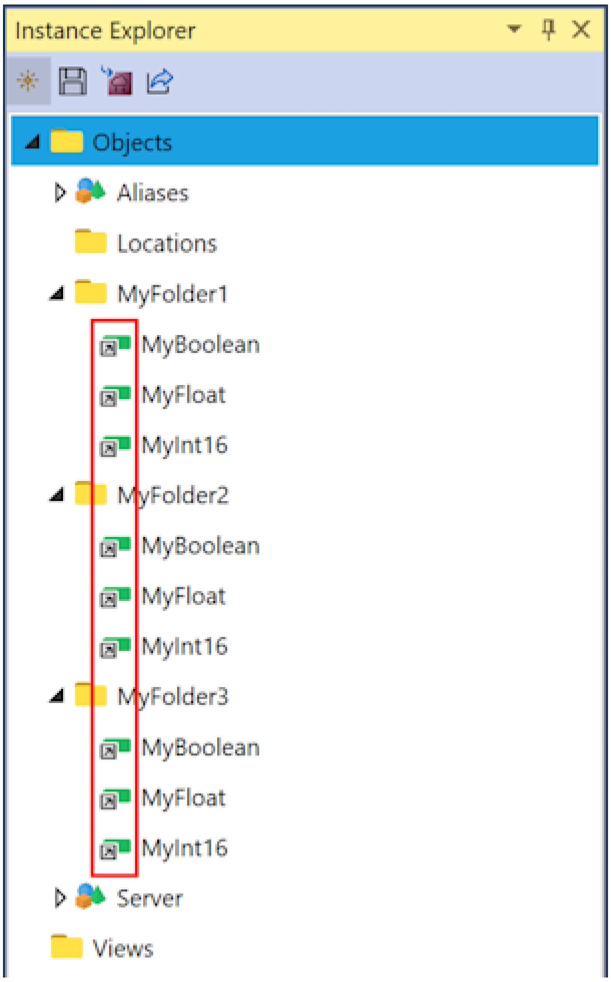
- NodeId: Numeric | 6000
- DisplayName: | InstanceOfBoolean
- BrowseName: 0 | MyBoolean
- Description:

ID	Uri	Loaded	Editab
0	http://opcfoundation.org/UA/	True	False
1	http://namespace.new/	True	True

MAIN > MyStruct1 > MyBoolean

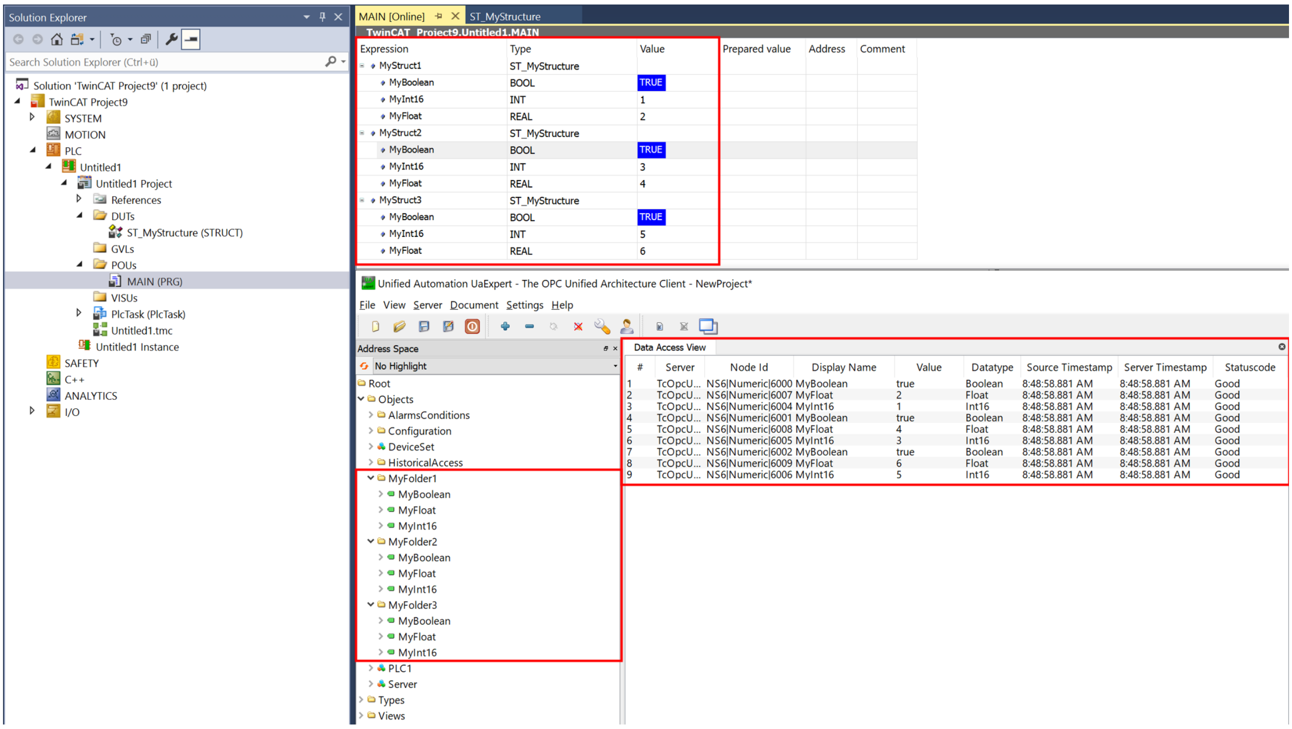
Name	Type	Size	Category	Full-Name	Comment	Subitems	Unit	Context-Mask	Index-Gr
MAIN		0	Struct	MAIN		3		0	0
MyStruct1	ST_MyStructure	8	Struct	MAIN.My...		3		0	4040
MyBoolean	BOOL	1	Primitive	MAIN.My...		0		0	4040
MyInt16	INT	2	Primitive	MAIN.My...		0		0	4040
MyFloat	REAL	4	Primitive	MAIN.My...		0		0	4040

Perform this procedure for all variables.



Deploying the project to TwinCAT OPC UA Server

To deploy the project to the TwinCAT OPC UA Server, click on the **Export to TwinCAT OPC UA Server** button in the Instance Explorer toolbar. This exports the project nodeset to the locally installed TwinCAT OPC UA Server and also creates a corresponding configuration file for the server. Please make sure that you restart the TwinCAT OPC UA Server after you have deployed the nodeset so that the configuration is loaded. The address space of the server then contains the created nodes, which were linked with the corresponding PLC variables.



TwinCAT_Project9.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address	Comment
MyStruct1	ST_MyStructure				
MyBoolean	BOOL	TRUE			
MyInt16	INT	1			
MyFloat	REAL	2			
MyStruct2	ST_MyStructure				
MyBoolean	BOOL	TRUE			
MyInt16	INT	3			
MyFloat	REAL	4			
MyStruct3	ST_MyStructure				
MyBoolean	BOOL	TRUE			
MyInt16	INT	5			
MyFloat	REAL	6			

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	TcOpcU...	NS6|Numeric|6000	MyBoolean	true	Boolean	8:48:58.881 AM	8:48:58.881 AM	Good
2	TcOpcU...	NS6|Numeric|6007	MyFloat	2	Float	8:48:58.881 AM	8:48:58.881 AM	Good
3	TcOpcU...	NS6|Numeric|6004	MyInt16	1	Int16	8:48:58.881 AM	8:48:58.881 AM	Good
4	TcOpcU...	NS6|Numeric|6001	MyBoolean	true	Boolean	8:48:58.881 AM	8:48:58.881 AM	Good
5	TcOpcU...	NS6|Numeric|6008	MyFloat	4	Float	8:48:58.881 AM	8:48:58.881 AM	Good
6	TcOpcU...	NS6|Numeric|6005	MyInt16	3	Int16	8:48:58.881 AM	8:48:58.881 AM	Good
7	TcOpcU...	NS6|Numeric|6002	MyBoolean	true	Boolean	8:48:58.881 AM	8:48:58.881 AM	Good
8	TcOpcU...	NS6|Numeric|6009	MyFloat	6	Float	8:48:58.881 AM	8:48:58.881 AM	Good
9	TcOpcU...	NS6|Numeric|6006	MyInt16	5	Int16	8:48:58.881 AM	8:48:58.881 AM	Good

4.1.2 Simple companion specification

This documentation article is intended to get you started using this product quickly. After successful installation [▶ 10] and licensing [▶ 10], perform the following steps to create an initial nodeset based on a companion specification.

OPC UA Base Nodeset

By default, the TwinCAT OPC UA Nodeset Editor automatically loads the OPC UA Base Nodeset, which defines the namespace http://opcfoundation.org/UA/.

TF6100 TC3 OPC UA Server

This product works together with the TwinCAT OPC UA Server to run a created nodeset and display it on the server’s address space. Please make sure that the product TF6100 TC3 OPC UA Server is installed on your system.

Companion specification sample nodeset

As an example of a companion specification, a nodeset is used that defines an ObjectType "MyObjectType" with a few member variables. This nodeset can be downloaded from our TE6100_Samples repository on GitHub [▶ 36].

The following Quick Start tutorial will guide you through the following steps:

- Preparing a PLC project
- Creating a new project
- Importing the companion specification nodeset
- Creating object instances
- Linking the created instances with PLC variables
- Deploying the project to TwinCAT OPC UA Server

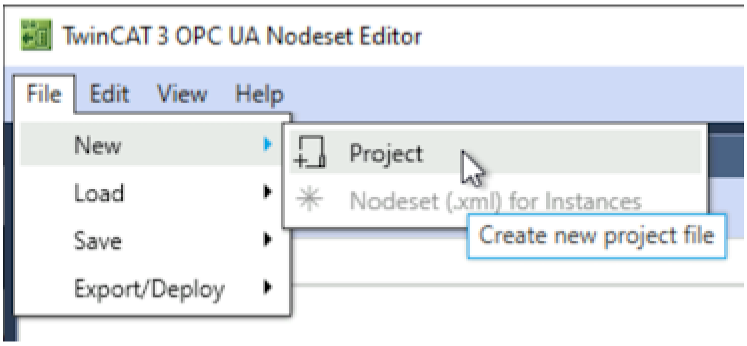
Preparing a PLC project

Create a new TwinCAT project and add a PLC project. Create the following data structure in the PLC project and instantiate it three times in the MAIN program.

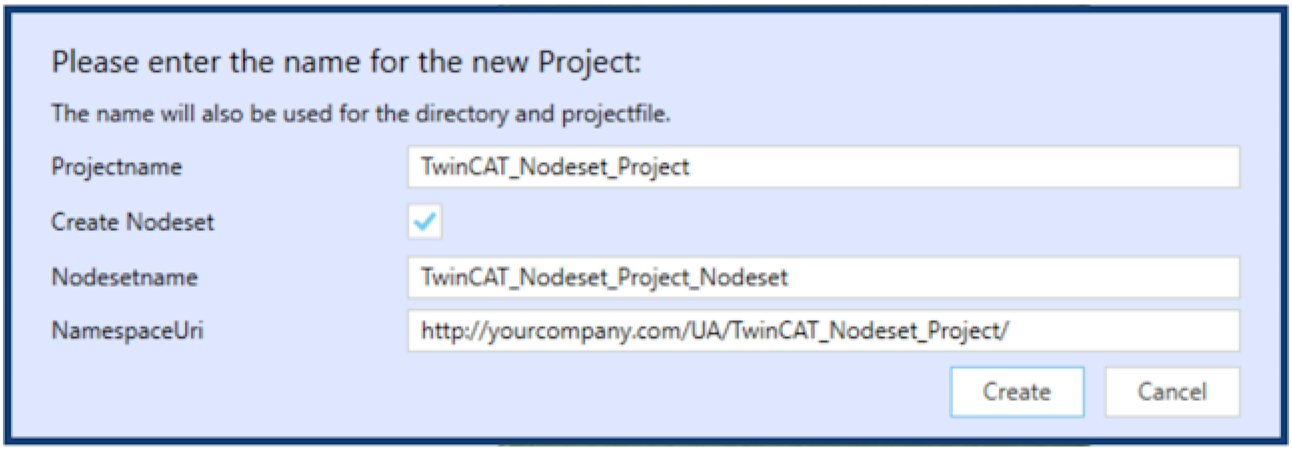
```
TYPE ST_MyStructure :
STRUCT
  MyBoolean : BOOL;
  MyInt16 : INT;
  MyFloat : REAL;
END_STRUCT
END_TYPE

PROGRAM MAIN
VAR
  MyStruct1 : ST_MyStructure;
  MyStruct2 : ST_MyStructure;
  MyStruct3 : ST_MyStructure;
END_VAR
```

Creating a new project

To create a new nodeset project, please open the menu **File** and click on **New \ Project**.



In the selection dialog that follows, you can specify the name and a target directory for the new project.



Please enter the name for the new Project:

The name will also be used for the directory and projectfile.

Projectname: TwinCAT_Nodeset_Project

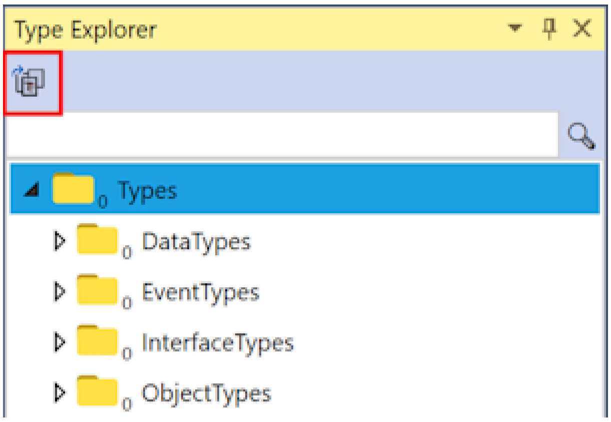
Create Nodeset: ☑

Nodesetname: TwinCAT_Nodeset_Project_Nodeset

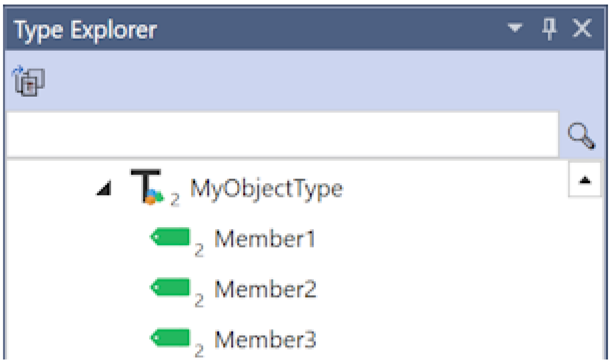
NamespaceUri: http://yourcompany.com/UA/TwinCAT_Nodeset_Project/

Importing the Companion Specification Nodeset

Import the file "te6100_quickstart_companion_specification.xml" from the GitHub repository into the TwinCAT OPC UA Nodeset Editor by clicking the button **LoadNodeset** in the Type Explorer.

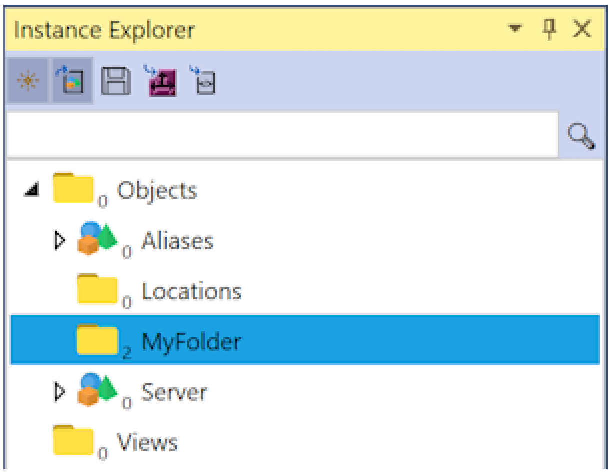


The import loads the type model from the nodeset into the Type Explorer and displays it there. The type model contains an ObjectType "MyObjectType", which we now want to instantiate. You can use the textbox to search for the type name or you can scroll to that type, which can be found under the ObjectTypes \ BaseObjectType node.

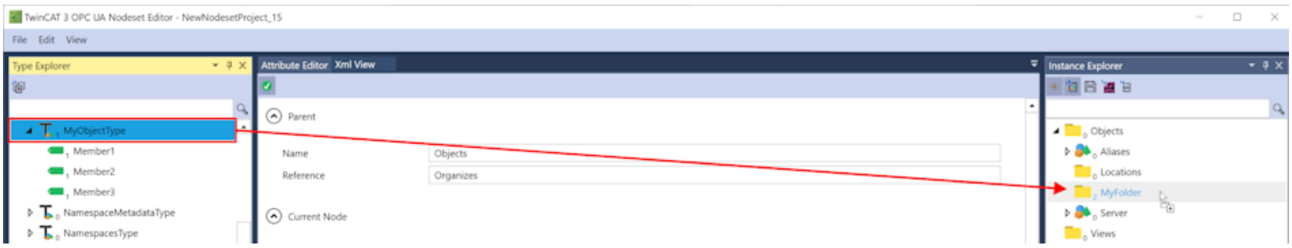


Creating object instances

Use the context menu to create a new folder in the Instance Explorer. You can rename the folder by double-clicking on it and then renaming the "Name" attribute in the Attribute Explorer. Then save this change via the **Save** button in the Attribute Explorer toolbar.



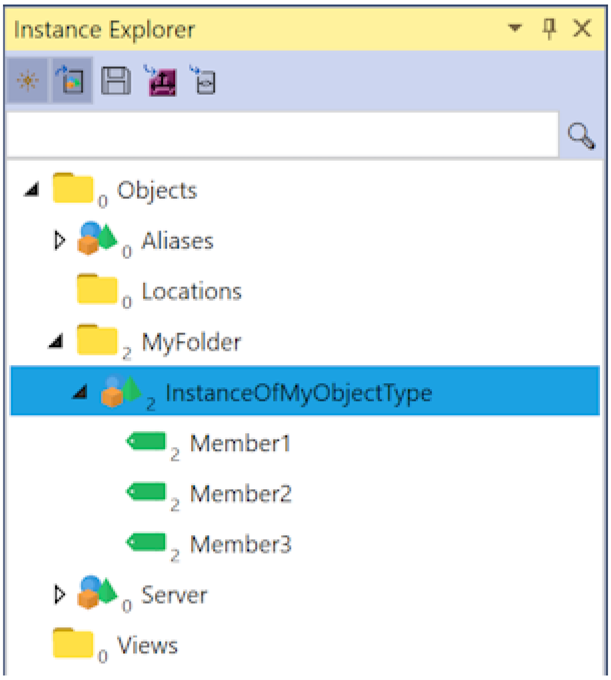
Then drag and drop the type "MyObjectType" from the Type Explorer into the created folder in the Instance Explorer.



Name Objects

Reference Organizes

This creates a new object instance of the type "MyObjectType" and places it in the folder created. All mandatory member variables of the type are transferred accordingly to the Instance Explorer and displayed there as nodes below the object instance. You can now link these nodes with SPS variables.



You can rename the instance via the Attribute Editor.

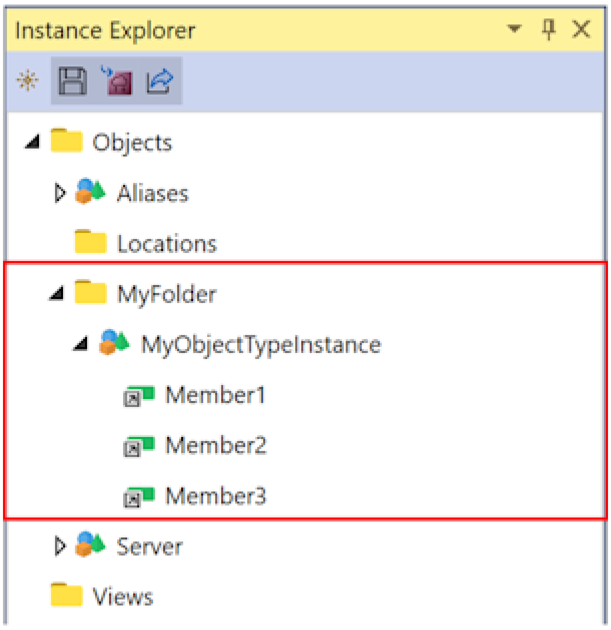
Linking the created instances with PLC variables

We now want to link the member variables of the now created object instance with variables from our prepared PLC project. For this purpose, the TwinCAT Target Browser can be used, which you can open in the menu **View**.

Navigate to the prepared PLC project in the Target Browser and link the individual member variables of the data structure with the corresponding member variables of the object instance by dragging and dropping them onto the corresponding node in the Instance Explorer. The data types here are as follows:

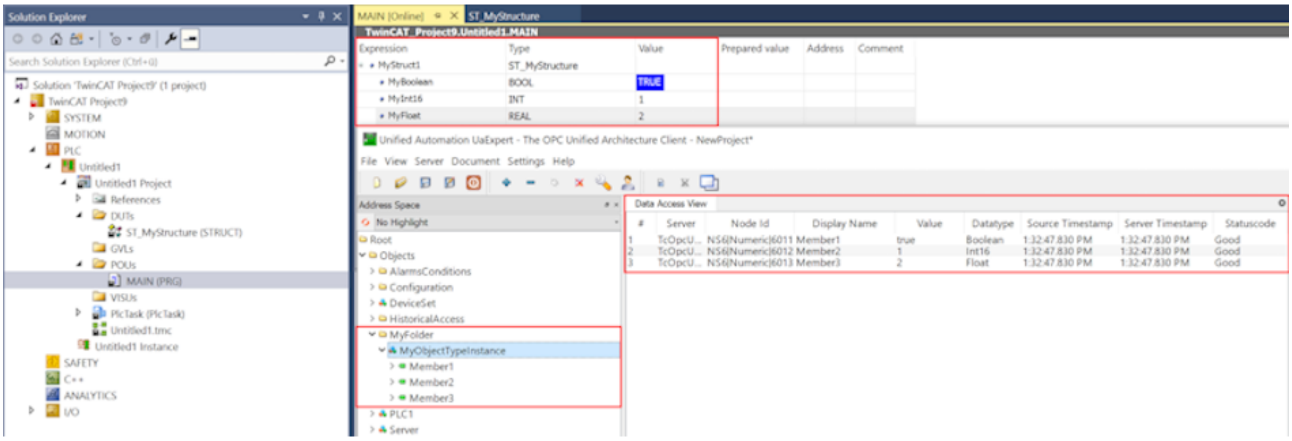
Member variable	Data type
Member1	Boolean (BOOL)
Member2	Int16 (INT)
Member3	Float (REAL)

You can also get this information from the Attribute Explorer if you double-click on the variable. As soon as you have linked all variables you can activate the project on the TwinCAT OPC UA Server in the next step.



Deploying the project to TwinCAT OPC UA Server

To deploy the project to the TwinCAT OPC UA Server, click on the **Export to TwinCAT OPC UA Server** button in the Instance Explorer toolbar. This exports the project nodeset to the locally installed TwinCAT OPC UA Server and also creates a corresponding configuration file for the server. Please make sure that you restart the TwinCAT OPC UA Server after you have deployed the nodeset so that the configuration is loaded. The address space of the server then contains the created nodes, which were linked with the corresponding PLC variables.



TwinCAT_Project9.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address	Comment
MyStruct1	ST_MyStructure				
MyBoolean	BOOL	TRUE			
MyInt16	INT	1			
MyFloat	REAL	2			

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	TcOpcU...	NS6|Numeric|6011	Member1	true	Boolean	1:32:47.830 PM	1:32:47.830 PM	Good
2	TcOpcU...	NS6|Numeric|6012	Member2	1	Int16	1:32:47.830 PM	1:32:47.830 PM	Good
3	TcOpcU...	NS6|Numeric|6013	Member3	2	Float	1:32:47.830 PM	1:32:47.830 PM	Good

4.2 Supported features

With the help of OPC UA, complex information models can be created. This page provides an overview about the currently available features of the TwinCAT OPC UA Nodeset Editor. Additional features will be provided as updates over time.

Feature	Description
Creation of nested folders	Allows the creation of (nested) folders, e.g. to structure instances.
Instantiation of simple data types	Allows the instantiation of simple data types from the OPC UA Base Nodeset. The following data types are supported: Boolean, Sbyte, Byte, Int16, UInt16, Int32, UInt32, Int64, UInt64, Float, Double, DateTime, String, NodeId, LocalizedText.
Instantiation of arrays of simple data types	Allows the instantiation of arrays of simple data types from the OPC UA Base Nodeset. The following data types are supported: Boolean, Sbyte, Byte, Int16, UInt16, Int32, UInt32, Int64, UInt64, Float, Double, DateTime, String, NodeId, LocalizedText.
Instantiation of object types	Allows instantiation of object types from an imported nodeset, e.g. a companion specification. Certain restrictions may apply, see below.
Instantiation of structured types	Allows the instantiation of structured types from an imported nodeset, e.g. a companion specification. Certain restrictions may apply, see below. This may require further configuration steps, which are explained in more detail in our documentation about Mapping [▶ 31].
Mapping of nodes	Allows linking of the created instances with variables from the PLC. The TwinCAT Target Browser serves as the basis for feature.
Handling of mandatory/optional nodes	When instantiating an object type, all sub elements are added automatically. Optional elements must be removed manually.
Export to TwinCAT OPC UA Server	Allows to export the created nodeset project to the TwinCAT OPC UA Server.

Restrictions

Below you will find an overview of which features are currently not available, yet. Additional features are added over time via software updates of this product.

Feature	Description
Dynamic array lengths	The instantiation of arrays with a dynamic length is possible in principle, but currently such arrays cannot be used in the TwinCAT PLC or the TwinCAT OPC UA Server.
Methods	Although the instantiation of methods is possible in principle, such methods can currently not be linked via the TwinCAT Target Browser.
Create/Modify types	Creating new data or object types or changing existing types is currently not supported.
Static node values	It is currently not possible to set a static value for a node.
References	Currently the references of a node cannot be modified. When adding folders or Instances of nodes, these are referenced using **Organizes** references.

4.3 License information

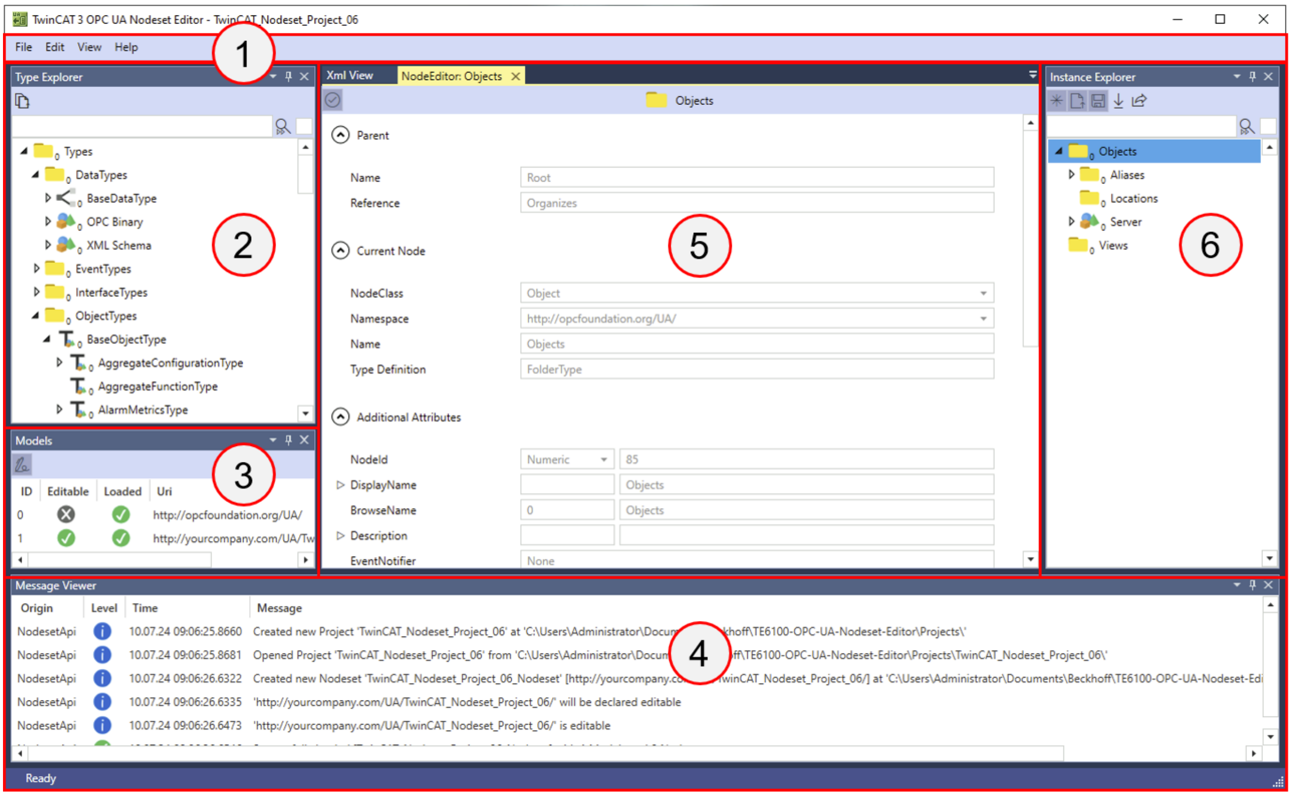
There are different licenses that can be purchased for the TwinCAT OPC UA Nodeset Editor. The following table provides an overview of the different licenses that are available for this product.

License	Description
7-day trial license	A 7-day trial license can be used with this product. It allows all the functionalities of the TE6100 license (see below) but doesn’t allow to save or deploy the project.
TE6100	This is the base license that is required to use the product. It allows access to all product functionalities but the ones defined by an add-on license.
TE6101	This license is an optional add-on license that allows to use the automatic code generation feature.

4.4 User interface

4.4.1 Overview

The user interface is divided into different areas, which will be described in more detail below. The following diagram illustrates the different areas.



ID	Editable	Loaded	Uri
0	☒	✓	http://opcfoundation.org/UA/
1	✓	✓	http://yourcompany.com/UA/Tw

Name: Root
Reference: Organizes
NodeClass: Object
Namespace: http://opcfoundation.org/UA/
Name: Objects
Type Definition: FolderType
NodeId: Numeric 85
DisplayName: Objects
BrowseName: 0 Objects
Description:
EventNotifier: None

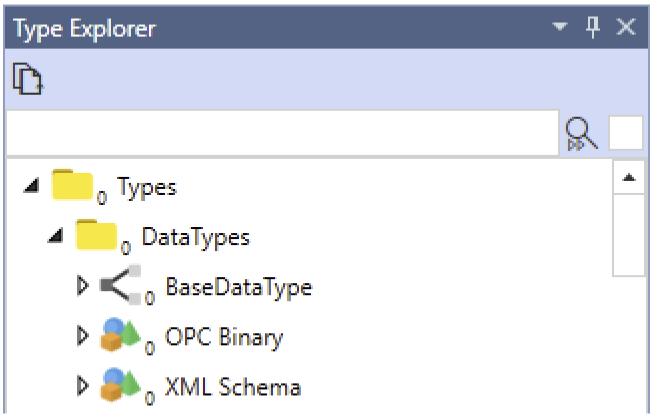
```
Origin      Level  Time                  Message
NodesetApi  i      10.07.24 09:06:25.8660  Created new Project 'TwinCAT_Nodeset_Project_06' at 'C:\Users\Administrator\Docum  khoff\TE6100-OPC-UA-Nodeset-Editor\Projects\'
NodesetApi  i      10.07.24 09:06:25.8681  Opened Project 'TwinCAT_Nodeset_Project_06' from 'C:\Users\Administrator\Docum  ff\TE6100-OPC-UA-Nodeset-Editor\Projects\TwinCAT_Nodeset_Project_06\'
NodesetApi  i      10.07.24 09:06:26.6322  Created new Nodeset 'TwinCAT_Nodeset_Project_06_Nodeset' [http://yourcompany.co  winCAT_Nodeset_Project_06/] at 'C:\Users\Administrator\Documents\Beckhoff\TE6100-OPC-UA-Nodeset-Edi
NodesetApi  i      10.07.24 09:06:26.6335  'http://yourcompany.com/UA/TwinCAT_Nodeset_Project_06/' will be declared editable
NodesetApi  i      10.07.24 09:06:26.6473  'http://yourcompany.com/UA/TwinCAT_Nodeset_Project_06/' is editable
```

Area	Name	Description
1	Menu [▶ 28]	Access to Load/Save mechanisms, as well as settings for the user interface and the TwinCAT Target Browser.
2	Type Explorer [▶ 25]	Displays the type system of the currently loaded nodesets. A textbox allows to search for nodes.
3	Models [▶ 28]	Displays the Namespaces of the currently loaded nodesets.
4	Message Viewer [▶ 28]	Contains logging information about the currently loaded project.
5	Attribute Editor [▶ 27]	Displays the attributes of the currently selected node. Both the attributes of nodes from the Type and Instance Explorer can be displayed here.
6	Instance Explorer [▶ 26]	Displays the currently configured instances. A textbox allows to search for nodes.

The individual areas are arranged in a so-called docked window concept and can be arranged differently depending on user requirements. The windows of the Type, Attribute and Instance Explorer also contain their own toolbar, which can be used to perform actions related to this area. These actions are described in more detail below.

4.4.2 Type Explorer

The Type Explorer displays all data types from the currently loaded nodesets. Further actions are available via the toolbar and a textbox allows to search for nodes.



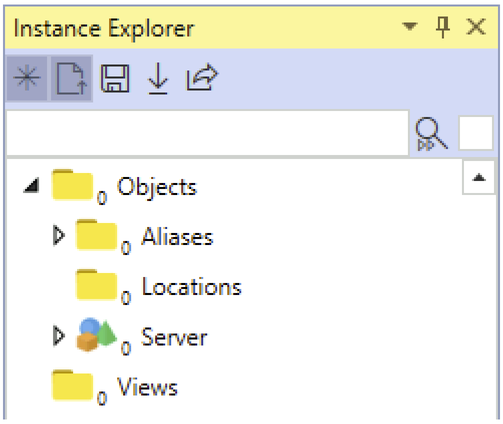
The numbers on each node represent the NamespaceIndex of a particular node and may be useful for easier identification of nodes.

The following table shows all actions that you can perform within the Type Explorer.

Action	Description
Import Nodeset	Allows to add more nodesets.
Drag&Drop	By using the Drag&Drop operation on a particular node and drag that node to the Instance Explorer, you can create an instance from that type.
Search	You can search for nodes by entering the name or NodeId of the node into the corresponding textbox. Additional search strings are: ns=x (searches for nodes in a particular NamespaceIndex)

4.4.3 Instance Explorer

The Instance Explorer displays all instances in this project. Further actions are available via the toolbar.

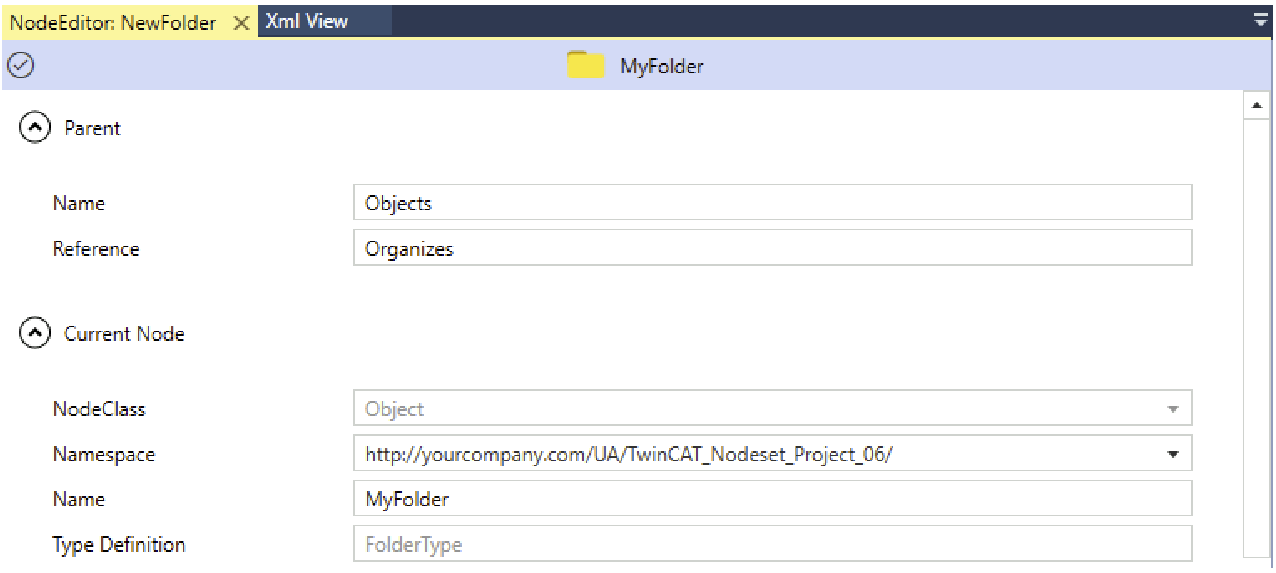


The following table shows all actions that you can perform within the Instance Explorer.

Action	Description
Create new Nodeset	Creates a new instance nodeset.
Save changes to Nodeset	Saves current changes to disk.
Deploy to TwinCAT OPC UA Server	Exports the current project to the TwinCAT OPC UA Server.
Export Nodesets	Export the current instance nodeset to a file.
Search	You can search for nodes by entering the name or NodeId of the node into the corresponding textbox. Additional search strings are: ns=x (searches for nodes in a particular NamespaceIndex) HasExtension (searches for nodes with a mapping to a PLC variable) HasNoExtension (searches for nodes without a mapping to a PLC variable)
Context menu (right mouse button) Delete node	Removes the currently selected node and tightly coupled children. Please note that other child nodes (without a ParentNodeId) are not removed and considered to be "orphaned" if they have no other parent node.
Context menu (right mouse button) Create folder	Creates a new node of type FolderType.

4.4.4 Attribute Editor

The Attribute Explorer displays the attributes of the currently selected node. Both the attributes of nodes from the Type Explorer and the Instance Explorer can be displayed. By double-clicking on a node, its attributes are loaded and displayed accordingly. Further actions are available via the toolbar.



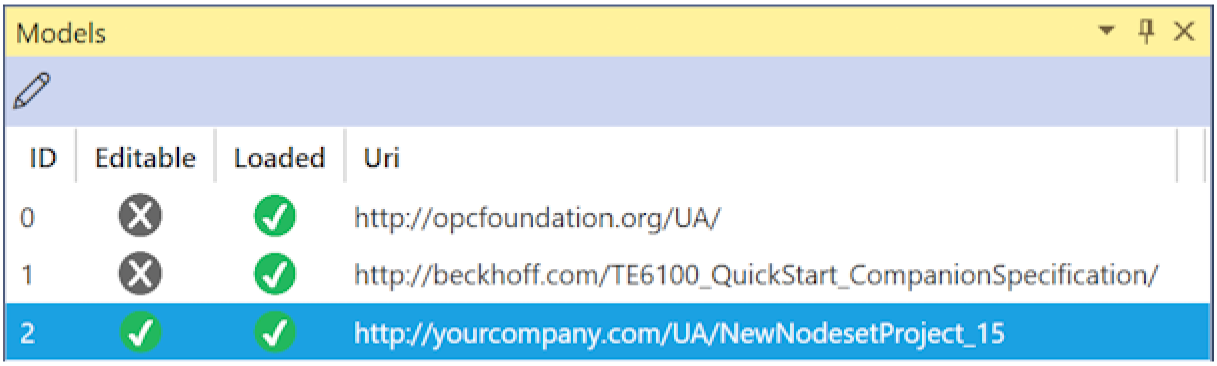
Name: Objects
Reference: Organizes
NodeClass: Object
Namespace: http://yourcompany.com/UA/TwinCAT_Nodeset_Project_06/
Name: MyFolder
Type Definition: FolderType

The following table shows all actions that you can perform within the Attribute Editor.

Action	Description
Edit attributes	Edit attributes of the currently opened node.
Save	Applies any changes to the attributes.
XML View	Allows to display the currently selected node as XML.

4.4.5 Models

The **Information Models** window shows all currently loaded namespaces – both from all loaded type or instance nodesets. This window allows you to choose which namespace you want to edit when creating instances out of a data or object type. The namespace that is currently marked as editable is used when creating new instances.



ID	Editable	Loaded	Uri
0	☒	✓	http://opcfoundation.org/UA/
1	☒	✓	http://beckhoff.com/TE6100_QuickStart_CompanionSpecification/
2	✓	✓	http://yourcompany.com/UA/NewNodesetProject_15

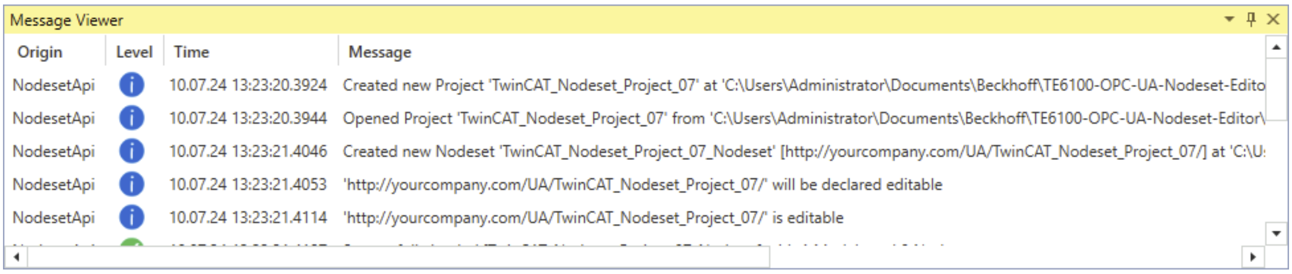
4.4.6 Menu

The menu includes different mechanisms related to the currently opened project. The following table provides more information about each menu entry.

Menu	Sub item	Description
File	New	Creates a new project.
File	Load	Loads an existing project.
File	Save	Saves the currently opened project.
File	Export/Deploy	Exports the currently opened project into a single nodeset file. In addition, the currently opened project can be deployed to the TwinCAT OPC UA Server.
View	Theme	Allows to switch between different visual themes.
View	Target Browser	Opens the TwinCAT Target Browser.
View	Type Explorer	Shows/Hides the Type Explorer.
View	Models	Shows/Hides the Models window.
View	Instance Explorer	Shows/Hides the Instance Explorer.
View	Message Viewer	Shows/Hides the Message Viewer.
View	XML View	Shows/Hides the XML View.
View	Welcome	Shows/Hides the Welcome screen.
Help	Info	Displays license information.

4.4.7 Message Viewer

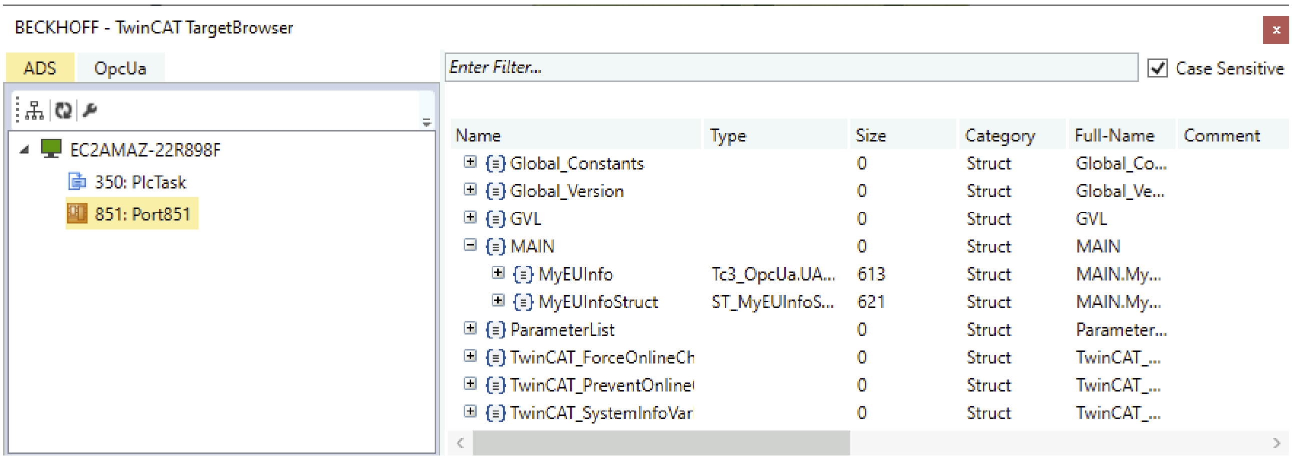
The Message Viewer contains logging information from your currently opened project. Messages are categorized into different severity levels and their origin. In addition, each message has a timestamp that indicates when the message has been thrown.



Origin	Level	Time	Message
NodesetApi	ⓘ	10.07.24 13:23:20.3924	Created new Project 'TwinCAT_Nodeset_Project_07' at 'C:\Users\Administrator\Documents\Beckhoff\TE6100-OPC-UA-Nodeset-Edito
NodesetApi	ⓘ	10.07.24 13:23:20.3944	Opened Project 'TwinCAT_Nodeset_Project_07' from 'C:\Users\Administrator\Documents\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\
NodesetApi	ⓘ	10.07.24 13:23:21.4046	Created new Nodeset 'TwinCAT_Nodeset_Project_07_Nodeset' [http://yourcompany.com/UA/TwinCAT_Nodeset_Project_07/] at 'C:\U:
NodesetApi	ⓘ	10.07.24 13:23:21.4053	'http://yourcompany.com/UA/TwinCAT_Nodeset_Project_07/' will be declared editable
NodesetApi	ⓘ	10.07.24 13:23:21.4114	'http://yourcompany.com/UA/TwinCAT_Nodeset_Project_07/' is editable

4.4.8 TwinCAT Target Browser

The TwinCAT Target Browser can be opened via the **View** menu. It has currently been implemented as a separate window but will be integrated into the main (docked) window structure in future versions.



BECKHOFF - TwinCAT TargetBrowser

ADS | OpcUa

EC2AMAZ-22R898F
- 350: PlcTask
- 851: Port851

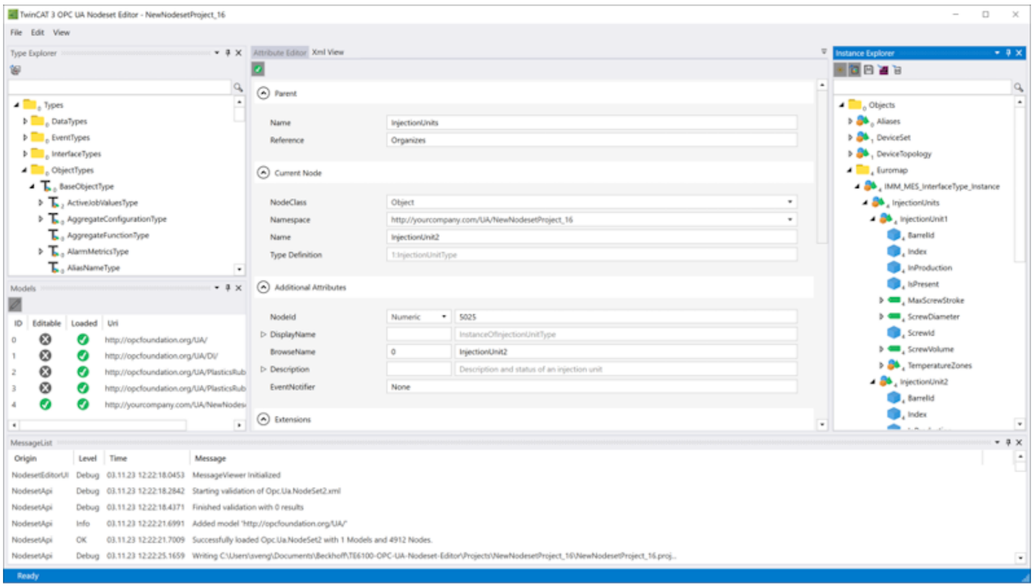
Enter Filter...

☑ Case Sensitive

Name	Type	Size	Category	Full-Name	Comment
⊞ Global_Constants		0	Struct	Global_Co...	
⊞ Global_Version		0	Struct	Global_Ve...	
⊞ GVL		0	Struct	GVL	
⊟ MAIN		0	Struct	MAIN	
⊞ MyEUInfo	Tc3_OpcUa.UA...	613	Struct	MAIN.My...	
⊞ MyEUInfoStruct	ST_MyEUInfoS...	621	Struct	MAIN.My...	
⊞ ParameterList		0	Struct	Parameter...	
⊞ TwinCAT_ForceOnlineCh		0	Struct	TwinCAT_...	
⊞ TwinCAT_PreventOnline		0	Struct	TwinCAT_...	
⊞ TwinCAT_SystemInfoVar		0	Struct	TwinCAT_...	

4.4.9 Themes

The TwinCAT OPC UA Nodeset Editor currently supports three different color themes: Light, Dark and Blue (default). You can select the theme of your choice via the **View** menu. The setting can be persisted in the application settings [▶ 34].



Parent

Name	InjectionUnits
Reference	Organizes

Current Node

NodeClass	Object
Namespace	http://yourcompany.com/UA/NewNodesetProject_16
Name	InjectionUnit2
Type Definition	1:InjectionUnitType

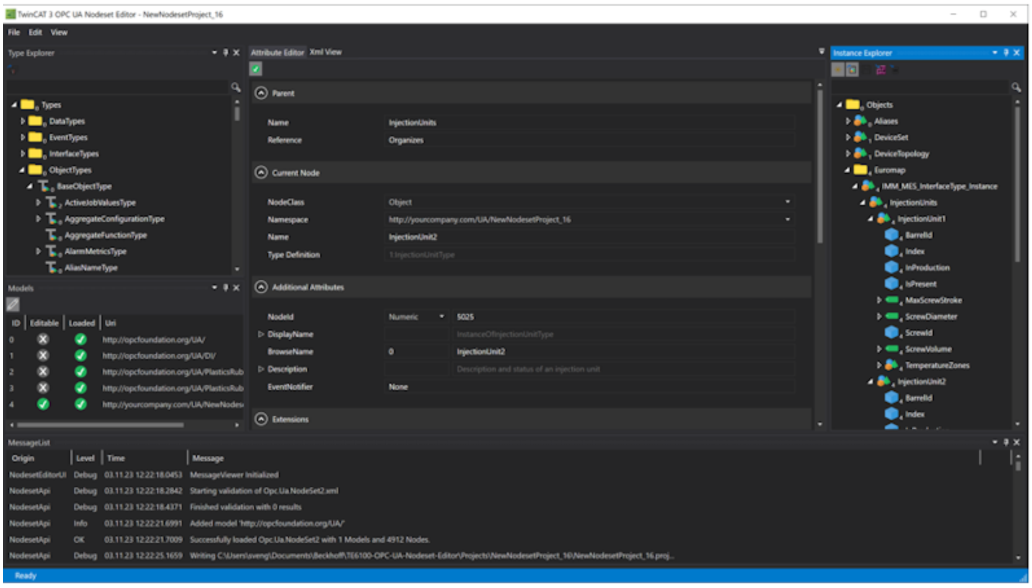
Additional Attributes

NodeId	Numeric	5025
▷ DisplayName		InstanceOfInjectionUnitType
BrowseName	0	InjectionUnit2
▷ Description		Description and status of an injection unit
EventNotifier	None	

Extensions

ID	Editable	Loaded	Uri
0	☒	✓	http://opcfoundation.org/UA/
1	☒	✓	http://opcfoundation.org/UA/DI/
2	☒	✓	http://opcfoundation.org/UA/PlasticsRub
3	☒	✓	http://opcfoundation.org/UA/PlasticsRub
4	✓	✓	http://yourcompany.com/UA/NewNodes

```
Origin          Level  Time                   Message
NodesetEditorUI Debug  03.11.23 12:22:18.0453 MessageViewer Initialized
NodesetApi      Debug  03.11.23 12:22:18.2842 Starting validation of Opc.Ua.NodeSet2.xml
NodesetApi      Debug  03.11.23 12:22:18.4371 Finished validation with 0 results
NodesetApi      Info   03.11.23 12:22:21.6991 Added model 'http://opcfoundation.org/UA/'
NodesetApi      OK     03.11.23 12:22:21.7009 Successfully loaded Opc.Ua.NodeSet2 with 1 Models and 4912 Nodes.
NodesetApi      Debug  03.11.23 12:22:25.1659 Writing C:\Users\sveng\Documents\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\Projects\NewNodesetProject_16\NewNodesetProject_16.proj...
```

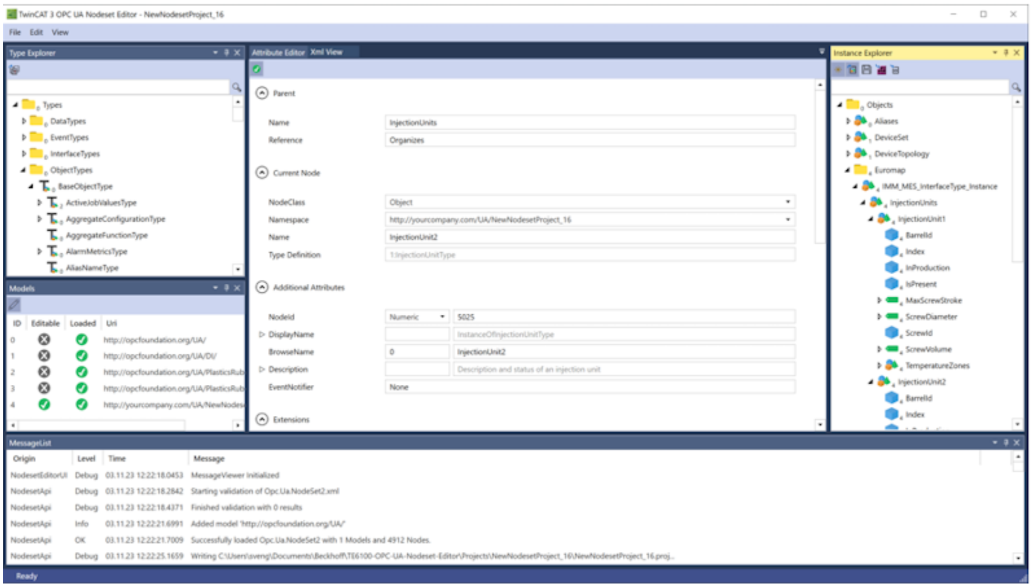


TwinCAT 3 OPC UA Nodeset Editor - NewNodesetProject_16

Name	InjectionUnits
Reference	Organizes
NodeClass	Object
Namespace	http://yourcompany.com/UA/NewNodesetProject_16
Name	InjectionUnit2
Type Definition	1:InjectionUnitType

ID	Editable	Loaded	Uri
0	☒	✓	http://opcfoundation.org/UA/
1	☒	✓	http://opcfoundation.org/UA/DI/
2	☒	✓	http://opcfoundation.org/UA/PlasticsRub
3	☒	✓	http://opcfoundation.org/UA/PlasticsRub
4	✓	✓	http://yourcompany.com/UA/NewNodes

```
Origin          Level  Time                   Message
NodesetEditorUI Debug  03.11.23 12:22:18.0453 MessageViewer Initialized
NodesetApi      Debug  03.11.23 12:22:18.2842 Starting validation of Opc.Ua.NodeSet2.xml
NodesetApi      Debug  03.11.23 12:22:18.4371 Finished validation with 0 results
NodesetApi      Info   03.11.23 12:22:21.6991 Added model 'http://opcfoundation.org/UA/'
NodesetApi      OK     03.11.23 12:22:21.7009 Successfully loaded Opc.Ua.NodeSet2 with 1 Models and 4912 Nodes.
NodesetApi      Debug  03.11.23 12:22:25.1659 Writing C:\Users\sveng\Documents\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\Projects\NewNodesetProject_16\NewNodesetProject_16.proj...
```



TwinCAT 3 OPC UA Nodeset Editor - NewNodesetProject_16

Name	InjectionUnits
Reference	Organizes
NodeClass	Object
Namespace	http://yourcompany.com/UA/NewNodesetProject_16
Name	InjectionUnit2
Type Definition	1:InjectionUnitType

ID	Editable	Loaded	Uri
0	☒	✓	http://opcfoundation.org/UA/
1	☒	✓	http://opcfoundation.org/UA/DI/
2	☒	✓	http://opcfoundation.org/UA/PlasticsRub
3	☒	✓	http://opcfoundation.org/UA/PlasticsRub
4	✓	✓	http://yourcompany.com/UA/NewNodes

```
Origin          Level  Time                   Message
NodesetEditorUI Debug  03.11.23 12:22:18.0453 MessageViewer Initialized
NodesetApi      Debug  03.11.23 12:22:18.2842 Starting validation of Opc.Ua.NodeSet2.xml
NodesetApi      Debug  03.11.23 12:22:18.4371 Finished validation with 0 results
NodesetApi      Info   03.11.23 12:22:21.6991 Added model 'http://opcfoundation.org/UA/'
NodesetApi      OK     03.11.23 12:22:21.7009 Successfully loaded Opc.Ua.NodeSet2 with 1 Models and 4912 Nodes.
NodesetApi      Debug  03.11.23 12:22:25.1659 Writing C:\Users\sveng\Documents\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\Projects\NewNodesetProject_16\NewNodesetProject_16.proj...
```

4.5 Mapping

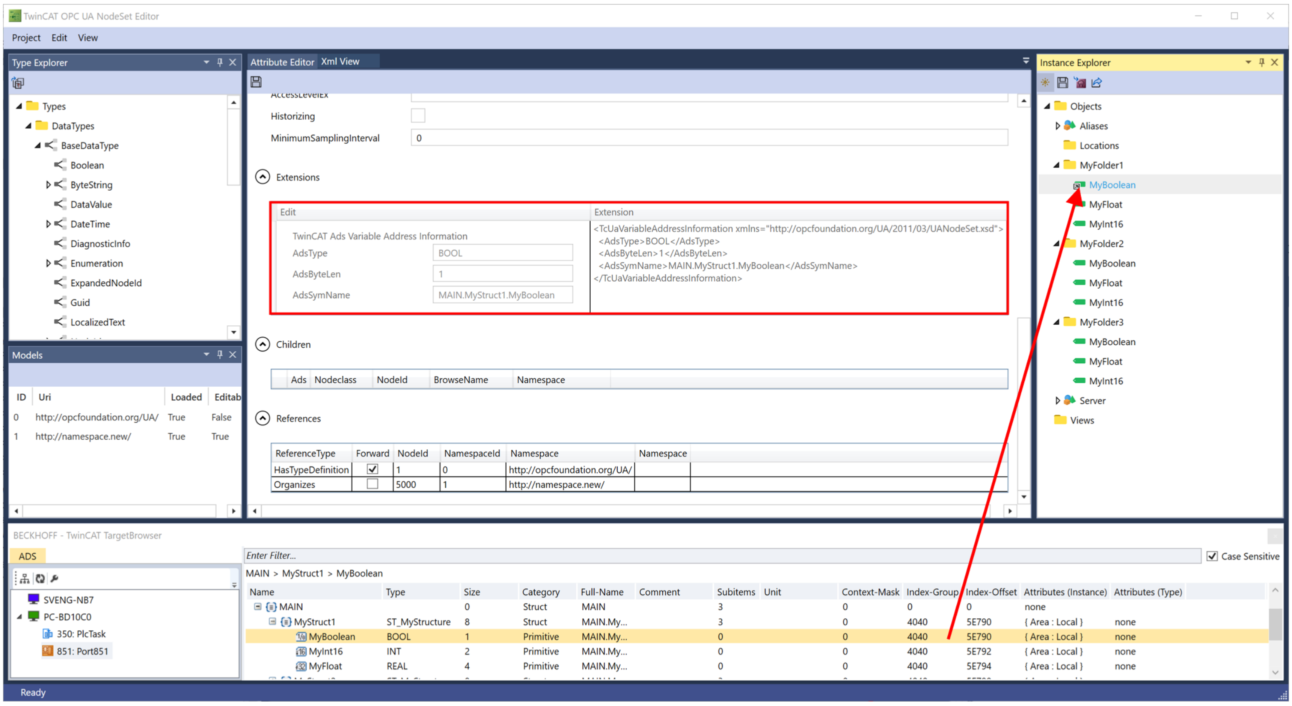
The TwinCAT Target Browser is available for linking OPC UA nodes to PLC symbols, which will add the address information of a PLC symbol to an OPC UA node. The nodeset can then deployed to the TwinCAT OPC UA Server.

Objects

Objects cannot be linked to PLC symbols, since they do not have a "value" that could be linked.

Variables and properties

The mapping of variables and properties of a simple data type (Int32, Double, Boolean, ...) can be done by using the TwinCAT Target Browser. Please note that the data type of the PLC variable must "match" the data type of the OPC UA node.



ID	Uri	Loaded	Editab
0	http://opcfoundation.org/UA/	True	False
1	http://namespace.new/	True	True

Historizing □

MinimumSamplingInterval 0

Extensions

Edit — TwinCAT Ads Variable Address Information

AdsType: BOOL

AdsByteLen: 1

AdsSymName: MAIN.MyStruct1.MyBoolean

Extension:

```
<TcUaVariableAddressInformation xmlns="http://opcfoundation.org/UA/2011/03/UANodeSet.xsd">
  <AdsType>BOOL</AdsType>
  <AdsByteLen>1</AdsByteLen>
  <AdsSymName>MAIN.MyStruct1.MyBoolean</AdsSymName>
</TcUaVariableAddressInformation>
```

Children

Ads	Nodeclass	NodeId	BrowseName	Namespace

References

ReferenceType	Forward	NodeId	NamespaceId	Namespace	Namespace
HasTypeDefinition	☑	1	0	http://opcfoundation.org/UA/	
Organizes	□	5000	1	http://namespace.new/	

Enter Filter... ☑ Case Sensitive

MAIN > MyStruct1 > MyBoolean

Name	Type	Size	Category	Full-Name	Comment	Subitems	Unit	Context-Mask	Index-Group	Index-Offset	Attributes (Instance)	Attributes (Type)
MAIN		0	Struct	MAIN		3		0	0	0	none	
MyStruct1	ST_MyStructure	8	Struct	MAIN.My...		3		0	4040	5E790	{ Area : Local }	none
MyBoolean	BOOL	1	Primitive	MAIN.My...		0		0	4040	5E790	{ Area : Local }	none
MyInt16	INT	2	Primitive	MAIN.My...		0		0	4040	5E792	{ Area : Local }	none
MyFloat	REAL	4	Primitive	MAIN.My...		0		0	4040	5E794	{ Area : Local }	none

The following table gives an overview of common, simple data types as defined in the OPC UA Base namespace http://opcfoundation.org/UA/. It also shows the corresponding IEC61131 data type.

OPC UA data type	PLC data type
Boolean	BOOL
Sbyte	SINT
Byte	INT
Int16	INT
UInt16	UINT
Int32	DINT
UInt32	UDINT
Int64	LINT
UInt64	ULINT
Float	REAL
Double	LREAL
DateTime	DT
String	STRING

Structures

Structured types can also be linked to a PLC structure via the TwinCAT Target Browser. However, there are a few special features to be considered here. On the one hand, the data structure in the PLC must "fit" the OPC UA data structure and, on the other hand, the data structure in the PLC still requires the OPC UA type information as a PLC attribute.

The following OPC UA structure is given as an example.

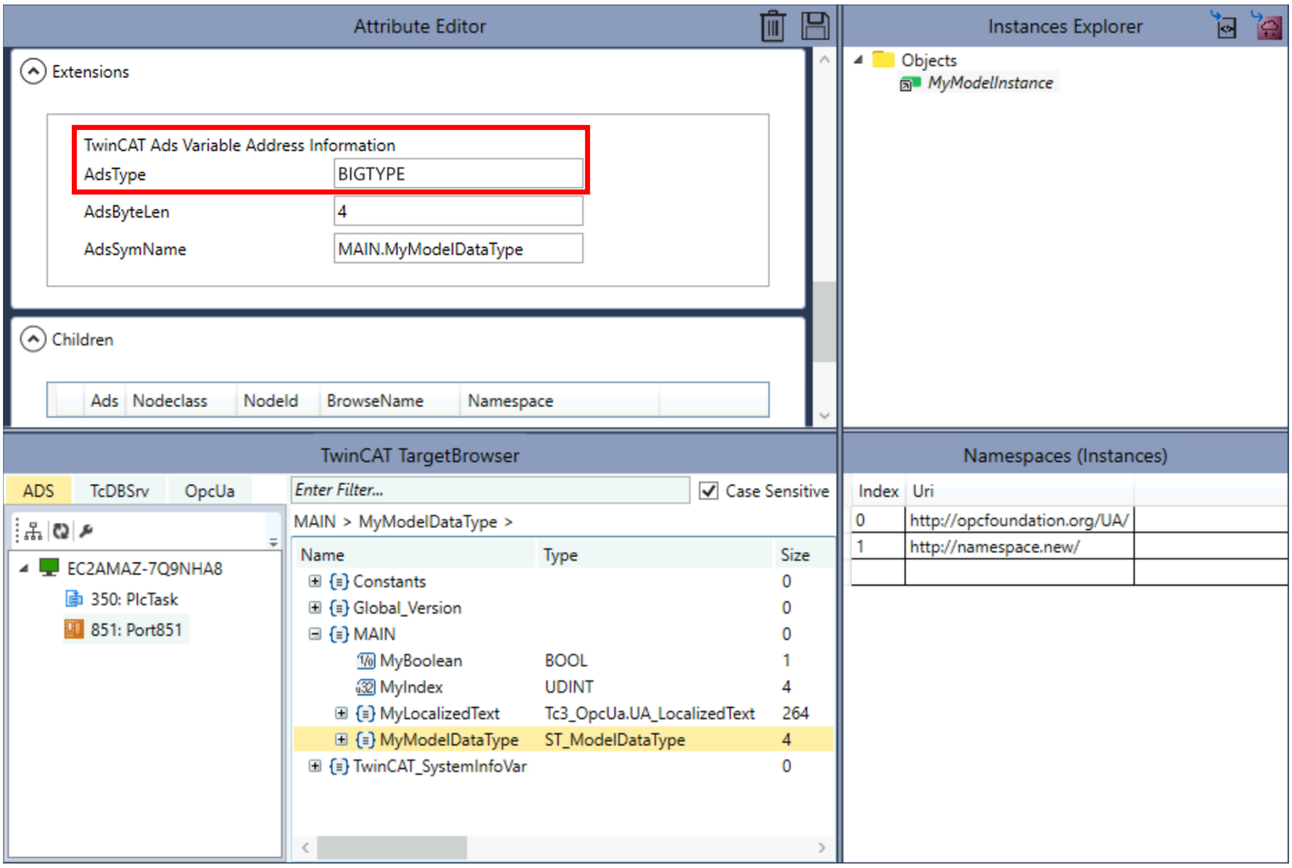
```
<UADataType NodeId="ns=1;i=3015" BrowseName="1:ModelDataType">
  <DisplayName>ModelDataType</DisplayName>
  <References>
    <Reference ReferenceType="HasEncoding">ns=1;i=5048</Reference>
    <Reference ReferenceType="HasEncoding">ns=1;i=5050</Reference>
    <Reference ReferenceType="HasEncoding">ns=1;i=5049</Reference>
    <Reference ReferenceType="HasSubtype" IsForward="false">i=22</Reference>
  </References>
  <Definition Name="1:ModelDataType">
    <Field DataType="Byte" Name="SwitchingObjectIdentification"/>
    <Field DataType="Byte" Name="SwitchingObjectNumber"/>
    <Field DataType="Date" Name="CurrentDate"/>
    <Field DataType="Int32" ValueRank="1" ArrayDimensions="1" Name="SwitchingTimes"/>
  </Definition>
</UADataType>
```

The structure in the PLC must then have the following structure:

```
PROGRAM MAIN
VAR
  MyModel : ST_ModelDataType;
END_VAR

{attribute 'OPC.UA.DA' := '1'}
{attribute 'OPC.UA.DA.StructuredType' := '1'}
{attribute 'pack_mode' := '1'}
{attribute 'OPC.UA.AdditionalStructuredType.NamespaceName' := 'nameSpaceNameNodeset'}
{attribute 'OPC.UA.AdditionalStructuredType.Id' := 'i=3015'}
TYPE ST_ModelDataType :
STRUCT
  SwitchingObjectIdentification : BYTE;
  SwitchingObjectNumber : BYTE;
  CurrentDate : DT;
  SwitchingTimes : ARRAY[0..2] OF DINT;
END_STRUCT
END_TYPE
```

The additional attributes OPC.UA.AdditionalStructuredType.NamespaceName and OPC.UA.AdditionalStructuredType.id are necessary so that the TwinCAT OPC UA Server can assign the type information. They represent the namespace and the NodeID of the data type of the StructuredType.

The link in the TwinCAT Target Browser can then be made to the structure instance MyTimeModel. Please make sure afterwards that the AdsType "BIGINT" is entered in the TcAdsExtension at the OPC UA Node.



TwinCAT Ads Variable Address Information

AdsType	BIGTYPE
AdsByteLen	4
AdsSymName	MAIN.MyModelDataType

Ads	Nodeclass	NodeId	BrowseName	Namespace

- Objects
  - MyModelInstance

- EC2AMAZ-7Q9NHA8
  - 350: PlcTask
  - 851: Port851

MAIN > MyModelDataType >

Name	Type	Size
⊞ Constants		0
⊞ Global_Version		0
⊟ MAIN		0
MyBoolean	BOOL	1
MyIndex	UDINT	4
⊞ MyLocalizedText	Tc3_OpcUa.UA_LocalizedText	264
⊞ MyModelDataType	ST_ModelDataType	4
⊞ TwinCAT_SystemInfoVar		0

Index	Uri
0	http://opcfoundation.org/UA/
1	http://namespace.new/

4.6 Application directories

This application uses different directories to store relevant information, e.g. configuration files, nodeset libraries and files related to the current project.

Installation folder

The installation folder of this application is located under the TwinCAT installation directory.

```
%TcInstallDir%\Functions\TE6100-OPC-UA-Nodeset-Editor
```

Configuration files

This application uses different configuration files, which are located in a sub folder of the ProgramData directory:

```
%ProgramData%\Beckhoff\TE6100-OPC-UA-Nodeset-Editor
```

Libraries

There is a central directory in which you can store nodeset files that you repeatedly need to use in your projects as libraries. Every time you want to import a nodeset file into your current project, the TwinCAT OPC UA Nodeset Editor opens this “Library” directory, which is located in a sub folder of the ProgramData directory:

```
%ProgramData%\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\Library
```

Projects

Per default, new projects are created in a sub folder of the user’s home directory:

```
%UserProfile%\Documents\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\Projects
```

Log files

The TwinCAT OPC UA Nodeset Editor creates log files that may contain useful information in case anything goes wrong. These log files are located in a sub folder of the ProgramData directory:

```
%ProgramData%\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\Log
```

4.7 Application and project settings

The application itself and each project includes XML-based configuration files, which contain different settings that influence the behavior of the application. Such settings are described in the following chapters.

Application settings

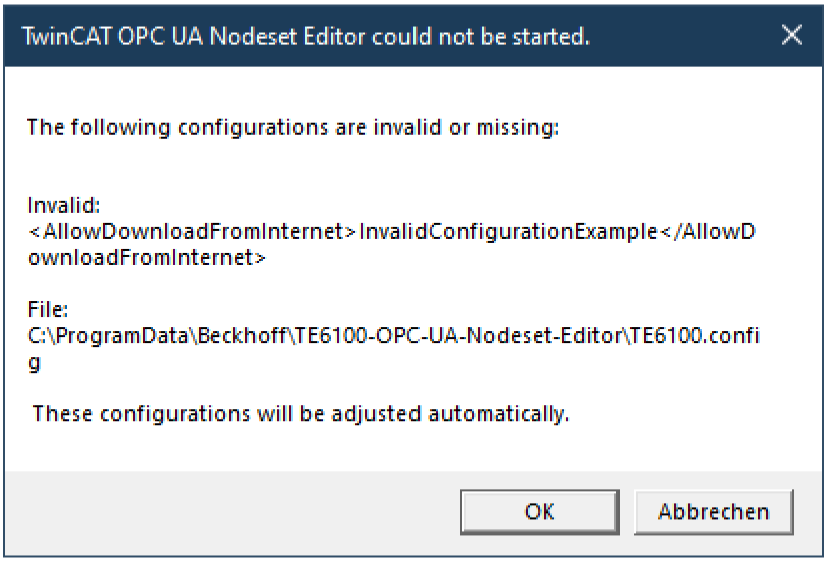
The configuration file directory (see Application directories [▶ 33]) includes the file TE6100.config which contains configuration parameters for the TwinCAT OPC UA Nodeset Editor application. You can open this file with any text editor of your choice. The following parameters are available.

Section	Parameter	Description
General	AllowDownloadFromInternet	The nodeset editor already contains a specific version of the base nodeset. By changing this setting to “true”, the most current version will be downloaded from the OPC Foundation instead.
General	DefaultLocalOpcUaServerDir	Default location of the TwinCAT OPC UA Server installation. This path is used when deploying a nodeset to the TwinCAT OPC UA Server.
General	ProjectsDir	Default base directory for new projects.
General	ShowNamespaceIndexInTree	Enables/Disables the NamespaceIndex to be shown as a number in Type and Instance Explorer.
General	Theme	Sets the default theme for the application.
Nodesets	CoreNodesetFilePath	Path to the OPC UA base nodeset (used for all base data and object types). This nodeset is loaded per default for every new project.
Nodesets	LibraryDir	Default base directory for type nodesets. This directory is opened per default whenever you import type nodesets.
Nodesets	OrderAttributesOnExport	If set to true, this will order the attributes of all XML Elements with common attributes first (e.g. NodeId and BrowseName) and others following in alphabetical order. Tip: If you want to compare a nodeset saved by this software with another nodeset (e.g. with “git diff”), it can be useful to set this to false to avoid unnecessary changes.
Nodesets	OrderNodesOnExport	If set to true, this will order the nodes by NamespaceId and NodeId and subelements (DisplayName etc.) uniformly. Tip: If you want to compare a nodeset saved by this software with another nodeset (e.g. with “git diff”), it can be useful to set this to false to avoid unnecessary changes.
Nodesets	XmlSchemaFilePath	Path to the OPC UA XmlSchemaFile, which is used for (syntax-) validation of nodesets.

Project settings

Parameter	Description
LocalOpcUaServer	Default location of the TwinCAT OPC UA Server installation. This path is used when deploying a nodeset to the TwinCAT OPC UA Server. When creating a new project, this path is taken from the corresponding application settings parameter (see above) but you can override it on a per project basis.
Nodesets	Contains a list of all nodesets that belong to this project including their path, NamespaceName, version and PublicationDate.

If an invalid configuration is recognized, you will be presented with a dialog informing you about the invalid parameter.



TwinCAT OPC UA Nodeset Editor could not be started.

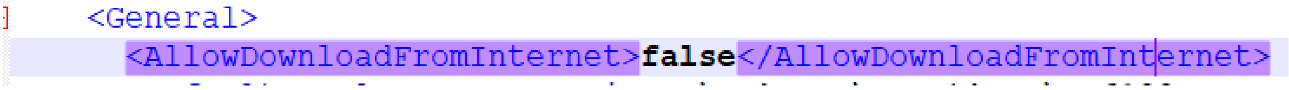
The following configurations are invalid or missing:

Invalid:
<AllowDownloadFromInternet>InvalidConfigurationExample</AllowDownloadFromInternet>

File:
C:\ProgramData\Beckhoff\TE6100-OPC-UA-Nodeset-Editor\TE6100.config

These configurations will be adjusted automatically.

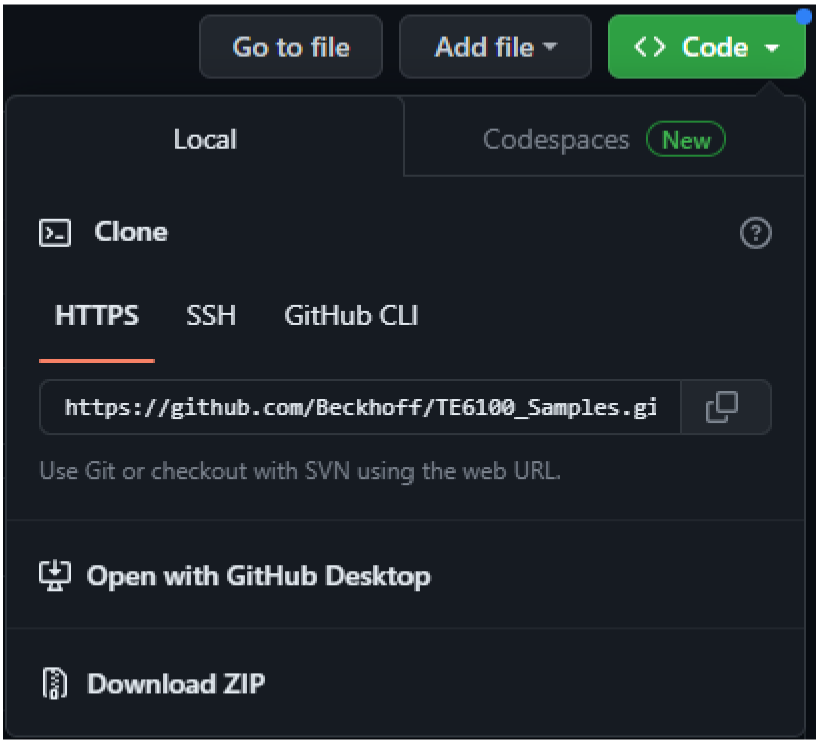
The software will attempt to correct invalid configurations automatically if you click **OK**. This is done by resetting the configuration to its default value. Example:



```
  <General>
    <AllowDownloadFromInternet>false</AllowDownloadFromInternet>
```

5 Samples

Sample code and configurations for this product can be obtained from the corresponding repository on GitHub: https://github.com/Beckhoff/TE6100_Samples. There you have the possibility to clone the repository or download a ZIP file with the sample.



https://github.com/Beckhoff/TE6100_Samples.gi

Use Git or checkout with SVN using the web URL.

We also recommend to check out our quick start tutorials [▸ 13], which provide a guided walkthrough of various configuration use cases.

6 Appendix

6.1 Terminology

Nodeset

A nodeset is a file that formally describes an OPC UA information model. Nodeset files are usually XML-based and must be structured according to an XML schema definition. Nodesets are used for processing information models with the computer. A nodeset defines nodes, their attributes, and references. A reference can point to nodes in the same document as well as to nodes outside the file. OPC UA itself defines some central information models with corresponding nodesets, for example:

Name	Namespace
Core („Base“)	http://opcfoundation.org/UA
DI	http://opcfoundation.org/UA/DI
GDS	http://opcfoundation.org/UA/GDS

Particularly noteworthy is the "Core" model, whose nodeset is also colloquially referred to as the "Base" nodeset. The core model usually serves as a basis for other information models and describes the basic data types that are available in OPC UA.

Companion specifications

Although the existing OPC UA information models are already applicable in many industries, it is possible to create own information models to meet specific requirements. For this purpose, companies and organizations join forces to define industry-specific models, so-called "companion specifications". Such companion specifications are available, for example, for the packaging industry (PackML) or for plastics processing (EUROMAP).

Further information on this topic can be found on the OPC Foundation website.

Node

A node is a fundamental part of the address space of a server and represents an object, variable or method. Even data and object types are represented on the server’s address space as a node.

6.2 Troubleshooting

Behavior	Description
Cannot export nodeset to TwinCAT OPC UA Server. The log shows an error “Failed to write …”.	Please make sure that you have write access to the TwinCAT OPC UA Server installation directory. If the TwinCAT OPC UA Server has been installed to the %ProgramFiles% directory, please make sure to start the TwinCAT OPC UA Nodeset Editor as Administrator using the “Run As Administrator” option.
I have deployed my nodeset to the TwinCAT OPC UA Server but the server namespace does not show my configured nodes.	Please make sure that you have configured the correct installation path of the TwinCAT OPC UA Server in the application settings [▶ 34]. On a 32-bit system, the Win32 installation directory is used whereas on a 64-bit system the Win64 directory is used to execute the TcOpcUaServer.exe. You can also double-check which directory used via the Windows Task Manager by navigating to the TcOpcUaServer process and selecting **Open file location** from the context menu.

7 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

More Information:
www.beckhoff.com/te6100

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