

## Automation Designer

### Getting Started

Concepts

1

Overview of this Getting  
Started

2

TeamCenter: Create Line  
Designer Collaborative  
Design

3

Line Designer: Create Line  
Designer workset, subset  
and Design Elements

4

Create Automation Designer  
workset and Engineering  
Objects

5

Map Line Designer and  
Automation Designer

6

Configure (non-template)  
EPLAN

7

Configure (non-template) TIA  
Portal

8

Template-related concepts

9

Configure template-ready  
EPLAN

10

Configure template-ready  
TIA Portal

11

Create/instantiate template

12

## Legal information

### Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

#### DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

#### WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

#### CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

#### NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

### Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

### Proper use of Siemens products

Note the following:

#### WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

### Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

### Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

# Table of contents

<b>1</b>	<b>Concepts.....</b>	<b>7</b>
1.1	Traditional engineering versus integrated engineering.....	7
1.2	Reuse library.....	9
1.3	TeamCenter (4GD) details.....	17
<b>2</b>	<b>Overview of this Getting Started.....</b>	<b>23</b>
2.1	Prerequisites.....	23
2.2	Workflow from the tools perspective.....	24
2.3	Workflow.....	25
<b>3</b>	<b>TeamCenter: Create Line Designer Collaborative Design.....</b>	<b>31</b>
3.1	Create plant design Collaborative Design.....	31
3.2	Create partition scheme.....	32
3.3	Create partition objects (and send to 4GD).....	33
<b>4</b>	<b>Line Designer: Create Line Designer workset, subset and Design Elements.....</b>	<b>37</b>
4.1	Create a Line Designer workset.....	37
4.2	Create a Line Designer subset and add partitions to recipe.....	38
4.3	Add 2 Line Designer conveyors.....	40
<b>5</b>	<b>Create Automation Designer workset and Engineering Objects.....</b>	<b>43</b>
5.1	Create project workset (and Collaborative Design + subset).....	43
5.2	Create Engineering Object Definitions.....	44
5.3	Create Engineering Object names.....	46
5.4	Add Engineering Objects.....	47
5.5	Location-Product aspects.....	49
<b>6</b>	<b>Map Line Designer and Automation Designer.....</b>	<b>53</b>
6.1	Link Automation Designer and Line Designer Collaborative Designs.....	53
6.2	Manage type mapping.....	54
6.3	Manage object mapping.....	55
<b>7</b>	<b>Configure (non-template) EPLAN.....</b>	<b>61</b>
7.1	Import EPLAN project template.....	61
7.2	Add PM250D macro.....	61
7.3	Modify ELPAN template.....	63
7.4	Modify EPLAN macro.....	64

7.5	Generate.....	66
<b>8</b>	<b>Configure (non-template) TIA Portal.....</b>	<b>69</b>
8.1	Receive hardware, software.....	69
8.2	Place the function blocks in aspects and create IDBs.....	71
8.3	Add tags.....	74
8.4	Create TL constant value.....	79
8.5	Dynamize software.....	81
8.6	Assign software to hardware.....	89
8.7	Send data to TIA Portal.....	91
<b>9</b>	<b>Template-related concepts.....</b>	<b>95</b>
9.1	Getting aspect chain ID of parent EO using expressions.....	95
9.2	Getting aspect chain ID of non-parent: 2 Ports + link.....	96
9.3	Getting aspect chain ID outside template with dynamic connection (software only).....	98
9.4	Automatic generation of calls for inserted software.....	99
<b>10</b>	<b>Configure template-ready EPLAN.....</b>	<b>103</b>
10.1	Function expression.....	103
10.2	KF01.name ports, link, and expression.....	105
10.3	Generate.....	109
<b>11</b>	<b>Configure template-ready TIA Portal.....</b>	<b>111</b>
11.1	Configure symbolic names.....	111
11.2	FRGEStop dynamic connection.....	116
11.3	OB Main calls.....	119
11.4	Generate.....	121
<b>12</b>	<b>Create/instantiate template.....</b>	<b>123</b>
12.1	Create template.....	123
12.2	Insert template.....	126
12.3	Synchronize changes.....	127

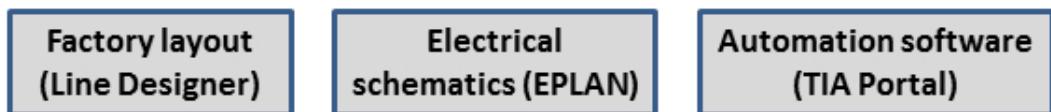
# Concepts

## 1.1 Traditional engineering versus integrated engineering

### Traditional engineering

Traditional engineering workflows include

1. Factory layout and line design using Line Designer to design a production line.
2. Electrical engineering using EPLAN to generate schematics for the production line.
3. Automation using TIA Portal to generate PLC software and tags for specific PLC hardware.

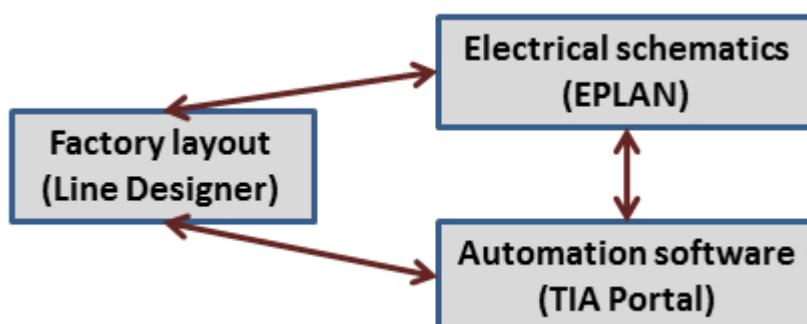


The tools are not linked, leading to the following problems:

1. The Line Designer, EPLAN, and TIA Portal designers must manually synchronize their configurations.
2. TIA Portal software and tag names have no relationship to EPLAN schematic variables.
3. EPLAN and TIA Portal components that repeat (such as conveyors) must be created individually.

### Integrated engineering with Automation Designer

Automation Designer solves the problems above by linking to the above tools to provide centralized functional automation engineering.02d\_02

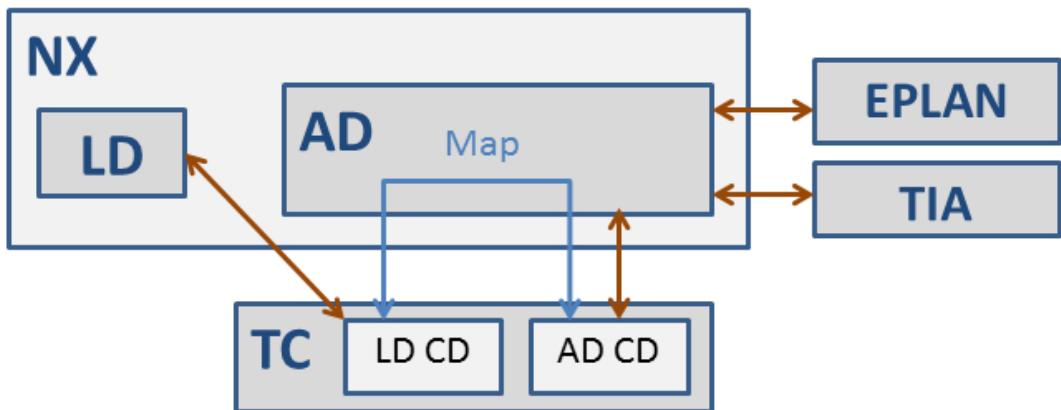


## 1.1 Traditional engineering versus integrated engineering

The Automation Designer solution allows you to

1. Easily determine when the Line Designer configuration is not synchronized with the Automation Designer configuration used to generate EPLAN and TIA Portal.
2. Derive EPLAN macro variables and TIA Portal software and tag names from the same source (from the Engineering Object aspect chain in Automation Designer).
3. Use templates to quickly create TIA Portal software and EPLAN reports for common components (conveyors in this Getting Started). Instantiated templates are automatically assigned unique names as specified in naming rules.

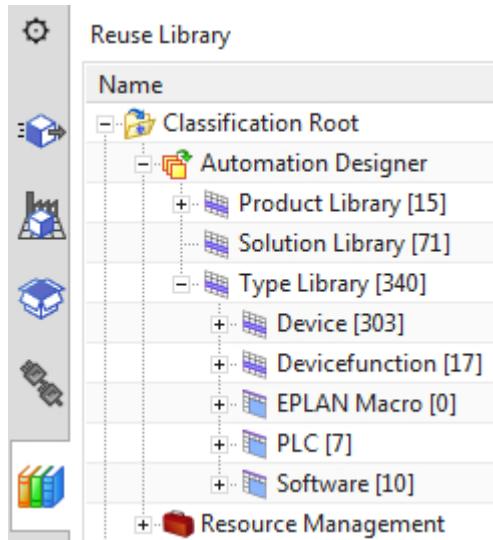
The following diagram shows in more detail how Automation Designer based on NX serves as the central development tool for the entire project lifecycle for mechanics (Line Designer), electrical (EPLAN) and automation (TIA Portal). The two TeamCenter Collaborative Designs are the central project databases for Line Designer and EPLAN/TIA Portal. You can link (map) Line Designer Collaborative Design Design Elements and Automation Designer Collaborative Design Engineering Objects (Design Elements are described in the next section). In this Getting Started the linked Design Element and Engineering Object represent a conveyor.



## 1.2

## Reuse library

Automation Designer distinguishes between library objects and project objects. Project objects are Engineering Objects.

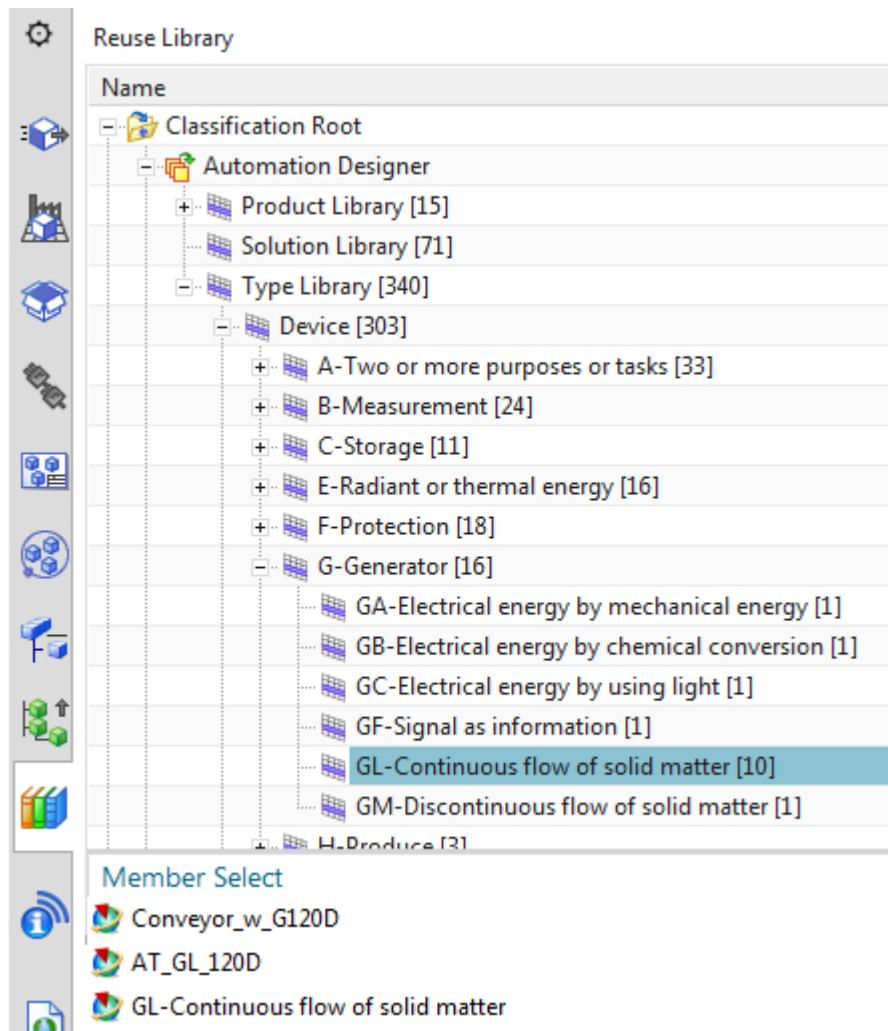


The Reuse Library provides the following objects.

- Types  
*Types* are prototypes for Engineering Objects. They are made available in the Type Library, where they are structured based on their classification classes. Objects with the same Type have the same characteristics.  
Availability in the Reuse Library: Classification Root→Automation Designer→Type Library→Device or Devicefunction
- Products  
*Products* are purchasable devices from a manufacturer. They have an article number. Library administrators can import products from catalogs.  
Availability in the Reuse Library: Classification Root→Automation Designer→Product Library
- Template Solutions  
Templates are reusable solutions that consist of several preconfigured objects. Every library object has a Classification Class.

### Engineering Objects

The following diagram shows Engineering Objects in the Reuse Library.



Engineering Objects are project objects. *Engineering Objects* are the physical and conceptual objects with which you carry out the electrical and automation engineering of a production system or machine in your projects.

To implement a machine or production system, you need the following Engineering Objects:

- Devices and device functions  
For example conveyors, motors, frequency converters, sensors, and signal converters.
- Objects for structuring the system  
For example a line, station, or building.
- EPLAN macros for preparing the generation of electrical schematics
- Program blocks whose code controls the devices and device functions

An Engineering Object can be general or very specific, depending on the number and quality of properties defined by its Type. The more details were provided, the easier it is for you to select a suitable product for a device or devicefunction from the Product Library.

### Relation between Classification Class, Type, and Engineering Object

*Classification Classes* represent the classes and subclasses of the objects that you need for your engineering as proposed by the IEC 81346 standard, part 2. These classes have a purpose- or task-related view of the objects. By creating Naming Rules, you can associate a character code to each Classification Class, to be used for the Engineering Objects' reference designations.

When library administrators create a Type, they must specify the Type's Classification Class. The class defines which properties the Type has by default. The library administrator can add further properties.

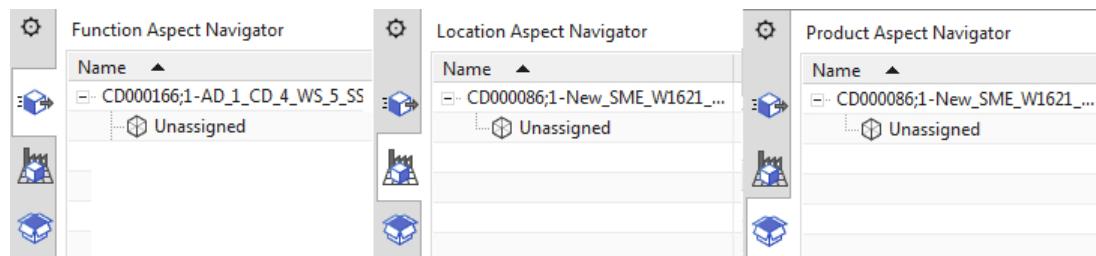
Project engineers can use each Type in their projects as many times as required, by inserting Engineering Objects with this Type. The Engineering Object will have the following data and defaults, as defined by its Type:

- Properties
- Ports
- Default aspects
- Which Line Designer should be used for this Type in Automation Designer (multidisciplinary type mapping)
- Settings for a label which appears in the graphics window if the object is mapped to an Line Designer object.

Project engineers can edit the properties and ports that an Engineering Object takes over from its Type, and they can add new ones. They can add or remove aspects, and, for objects with a Line Designer type mapping, map the object to a Line Designer object with a different type.

## Aspects

The following diagram shows the the 3 aspects.



The IEC 81346 standard describes principles for structuring and naming objects and their associated information in industrial systems, installations, equipment, and industrial products. The goal of these principles is to handle the large sets of information that are available in these systems efficiently. Aspects are a central part of these principles.

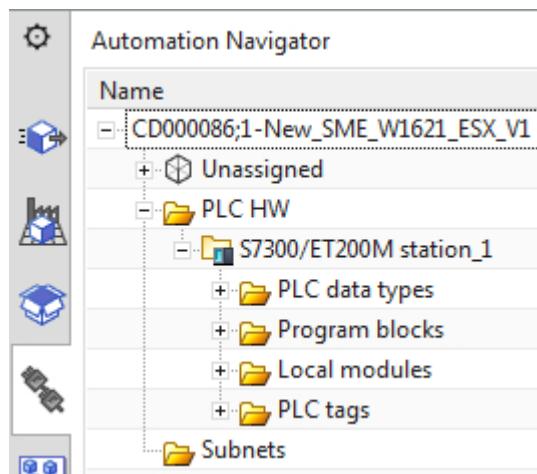
When you automate a production system or machine, you can look at the system or machine from different perspectives. Every engineering task requires a specific view of the machine or production system, of the physical and conceptual objects it takes to realize the machine or production system, and of the relations between these objects.

IEC 81346, part 1, calls these different views *aspects*. It defines the following aspects:

- Functional aspect: What is the functional purpose of an object within the production system or machine or what does the object actually do?
- Location aspect: Where in the production system or machine is the object installed, builtin, or placed, and is it in itself an installation place for other objects?
- Product aspect: Which products are needed and must be ordered to implement the intended function? What are their constructional relations?

You can view the same object under one or more aspects. For each aspect, you consider only those features and relations that are relevant for that specific aspect. The following diagram illustrates this, using the example of a programmable logic controller (PLC).

#### AD automation tab



The **Automation Navigator** is not an Aspect Navigator. It represents the TIA Portal view on the control-related hardware devices, tags, and program blocks of your production system or machine. Its structure is based on the TIA Portal structure and its object tree displays the same names as in TIA Portal.

When you import a hardware device from TIA Portal, it is initially available only in the **Automation Navigator**. By placing a product for the hardware device, you create an Engineering Object for the hardware device. Automation Designer links the hardware device and the Engineering Object, so that the hardware device is also available in the Aspect Navigators. When you select the hardware device, its Engineering Object is selected in all the Aspect Navigators in which it is available.

Program blocks that you imported to the **Automation Navigator** and tags that were created by importing their hardware devices or program blocks to the **Automation Navigator** are initially also available only in the **Automation Navigator**.

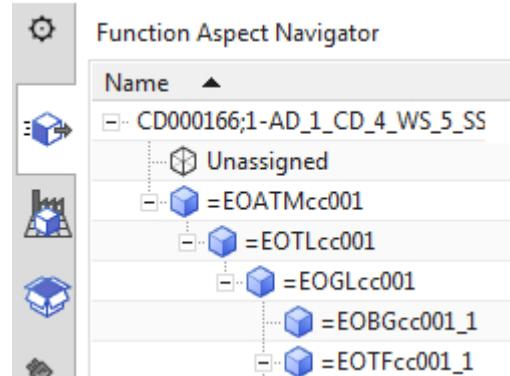
- If a program block is related to a specific Engineering Object, you can place it in one of the aspects that this Engineering Object has. It is then available in the corresponding Aspect Navigator.
- If a tag is related to a specific Engineering Object and you assign the tag to the Engineering Object, the tag is available in the same Aspect Navigators as that Engineering Object.

The project root of the Automation Navigator has the following structure:

- The Unassigned folder collects unassigned tags and program blocks.
- The PLC HW folder collects all the PLC stations of the project. Every PLC station has the following nodes:
  - The PLC modules folder collects the PLC station components. It contains the I/O modules of the PLC station, decentralized stations, or field devices. The channels are under their I/O module.
  - The PLC tags folder collects the tags that belong to the PLC station's control scope.
  - The Program blocks folder collects the program blocks that belong to the PLC station's control scope.

### Engineering Objects in aspects

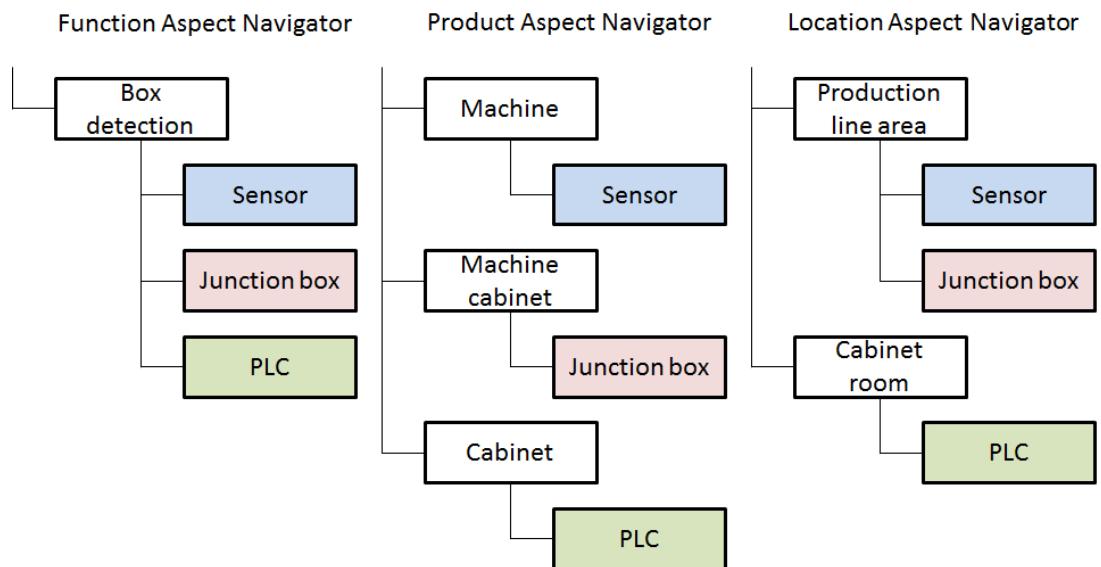
The following diagram shows the Function Aspect Navigator with Engineering Objects. The below configuration reflects the structure of the plant and Line Designer elements and is used to create symbolic names.



Each Engineering Object can have more than one aspect. If an Engineering Object has an aspect, it is visible in the corresponding Aspect Navigator. If an Engineering Object has several aspects, the same object is visible in more than one Aspect Navigator.

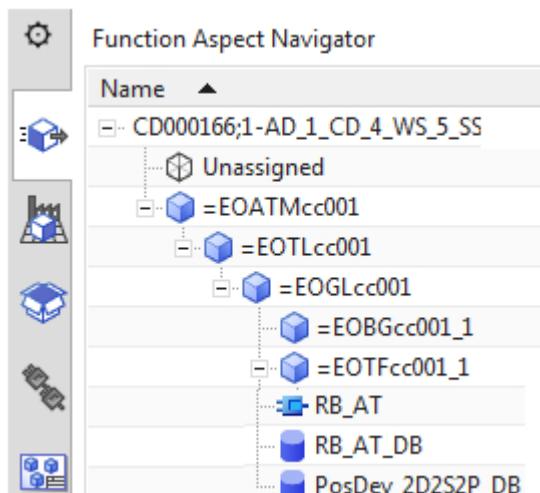
Because the hierarchical structure of objects in an Aspect Navigator depends on the aspect, the Engineering Object can have different parents and different children in each Aspect Navigator. This means that the hierarchical structures of objects in the Aspect Navigators are independent of each other.

The following example illustrates a sensor monitoring the movement of packaging boxes on a conveyor. When the sensor detects a box, it sends a signal to its PLC. The sensor and PLC are wired through a junction box. For each of these components there is one Engineering Object that has a function, location, and product aspect. In each Aspect Navigator, the objects have different parents and siblings.



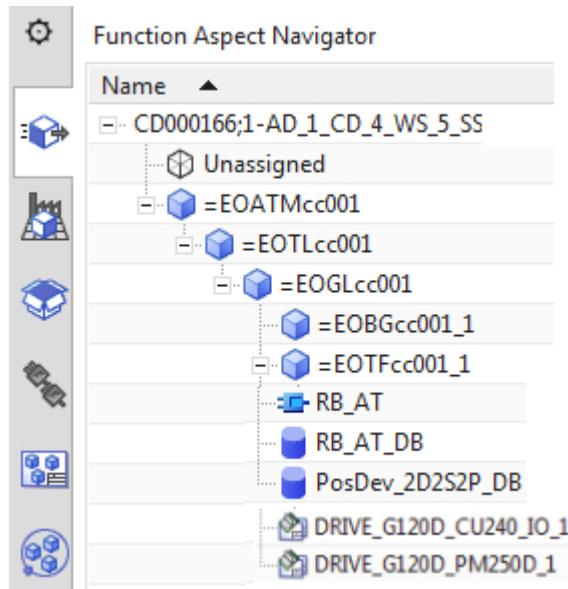
### TIA Portal software in aspects

Software placed in the aspects under an Engineering Object. This aspect chain above this Engineering Object is used to determine the symbolic name (unique identifier) of the Engineering Object. This name is then used in TIA Portal software (using symbolic names for software and tag names). The software is later assigned to hardware in the Automation tab and then sent to the TIA Portal.



### Macros in aspects

Software and macros are placed in the aspects under an Engineering Object. This aspect chain above this Engineering Object is used to determine the symbolic name (unique identifier) of the Engineering Object. This name is then used in EPLAN reports (using symbolic names for variables).



### Ports and links

Sometimes the required symbolic reference does not belong to the parent Engineering Object of the software or macro, so you must create in the parent Engineering Object a link via a port between the parent and target Engineering Objects using ports.

Objects have vertical relations and horizontal relations to other objects.

- *Vertical relations* define parent-child relations in the Aspect Navigators.
- *Horizontal relations* are connections between ports. They do not define parent-child relations. They can connect objects from different navigators or in the same navigator.

*Ports* are a means of connecting objects. They are available for Engineering Objects and tags. You connect the port of one object, the source port, to the port of another object, the target port. This creates a bidirectional connection. *Connections* represent port-based relations between objects.

Every port belongs to the object for which it was created. It is an integral part of that object. Every port has a port type, connection type, direction, and cardinality. The port type determines which connection type the port can have. Automation Designer allows you to connect only ports with compatible settings.

If an object has a port connection, you can use navigation expressions to navigate to the connected port. Then you can use navigation expressions to access the following data of the connected object:

- Its properties
- Its ancestors and descendants in the Aspect Navigators
- Its port connections

You have recursive access to the properties, ancestors, descendants, and port connections of further objects.

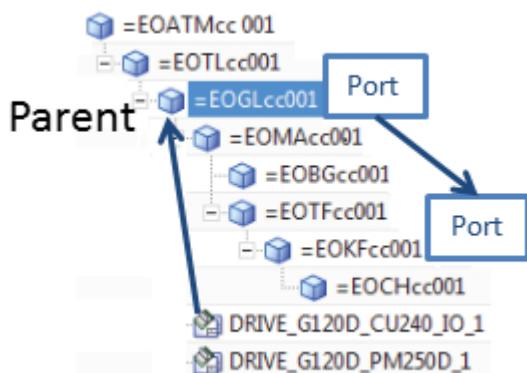
*System-defined ports* are automatically created by Automation Designer, either when you insert an object in a project, or when you carry out actions that internally require port

connections. You can connect or disconnect system-defined ports. You cannot create or delete them, or edit their settings.

*User-defined ports* are created by users. You can create, edit, connect, disconnect, and delete user-defined ports.

Tags have only system-defined ports. You cannot create ports at tags. Engineering Objects have system-defined and user-defined ports.

The solution is shown in the following diagram. The macros access Engineering Object KF using a link between a port on the parent GL and a KF port.



To configure this you do the following:

1. Create a port for parent Engineering Object GL.
2. Create a port for target Engineering Object KF.

### Expressions

An *expression* is a formula that returns a value. The value can be of raw type, an object, or a list of objects or raw type values. The expression formula can consist of function calls, variables, numbers, operators, and symbols. Automation Designer extends NX functions with navigation functions. You use expressions to do the following:

- To set a property value. You can use navigation expressions or an expression that creates an object reference.
- To create a dynamic connection between objects. Use navigation expressions.
- In program blocks, to create dynamic connections for operand ports, caller ports, or method ports, and to define conditions for inserting calls, methods, or replacing operands.

A *dynamic connection* is a connection that you link to a navigation expression. The navigation expression returns the target port.

### Templates

*Templates* are reusable, ready-made solutions that reduce the complexity of engineering decisions to choosing between prepared solutions. They allow you to take an engineering solution from one project and to reuse it in the same project and in other projects.

A template consists of all objects and aspects that are required to implement this solution, including EPLAN macros, program blocks, and tags. It defines the property values of these

objects and their relations, both within an aspect and between aspects. If required, templates can use expressions to define the properties and relations.

Templates exist in the Solution Library and in the projects in which they are reused. We use the term *template* if it is clear from the context whether template refers to a template in the library or to a template in a project. Else we use the terms *template definition* and *template usage*.

A *template definition* is the blueprint for an engineering solution that project engineers want to reuse in their projects. Template engineers create template definitions in the template environment, where they build the content of the template. This process is called template creation.

The template definitions are made available to the project engineers in the Solution Library and in the Type Library. In the projects, the project engineers can use each template as many times as required.

In Automation Designer, you can use rules and expressions to define or change the property values, tags, and relations of the hardware configuration, EPLAN macros, program blocks, and other objects that are members of a template. If the project environment changes, these property values, tags, and connections adapt to the changes and are updated automatically. This process increases the reusability of the templates.

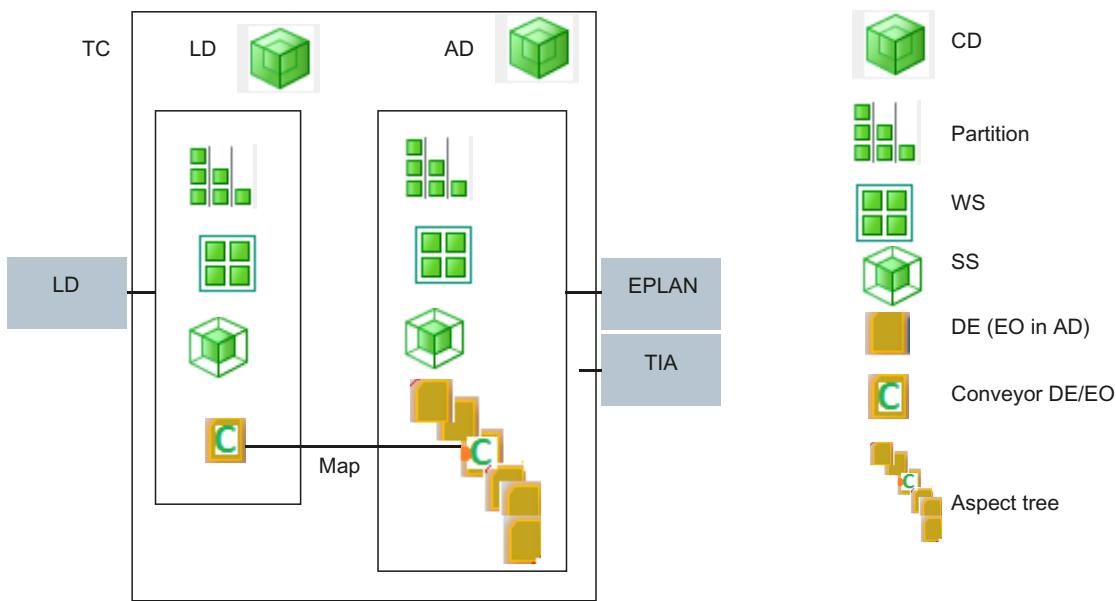
## 1.3 TeamCenter (4GD) details

This section provides a short introduction to the following 4GD components :

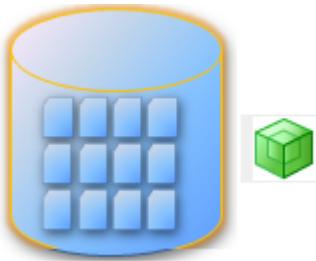
- Collaborative Design
- Partition scheme
- Partition
- Workset
- Subset
- Design element

## Concepts

### 1.3 TeamCenter (4GD) details



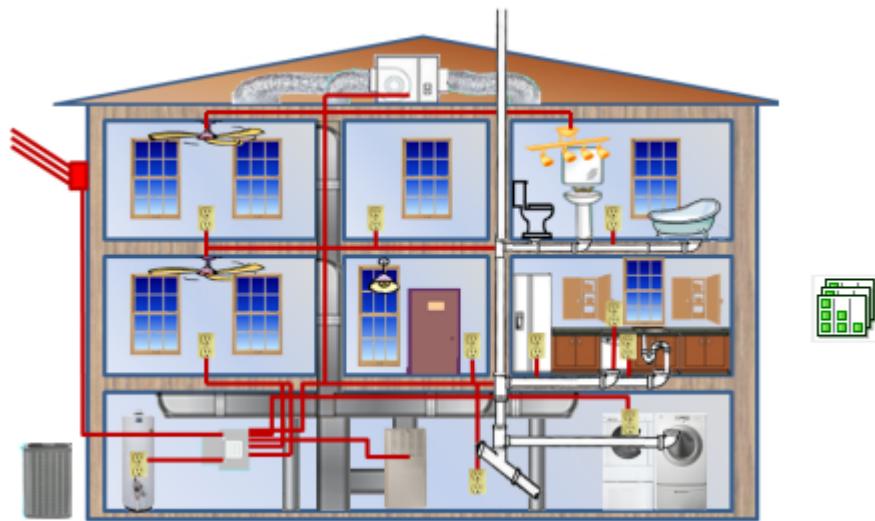
**CD**



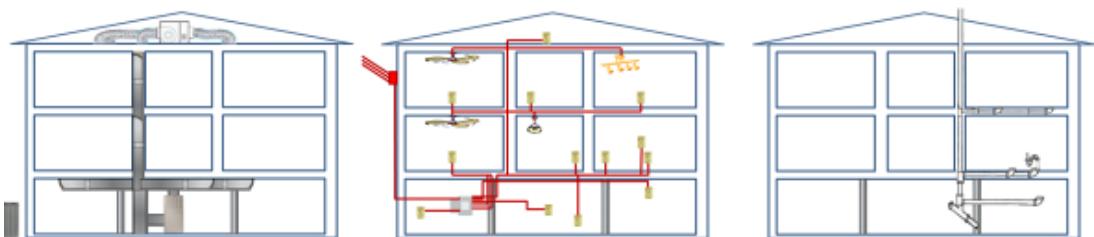
A collaborative design is a model of a project/product that is developed by a team of contributors. The elements of the model are arranged in a hierarchy that allows team members to collaborate and author common project/product information in an efficient manner. A CD is the container object in TC of all the design data that defines a product or a class of products.

#### Partition scheme

Partition schemes can be functional, spatial, or physical. Partitions are created within partition schemes. For example, in a 4GD design of this house, different types of partition schemes can be used to organize the design elements.



Functional: A functional partition scheme could contain partitions for the HVAC (heating, ventilation and conditioning), electrical, and plumbing systems.

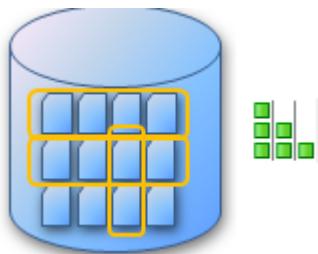


Spatial: A spatial partition scheme could contain partitions for each floor. By default, spatial partitions are defined by a recipe so that new design elements are automatically added to the partition.

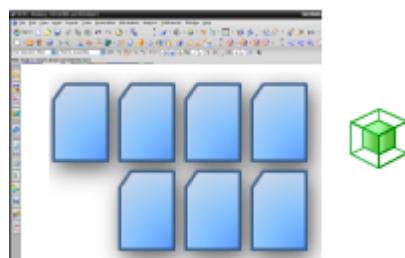


Physical: A physical partition scheme could contain partitions organizing each individual physical room.



**Partition**

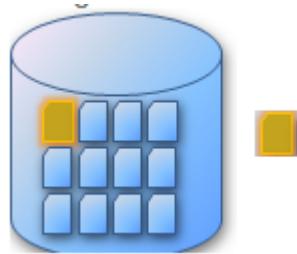
A partition object is a container for data. Partitions can be organized in by a partition scheme (such as function, spatial, or physical as listed above). Unlike traditional subassemblies, partitions do not control the position or any other property of a DE. DE's can be placed in multiple partitions. For example, in a CD of a house, a section of pipe might be part of a plumbing partition and part of the kitchen partition. Partitions can be static, requiring manual addition of DEs, or dynamic, where the partition contents are defined by search criteria.

**Workset**

A workset object is the collection of DEs in your NX session. A workset is defined by one or more subsets. There may be many DEs within the workset you work on in your NX session.

**Subset**

A subset object selects the design elements for a workset. The subset may include specific DEs, or it may contain a dynamic recipe which defines partitions to search. The diagram above shows a session with 2 subsets.

**Design element**

A design element object is a representation of a component in the product. It is a unique occurrence of 3D geometry in a specific location in the product design. There are different types of DEs. A DE can reference an NX part or assembly model, or other types of geometry.



# Overview of this Getting Started

## 2.1 Prerequisites

This Getting Started assumes you have the following already configured:

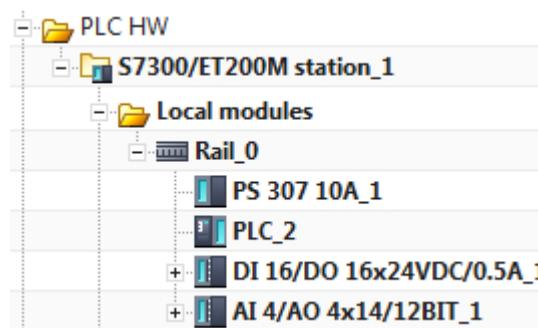
1. NX, TeamCenter and Line Designer with

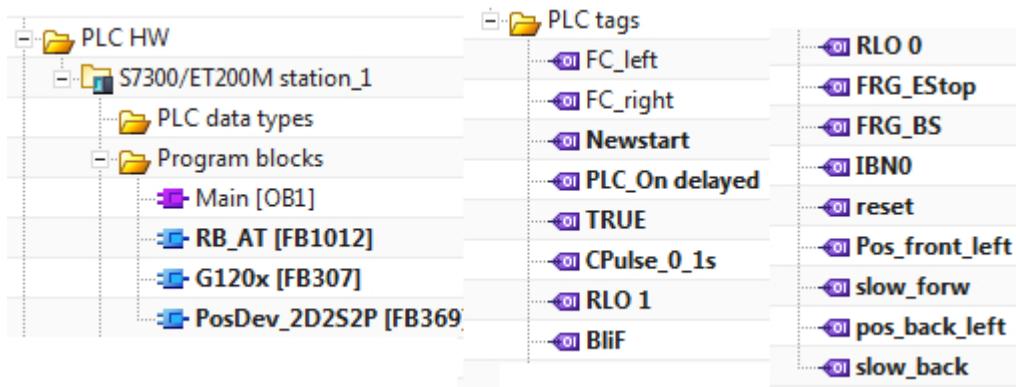
- A conveyor.
- The following in the reuse library Classification Root
  - Device / A ->1 purpose or task / AT
  - Device / U-Keep
  - Device / G-Generator / GL-Continuous flow
  - Device / M-Motor / MA-Electromagnetic
  - Device / B-Measurement / BG-Gauge,position
  - Device / T-Conversion / TF-Signals
  - Device / K-Processing / KF-Electrical signals
  - Devicefunction / Electrical / Input/output

2. EPLAN with:

- Template IEC\_bas001.zw9.
- Macro DRIVE\_G120D\_PM250D\_1.emp

3. TIA Portal with the following hardware and software:



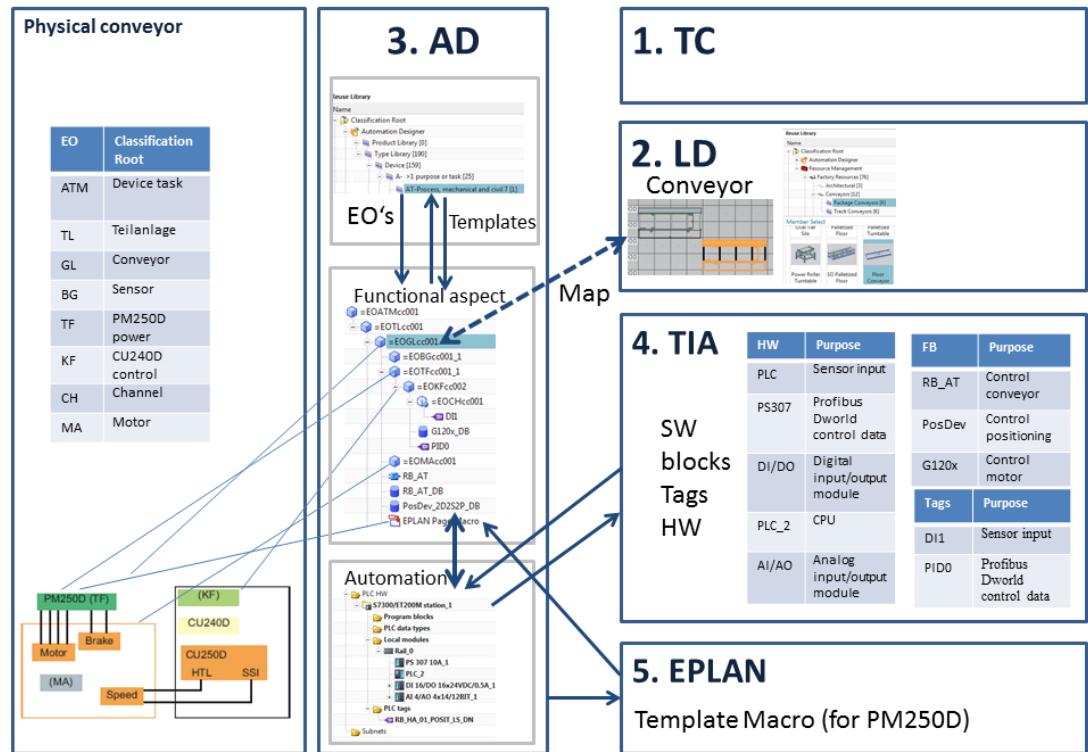


## 2.2

## Workflow from the tools perspective

The following diagram shows what you do in this Getting Started from the perspective of the tools used.

1. TeamCenter: Create a Line Designer 4GD Collaborative Design.
2. Line Designer: Create a 4GD workset, subset, and add two conveyor Design Elements.
3. Automation Designer: Create an Automation Designer workset, Collaborative Design, and subset. Model the plant equipment by adding Engineering Objects from the Reuse Library to the Function aspect. Map the Line Designer Design Element for the conveyor to the Automation Designer Engineering Object for the conveyor.
4. Automation Designer/TIA Portal: Import hardware, software FB blocks and tags from TIA Portal into Automation Designer, placing them in the Automation tab or the Function aspect. Copy software/tags to the aspects, dynamize, and export to TIA Portal. Make the software/tags template-ready, then create and instantiate the template.
5. Automation Designer/EPLAN: Import the EPLAN template and all required macros into Automation Designer in the aspects, set EPLAN variables and generate EPLAN reports. Make the macros template-ready, then create and instantiate a template.



## 2.3 Workflow

The following describes what you do in this Getting Started.

The workflow can be organized into three parts:

Part 1: Create Line Designer/Automation Designer mechatronic models

Part 2: Mapping Line Designer-Automation Designer, generating EPLAN and TIA Portal (without templates)

Part 3: Generating EPLAN and TIA Portal with templates

### Part 1: Create Line Designer/Automation Designer mechatronic models

Automation Designer is based on the NX framework and uses Teamcenter as the data backbone. This allows seamless data exchange from Line Designer, another NX-based and Teamcenter-based solution.

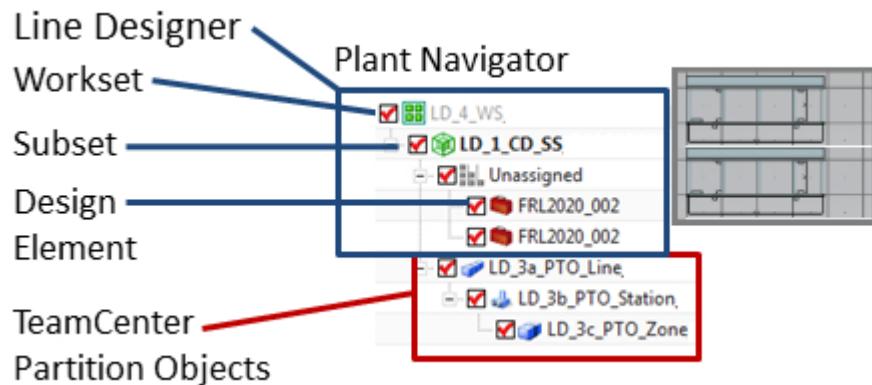
Line Designer objects are an integral part of the Automation Designer system design. The automation engineer sees the line design as the mechanical engineer sees it.

Objects from the Line Designer layout are used in Automation Designer during system design and further enriched during electrical and automation engineering.

### Create the Line Designer Collaborative Design, Line Designer workset, subset and Design Elements

A project is the container that stores the objects you need to carry out the electrical and automation engineering for a production system or machine. In Automation Designer this container is called *project*, in Teamcenter it is called *collaborative design object*. For every Automation Designer project there is one collaborative design object in Teamcenter.

- (1) In TeamCenter create the LC Collaborative Design (partition).
- (2) In Line Designer create the Line Designer workset, subsets.
- (3) In Line Designer add 2 conveyor Design Elements.



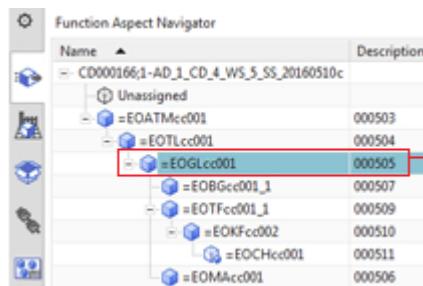
### Create the Automation Designer workset, subset and Engineering Objects

To work in a project, you need a workset. The *workset* is your work context and gives you access to the data that was saved in the workset's project. In a workset, you do the following:

- Add new data to the project.
- Access data that other users added in worksets belonging to the same project.
- Edit the existing data.

When you create a new project, Automation Designer automatically creates a workset for the project. If you have access rights for an existing project, you can also create new worksets for this project. You must create a workset to be able to work in that project.

- (1) In Automation Designer create the workset (this automatically creates the TeamCenter Collaborative Design and subset).
- (2) In Automation Designer create the Engineering Object aspect tree that models line components (GL is the conveyor). Engineering Object GL corresponds to the conveyor Design Element in the Line Designer Collaborative Design.



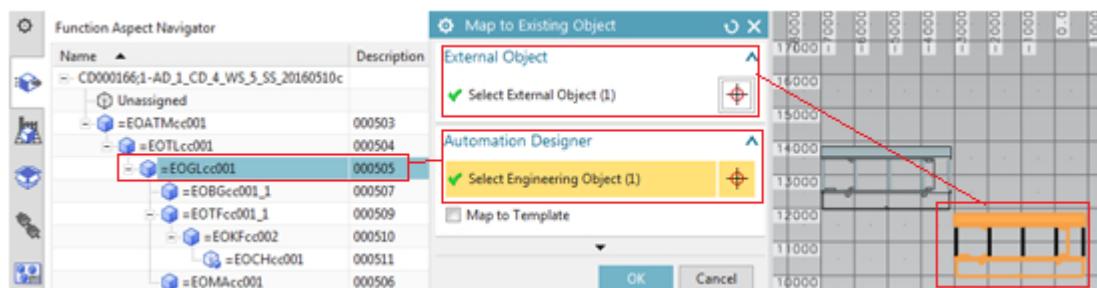
## Part 2: Mapping Line Designer-Automation Designer, generating EPLAN and TIA Portal (without templates)

### Map Line Designer-Automation Designer

You can map external object types from the Reuse library to Types or to template definitions from the Reuse Library.

If you want to map external objects to new Engineering Objects, you can use the type mappings that were defined as default for all projects.

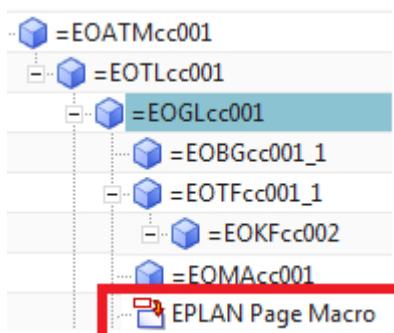
Map the the Line Designer conveyor to the Automation Designer conveyor Engineering Object. This allows you to track changes in Automation Designer and Line Designer.



### Configure a basic AD project for EPLAN.

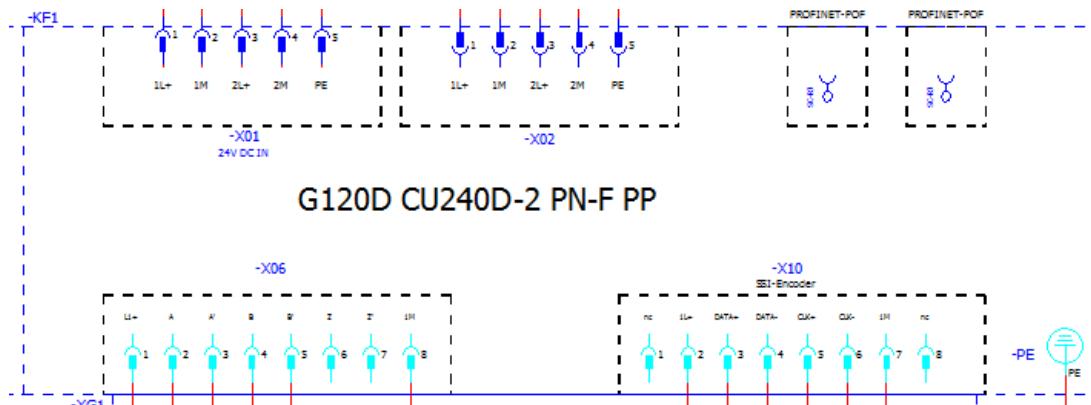
Using adaptive ECAD templates and the information provided during electrical engineering, Automation Designer uses the EPLAN Electric P8 API to generate an ECAD project for the automation system in EPLAN Electric P8.

(1) Import an EPLAN template and macro into Automation Designer. You can import macros into a project and use them directly or import them into a template and reuse the template.



## 2.3 Workflow

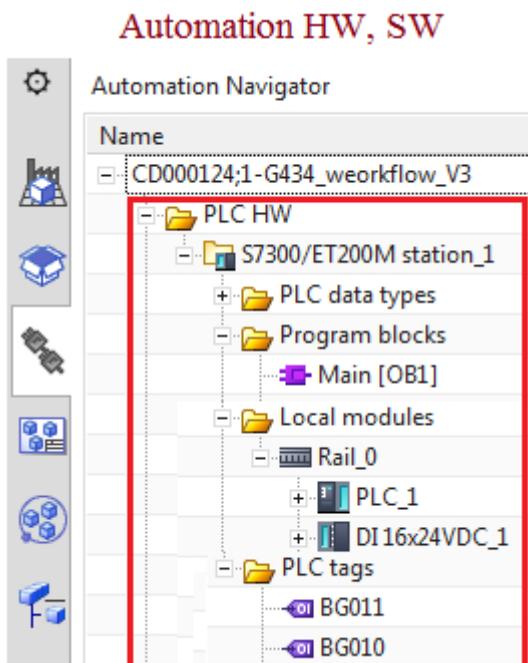
(2) Generate a report with default values. Use the EPLAN project template and EPLAN macros to generate an EPLAN project with electrical schematics for the automation system.



### Configure a basic AD project for TIA Portal.

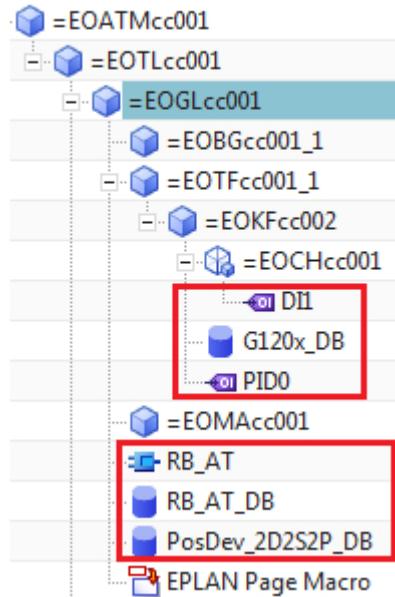
Automation Designer has a connection to TIA Portal. This connection makes the control hardware available in Automation Designer and allows users to change the hardware configuration through TIA Portal. It is also used to transfer tags and control code from Automation Designer to a TIA Portal project and vice versa. The project can be updated at any time.

(1) Import TIA Portal hardware into the Automation tab..



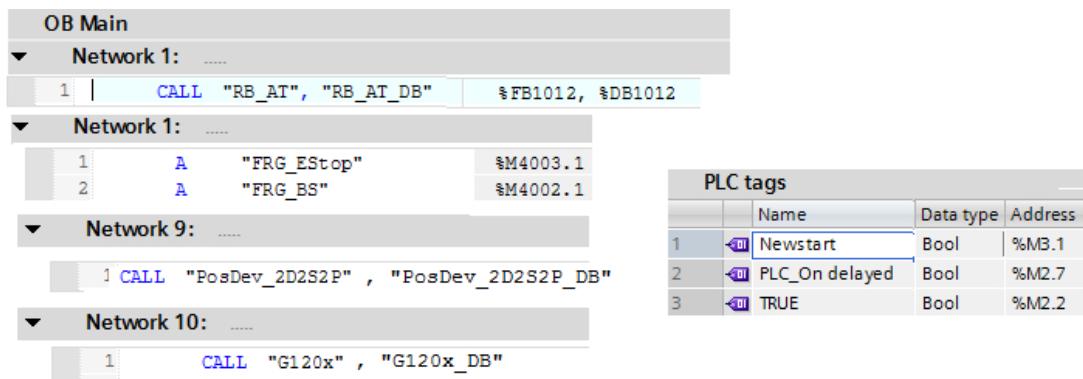
(2) Import software into the Automation tab. Place the software in the aspect tree and create IDB's and tags.

## Aspect tree SW



(3) Dynamize the software. Dynamization of software means that the imported software is enhanced in such a way that it can be used in templates for auto-generation of software.

(4) Generate output to TIA Portal. Transfer the hardware configuration, tags, and PLC program to TIA Portal.



## Part 3: Generating EPLAN and TIA Portal with templates

### Template-related concepts

You can insert objects using templates.

Suppose you need a conveyor that is controlled by a frequency converter for your production system. To implement this conveyor, you need the following Engineering Objects:

- The conveyor, a motor, a frequency converter, sensors, and signal converters
- EPLAN macros for preparing the generation of the electrical schematics for this conveyor
- Program blocks whose code controls the conveyor, motor, frequency converter, and sensors

## 2.3 Workflow

If the library has a template that consists of all the objects that are required for such a conveyor, you can insert the template in your project. Automation Designer then inserts all the objects that belong to this template, and assigns their aspects and parents as defined in the template.

### Configure a template-ready AD project for EPLAN

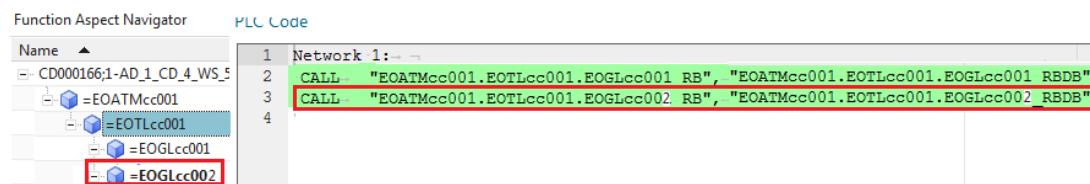
*EPLAN macros* are Engineering Objects for graphical schematic templates. Use them to prepare the electrical schematics of the production equipment or devices used in your production system or machine.

Set the EPLAN variables to a value based on the Function aspect chain. The following shows the result.



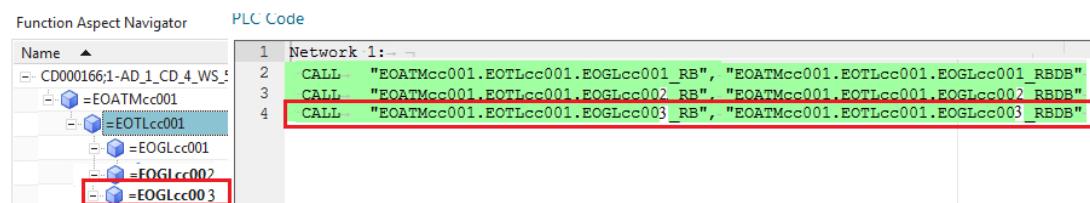
### Configure a template-ready AD project for TIA Portal

Set software block and tag names to a value based on the Function aspect chain. The following shows the result after copying a template-ready conveyor.



### Create/instantiate template

The following shows an instantiated template.

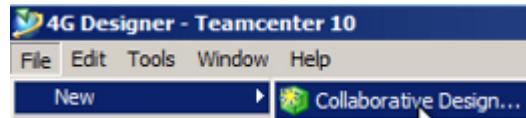


# TeamCenter: Create Line Designer Collaborative Design

3

## 3.1 Create plant design Collaborative Design

1. In 4GD Designer select File→New→Collaborative Design.



2. Select Plant Design.



3. Click Assign.

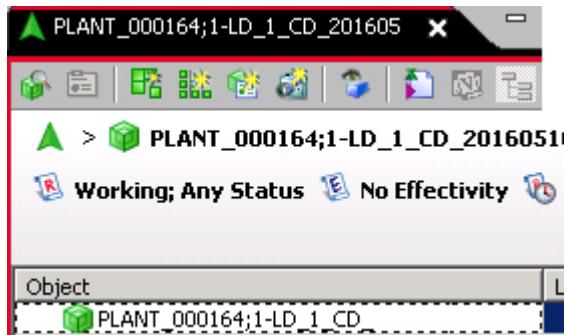
4. For Name enter "LD\_1\_CD" (Line Designer Collaborative Design).



5. Click Finish.

6. Click Close.

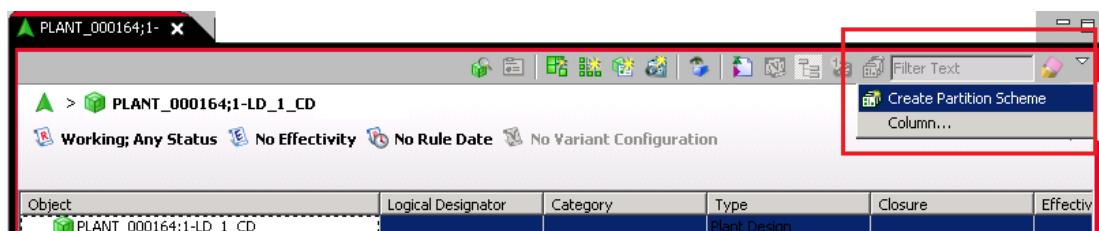
### 3.2 Create partition scheme



## 3.2 Create partition scheme

Partition schemes can be functional, spatial, or physical. Partitions are created within partition schemes . For this Getting Started you create a single partition scheme.

1. Click on **Create Partition Scheme**.



2. Select **Production Unit Scheme**.



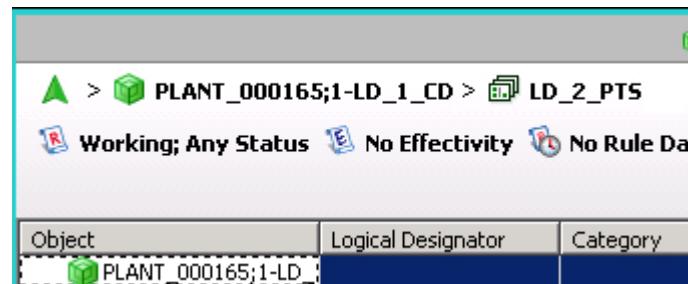
3. Click **Next**.

4. Set **Name = "LD\_2 PTS"** (Line Designer Partition Scheme).



5. Click **Finish**.

6. Click **Close**.



### 3.3

### Create partition objects (and send to 4GD)

A partition object is a container for data. Partitions can be organized in by a partition scheme (such as function, spatial, or physical as listed above). Unlike traditional subassemblies, partitions do not control the position or any other property of a Design Element. Design Elements can be placed in multiple partitions. For example, in a Collaborative Design of a house, a section of pipe might be part of a plumbing partition and part of the kitchen partition. Partitions can be static, requiring manual addition of Design Elements, or dynamic, where the partition contents are defined by search criteria.

Create the partition objects (business objects) line, station and zone.

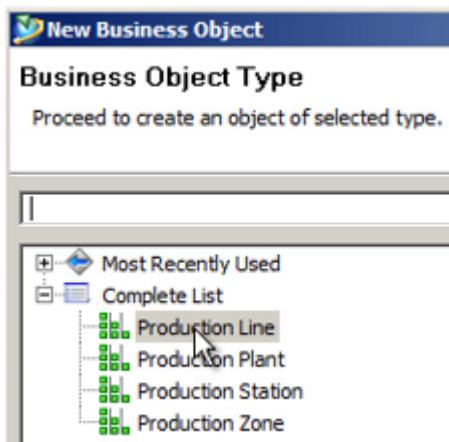
1. Click on **Create partition**.



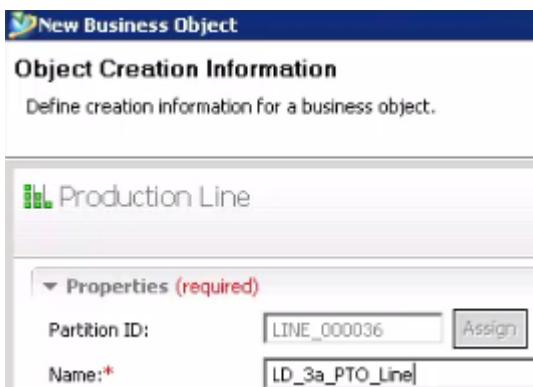
2. Select **Production Line**.

3. Click **Next**.

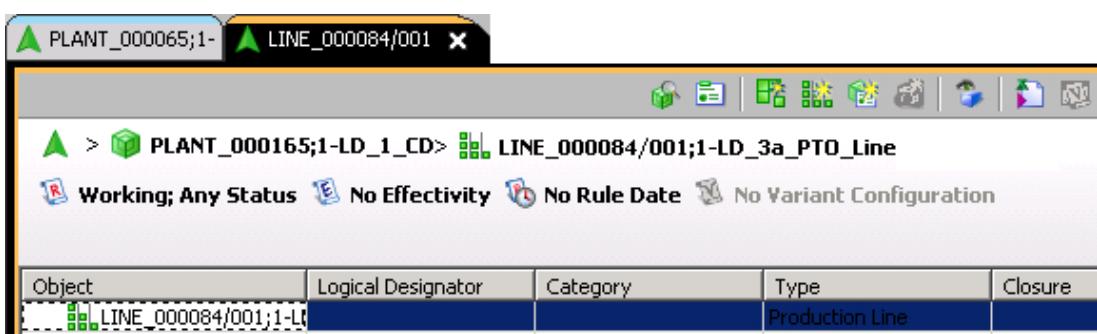
## 3.3 Create partition objects (and send to 4GD)



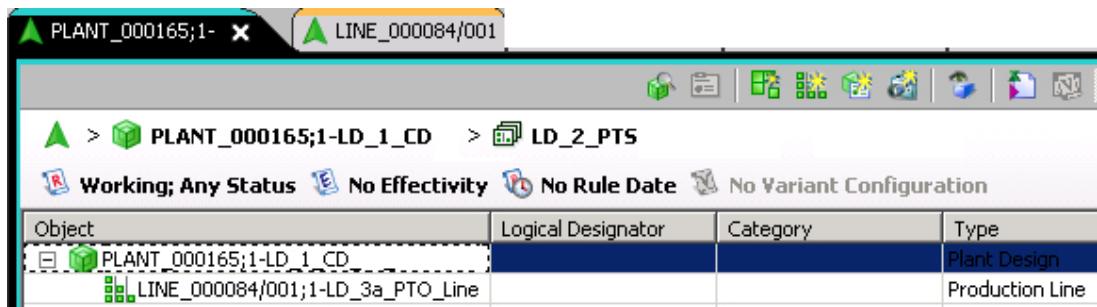
4. Click **Assign**.
5. Set **Name** = "LD\_3a\_PTO\_Line".



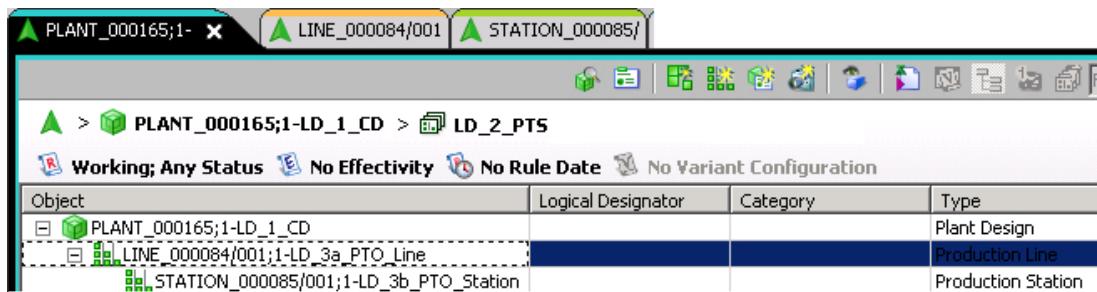
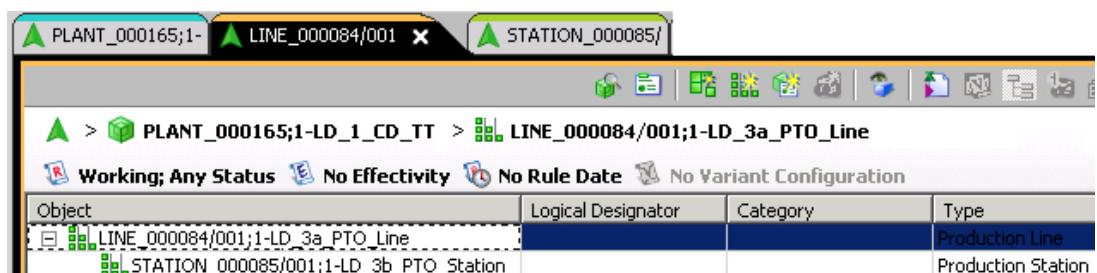
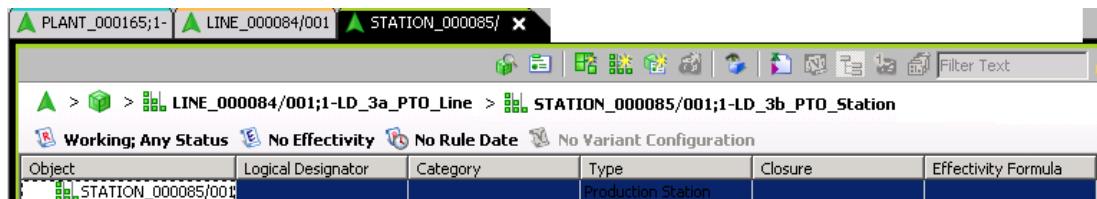
6. Click **Finish**.
7. Click **Close**. The following shows what you have created so far.



## 3.3 Create partition objects (and send to 4GD)

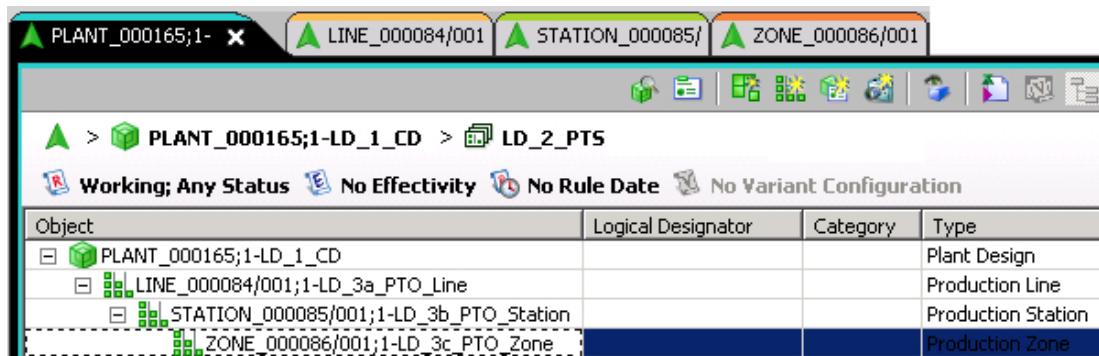


8. Click on the line under Plant in the Plant tab.
9. Click the **Create Partition** Icon.
10. Select **Production Station**.
11. Click **Next**.
12. Click **Assign**.
13. Enter **Name** = "LD\_3b\_PTO\_Station".
14. Click **Finish**.
15. Click **Close**. The following shows what you have created so far.



16. Create a "Production Zone" partition under the station partition with **Name** = "LD\_3c\_PTO\_Zone".

## 3.3 Create partition objects (and send to 4GD)



17. Send to 4GDesigner.

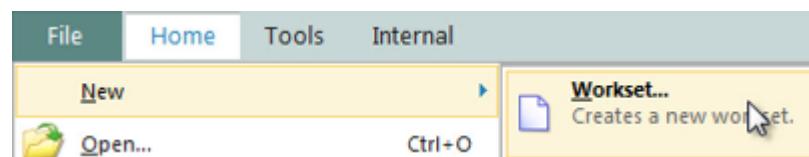


# Line Designer: Create Line Designer workset, subset and Design Elements

## 4.1 Create a Line Designer workset

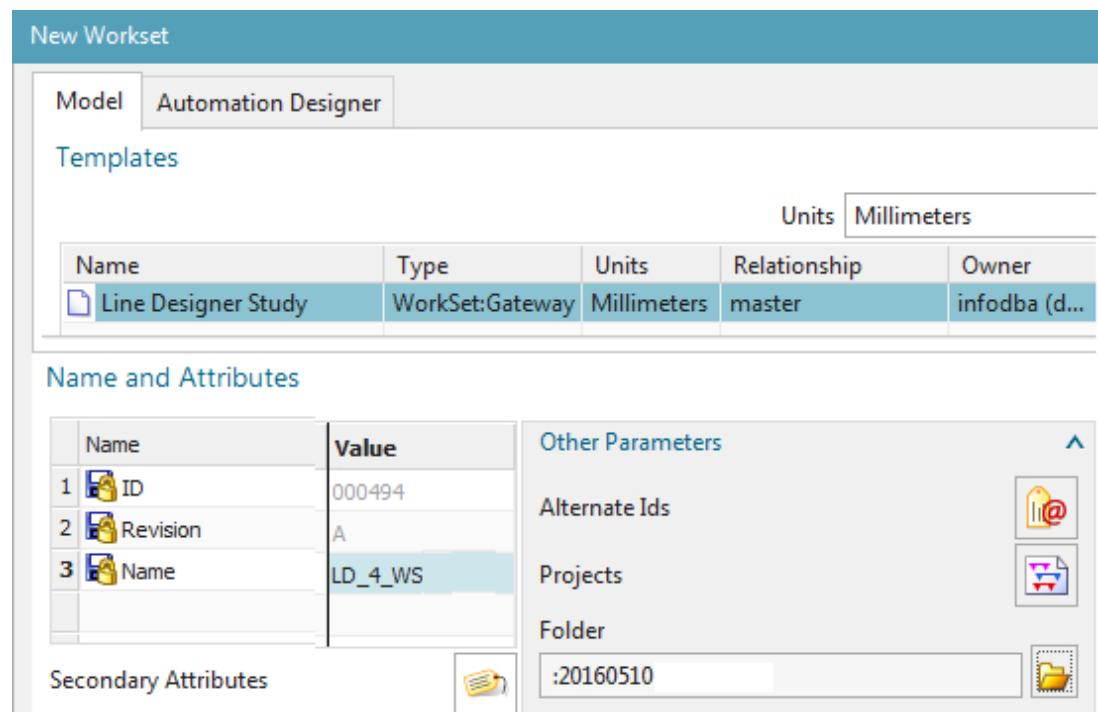
A workset object is the collection of Design Elements in your NX session. A workset is defined by one or more subsets. There may be many Design Elements within the workset you work on in your NX session.

1. Create a new workset.



2. Select Model→Line Designer Study.

3. Set Name = "LD\_4\_WS". If you not specify a folder then the project will be put in "Newstuff".



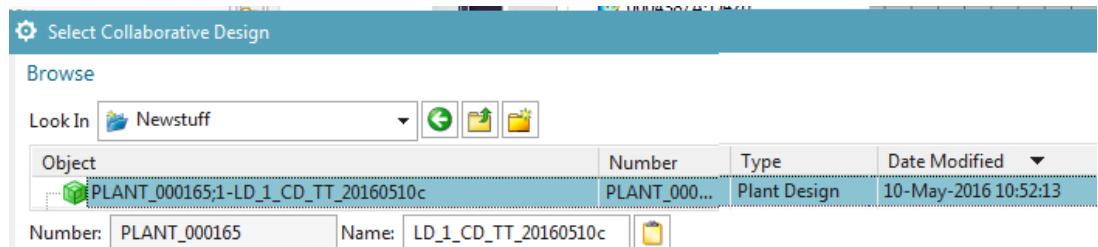
4. Click OK. The "Create Subset" dialog appears.

#### 4.2 Create a Line Designer subset and add partitions to recipe

## 4.2 Create a Line Designer subset and add partitions to recipe

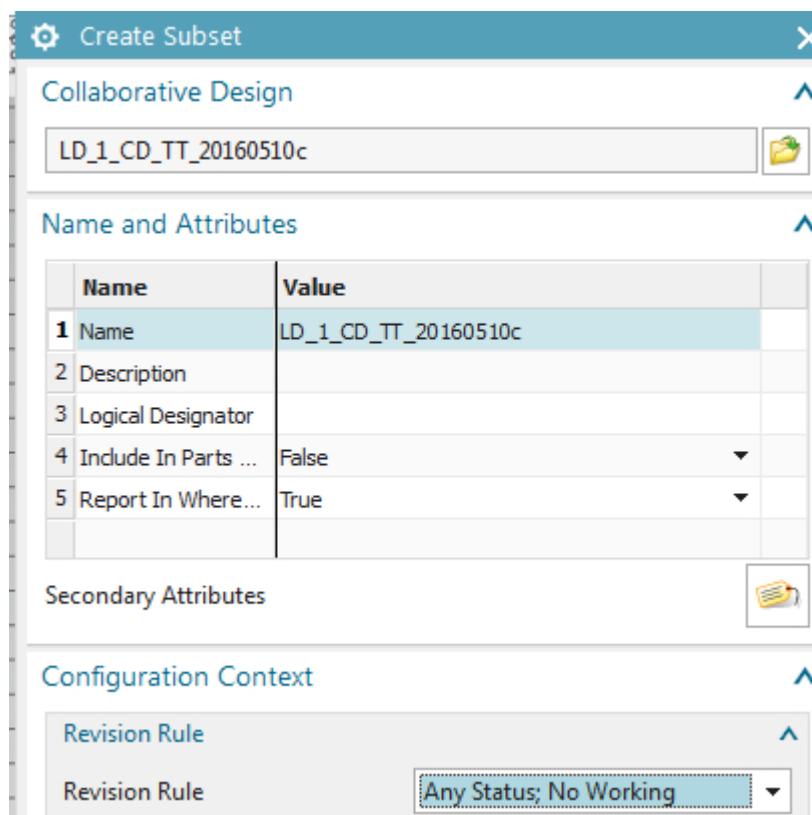
A subset object selects the design elements for a workset. The subset may include specific Design Elements, or it may contain a dynamic recipe which defines partitions to search. The diagram above shows a session with 2 subsets.

1. For **Collaborative Design** select "LD\_1\_CD".



2. Click **OK**.

3. For **Revision Rule** select **Any Status, No Working**.



4. Note that the subset has the same name as the Collaborative Design. To rename the subset, click and type in the new name.

## 4.2 Create a Line Designer subset and add partitions to recipe

## Name and Attributes

Name	Value
1 Name	LD_1_CD_SS_20160510c

5. Click OK. The Subset Definition appears.

The screenshot shows the 'Subset Definition' dialog box. The table lists objects under the 'Object' column, their 'Number' (e.g., PLANT\_000165;1-LD\_1\_CD\_TT\_20160510c), 'Access' (e.g., Production Uni...), 'Type' (e.g., Production Line), and 'D. Name' (e.g., PLANT\_000165;1-LD\_1\_CD\_TT\_201...). The 'View Style' dropdown is set to 'Tree'.

Object	Number	Access	Type	D. Name
PLANT_000165;1-LD_1_CD_TT_20160510c	PLANT_000165;1-LD_1_CD_TT_201...		Production Uni...	PLANT_000165;1-LD_1_CD_TT_201...
LD_2 PTS_20160510c			Production Line	LD_2 PTS_20160510c
LINE_000084/001;1-LD_3a_PTO_Line_20160510c	LINE_000084	+	Production Line	LD_3a_PTO_Line_20160510c
STATION_000085/001;1-LD_3b_PTO_Station_20160510c	STATION_000085	+	Production Stat...	LD_3b_PTO_Station_20160510c
ZONE_000086/001;1-LD_3c_PTO_Zone_20160510c	ZONE_000086	+	Production Zone	LD_3c_PTO_Zone_20160510c

6. Select the tree if not shown as above.

The screenshot shows the 'Content Search' interface with the 'View Style' dropdown set to 'Tree'. Other options include 'Tile' and 'Collaborative'.

7. Select all, right click and select Add to Recipe→Include. This adds the subset to the recipe.

The screenshot shows the subset definition table with a context menu open over the last row (ZONE\_000086). The menu includes 'Add to Recipe', 'Include', 'Exclude', and 'Filter'.

8. Click Finish. The workset and subset are shown. Note that you are in the Gateway.

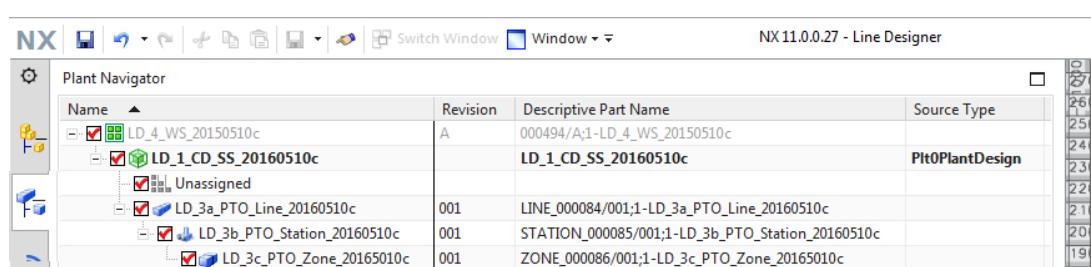
The screenshot shows the 'Gateway' interface. The 'Assembly Navigator' pane shows sections and worksets. A table below lists worksets and subsets, including 'LD\_4\_WS\_20150510c' and 'LD\_1\_CD\_SS\_20160510c'.

Object	Number	Revision	Info	Name	Source	Type
Sections						
000494/A;1-LD_4_WS_20150510c (Order: Chronological)	000494	A		LD_4_WS_20150510c	000494/A;1-LD_4_WS_20150510c	Workset
LD_1_CD_SS_20160510c	LD_1_CD_SS_20160510c			LD_1_CD_SS_20160510c		Subset

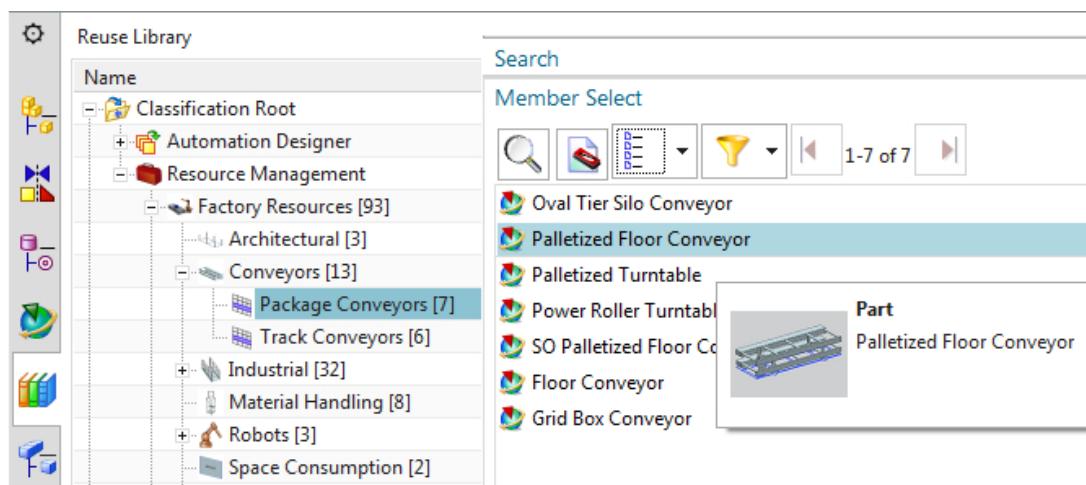
#### 4.3 Add 2 Line Designer conveyors

You now add two conveyors that will be linked (mapped) later to Engineering Objects.

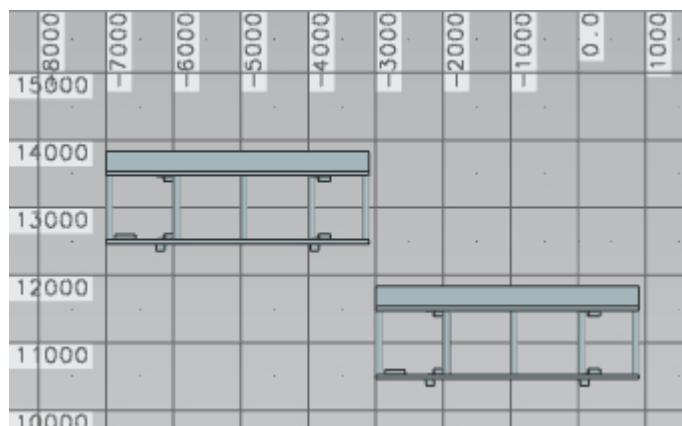
1. Switch to Line Designer. Note the hierarchy under the plant navigator, which shows what you created so far (a workset, subset and the partitions line, station, and zone).



2. Drag and drop 2 conveyors from the Reuse Library.



The following shows the resulting conveyors.



The following shows the conveyors under the subset in the assembly navigator.

## 4.3 Add 2 Line Designer conveyors

Object	Number	Revision	Info	Name	Source	Type	Description	M	Partition ...	Effectivity	Q.	Projects
Sections	000494/A;1-LD_4_WS_20150510c (Order: Chronological)	A		LD_4_WS_20150510c	000494/A;1-LD_4_WS_20150510c	Workset	000494	<input checked="" type="checkbox"/>				
	LD_1_CD_SS_20160510c			LD_1_CD_SS_20160510c	LD_1_CD_SS_20160510c	Subset		<input checked="" type="checkbox"/>	Not Set			
	RES_000081/001;1-FRL2020_002	001		FRL2020_002	000496/A;1	Resource...		<input checked="" type="checkbox"/>		1		
	RES_000083/001;1-FRL2020_002	001		FRL2020_002	000498/A;1	Resource...		<input checked="" type="checkbox"/>		1		

The following shows the conveyors in the plant navigator.

Name	Revision	Descriptive Part Name	Source Type
LD_4_WS_20150510c	A	000494/A;1-LD_4_WS_20150510c	
LD_1_CD_SS_20160510c		LD_1_CD_SS_20160510c	Plt0PlantDesign
Unassigned			
FRL2020_002	001	000498/A;1	
FRL2020_002	001	000496/A;1	
LD_3a_PTO_Line_20160510c	001	LINE_000084/001;1-LD_3a_PTO_Line_20160510c	
LD_3b_PTO_Station_20160510c	001	STATION_000085/001;1-LD_3b_PTO_Station_20160510c	
LD_3c_PTO_Zone_20160510c	001	ZONE_000086/001;1-LD_3c_PTO_Zone_20160510c	

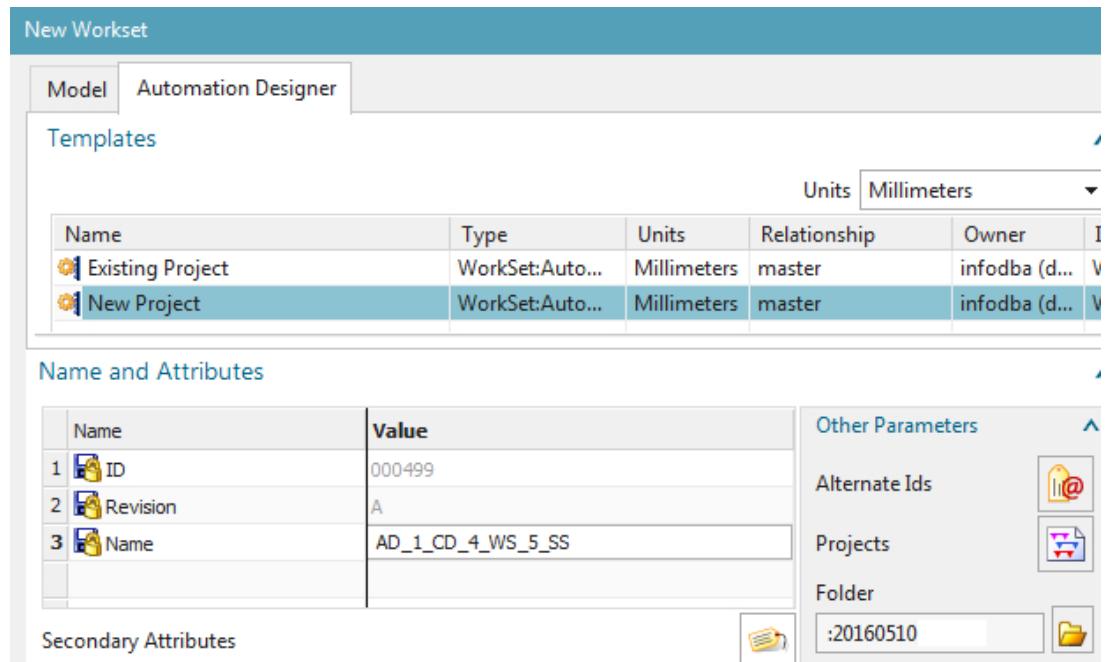
**4.3 Add 2 Line Designer conveyors**

# Create Automation Designer workset and Engineering Objects

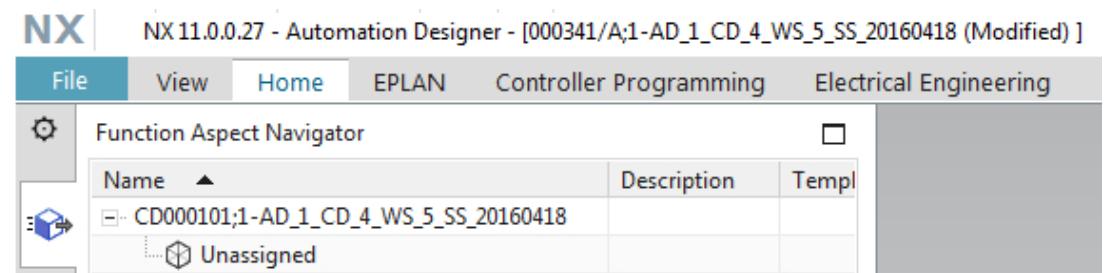
5

## 5.1 Create project workset (and Collaborative Design + subset)

1. Select File→New→Workset.
2. In tab "Automation Designer" select New Project.
3. Create a new Automation Designer project with name "AD\_1\_CD\_4\_WS\_5\_SS".

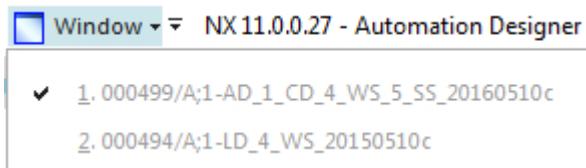


4. Click OK.



Note that the Line Designer workset and the Automation Designer Collaborative Design, workset and subset are open.

## 5.2 Create Engineering Object Definitions



## 5.2 Create Engineering Object Definitions

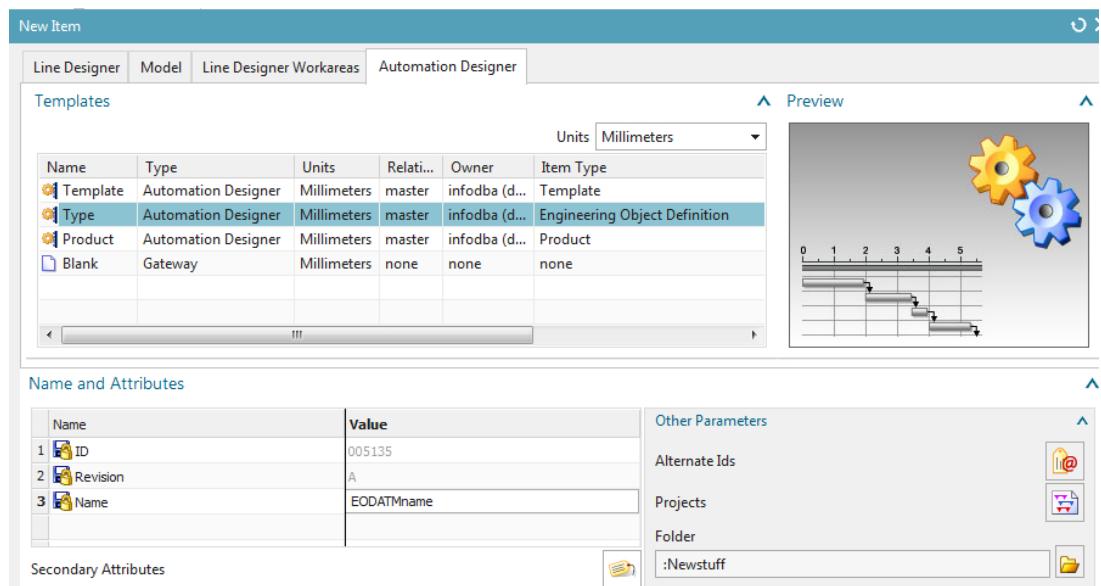
Now you create the definitions for the Engineering Objects you create later. These definitions specify the classification class of the Engineering Objects.

Create the first Engineering Object Definition.

1. Select **File→New→Item**.

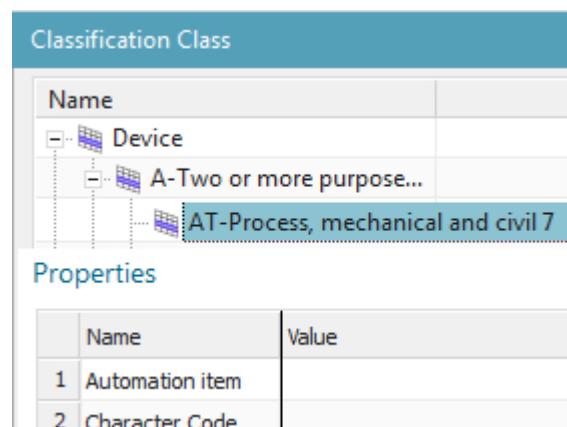
2. In tab **Automation Designer** select **Type**.

3. Enter the name "EODATMname". This will be locked after you set it. This is the "description" when you add an Engineering Object.



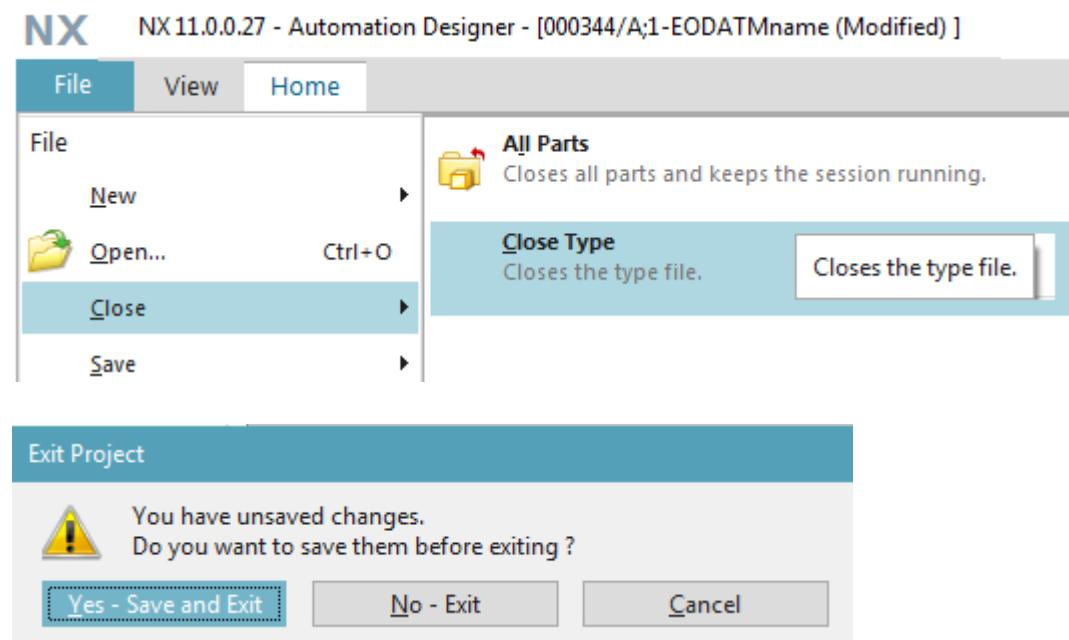
4. Click **OK**.

5. In the **Classification Class** dialog select **Device / A / AT**.



6. Click **OK**.

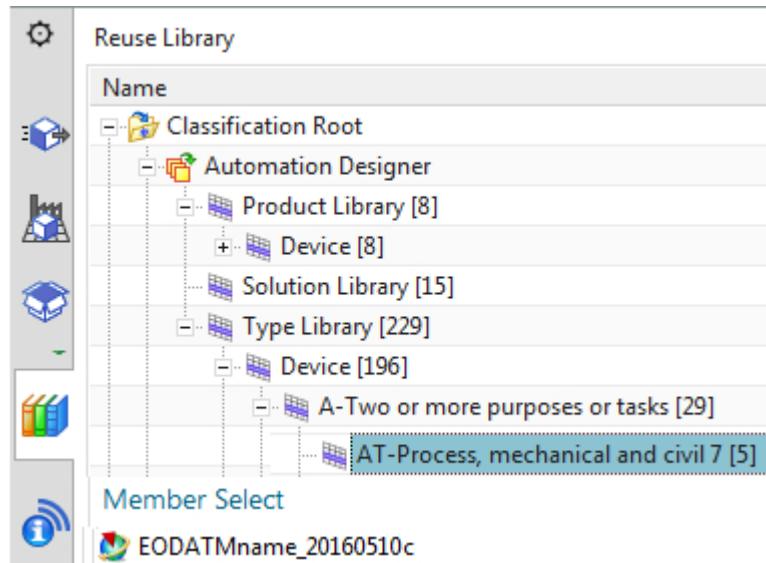
7. Select **File→Close→Close type**.



8. Click **Yes - Save and Exit**.

9. Verify that the Engineering Object Definition is in the Reuse Library.

### 5.3 Create Engineering Object names



10. Create the remaining Engineering Object Definitions:

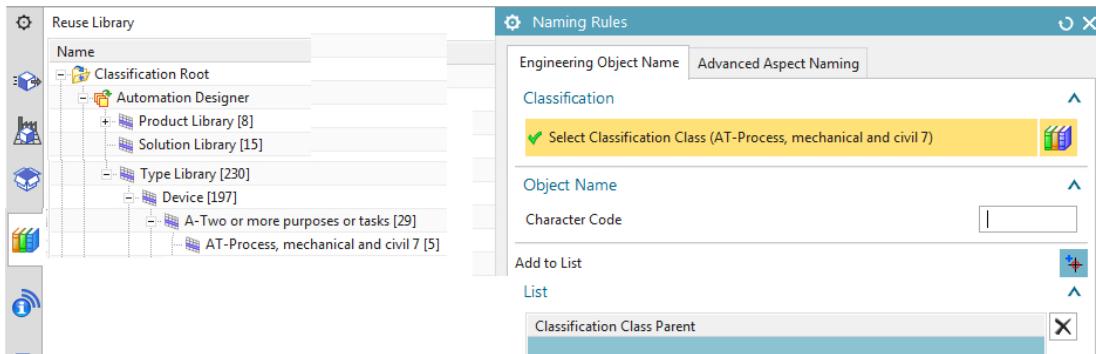
EODef	Classification Root
1. EODATMname (created above)	Device / A ->1 purpose or task / AT
2. EODTLname	Device / U-Keep
3. EODGLname	Device / G-Generator / GL-Continuous flow
4. EODMAname	Device / M-Motor / MA-Electromagnetic
5. EODBGname	Device / B-Measurement / BG-Gauge,position
6. EODTFname	Device / T-Conversion / TF-Signals
7. EODKFname	Device / K-Processing / KF-Electrical signals
8. EODCHname	Devicefunction / Electrical / Input/output

## 5.3

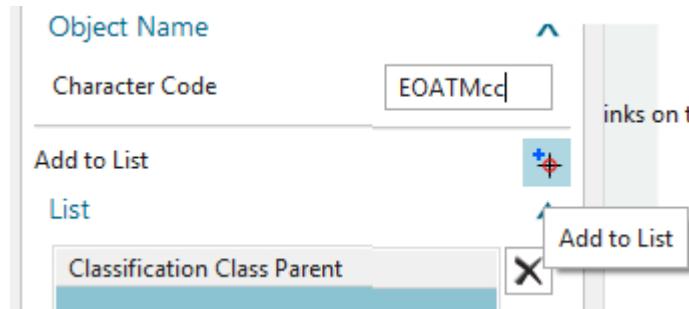
### Create Engineering Object names

You now create the Engineering Object names that will display in the aspect tree.

1. Click on Home→Naming Rules.
2. Select Classification Root/Automation Designer/Type Library/Device/A/AT.



3. For “Character Code” enter “EOATMcc”.



4. Click **Add to List**.



The following is the result.

List		Classification Class	Character Code
Classification Class Parent	TC Classification Root->Classification Root->Automation Designer->Type Library->Device->A-Two or more purposes or tasks->AT-Process, mechanical and civil 7	AT-Process, mechanical and civil 7	EOATMcc

5. Create the remaining Engineering Object names.

Character code	Classification parent
1. EOATMcc (created above)	Device / A ->1 purpose or task / AT
2. EOTLcc	Device / U-Keep
3. EOGLcc	Device / G-Generator / GL-Continuous flow
4. EOMAcc	Device / M-Motor / MA-Electromagnetic
5. EOBGcc	Device / B-Measurement / BG-Gauge,position
6. EOTFcc	Device / T-Conversion / TF-Signals
7. EOKFcc	Device / K-Processing / KF-Electrical signals
8. EOCHcc	Devicefunction / Electrical / Input/output

The following shows the result.

Naming Rules			
Engineering Object Name	Advanced Aspect Naming	Classification Class	Character Code
<b>List</b>			
Classification Class Parent	TC Classification Root->Classification Root->Automation Designer->Type Library->Device->A-Two or more purposes or tasks->AT-Process, mechanical and civil 7	AT-Process, mechanical and civil 7	EOATMcc
	TC Classification Root->Classification Root->Automation Designer->Type Library->Device->U-Keep	U-Keep	EOTLcc
	TC Classification Root->Classification Root->Automation Designer->Type Library->Device->G-Generator->GL-Continuous flow of solid matter	GL-Continuous flow of solid matter	EOGLcc
	TC Classification Root->Classification Root->Automation Designer->Type Library->Device->M-Motor->MA-Electromagnetic	MA-Electromagnetic	EOMAcc
	TC Classification Root->Classification Root->Automation Designer->Type Library->Device->B-Measurement->BG-Gauge, position, length	BG-Gauge, position, length	EOBGcc
	TC Classification Root->Classification Root->Automation Designer->Type Library->Device->K-Processing->KF-Electrical signals	KF-Electrical signals	EOKFcc
	TC Classification Root->Classification Root->Automation Designer->Type Library->Devicefunction->Electrical->Input/output	Input/output	EOCHcc

## 5.4

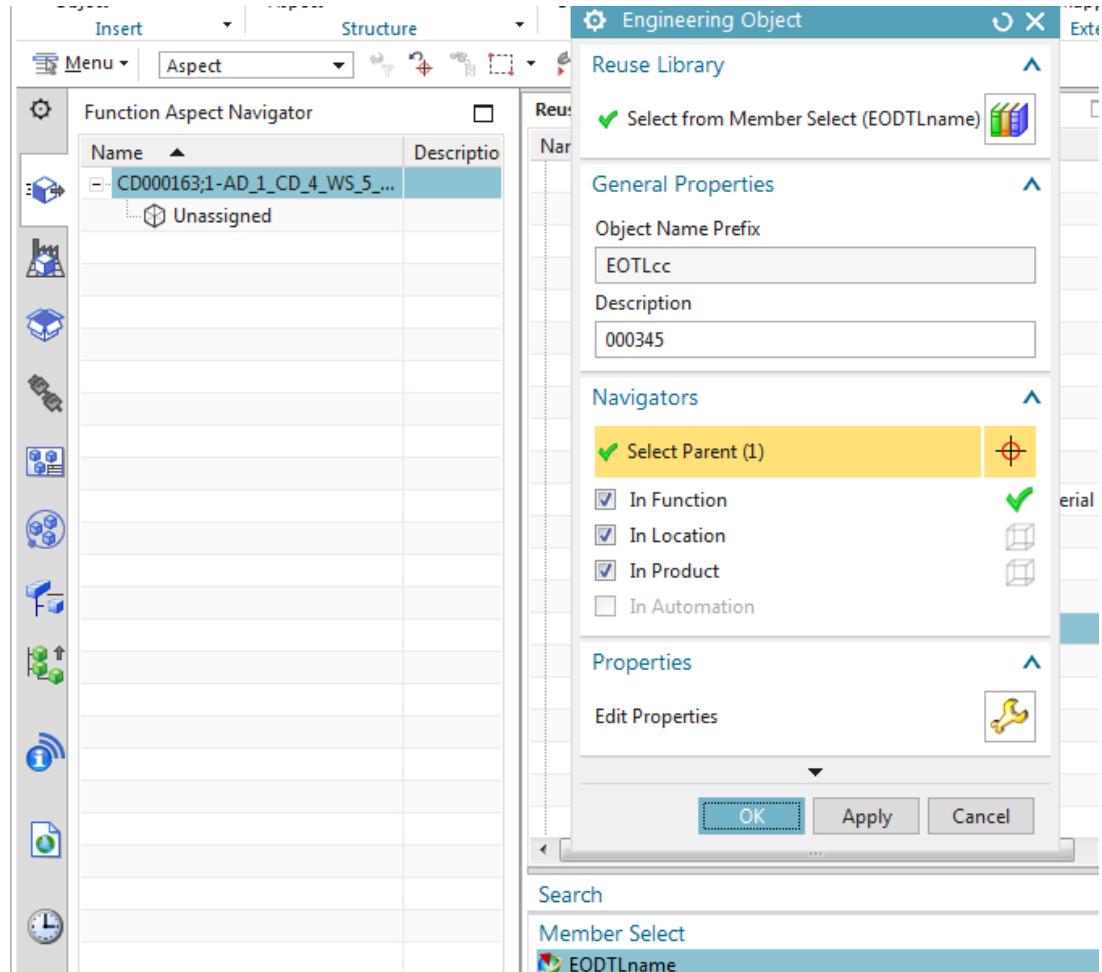
## Add Engineering Objects

In this section you will only add Engineering Objects in the Function aspect.

#### 5.4 Add Engineering Objects

Drag and drop the Engineering Object Definitions to create the Engineering Objects in the aspect tree.

1. Drag and drop EOTLcc.



3. Drag the remaining Engineering Objects to create the following Function aspect tree.

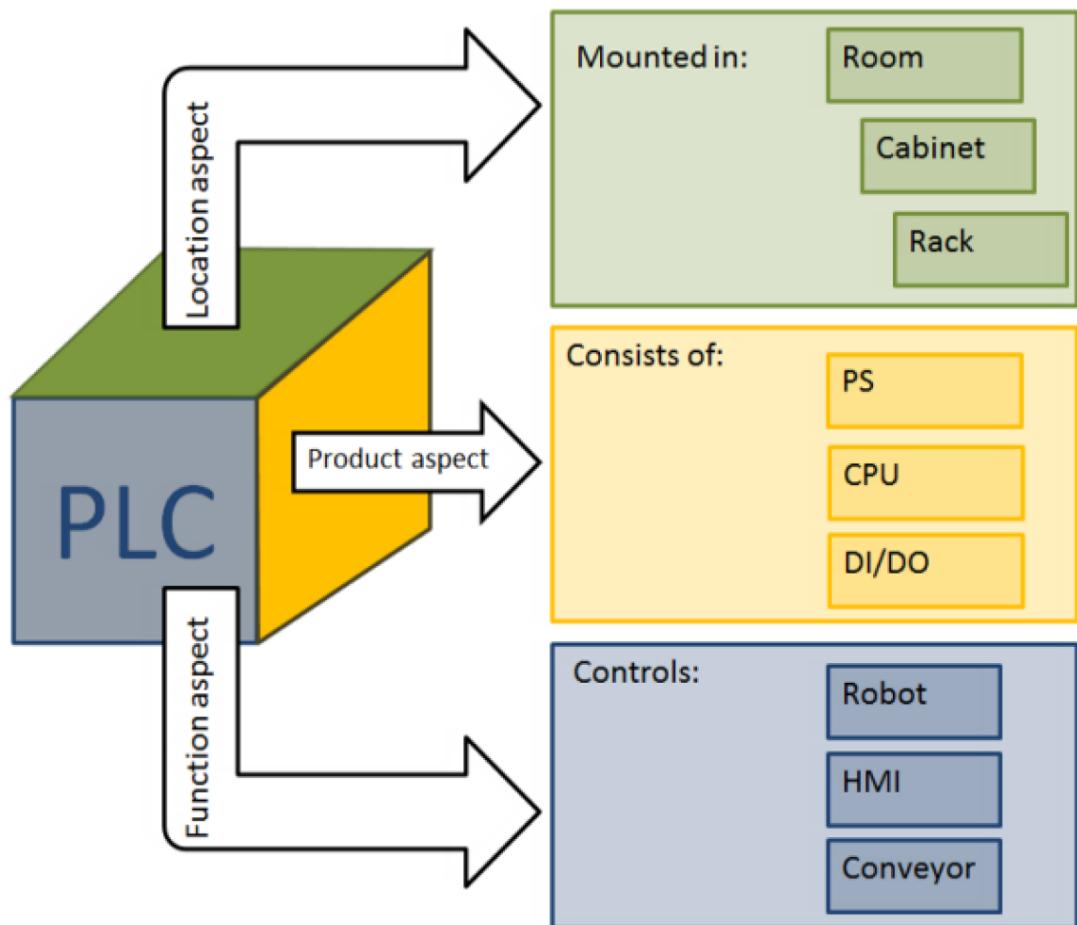
Name	Description
CD000101;1-AD_1_CD_4_WS_5_SS_20160418	
Unassigned	
=_001 [EODATMname]	000344
=_004 [EODTLname]	000345
=ConveyorF001 [EODGLname]	000346
=_MotorF001 [EODMAname]	000347
=_SensorF001 [EODBpname]	000348
=_DrivePowerF001 [EODTFname3]	000351
=_DriveControlF001 [EODKFname]	000352
=_EOCHcc001 [EODCHname]	000353

## 5.5 Location-Product aspects

### Introduction to Engineering Objects and aspects

You can view the same object under one or more aspects. For each aspect, you consider only those features and relations that are relevant for that specific aspect. The following diagram illustrates this, using the example of a programmable logic controller (PLC).

## 5.5 Location-Product aspects



### Relevance of aspects for engineering applications

Many engineering applications are designed for a specific discipline and support only one particular view. The view influences how you model the production system or machine in the engineering application. It determines the following:

- The objects with which you work
- The hierarchical structure of the objects
- Object names

By providing separate Aspect Navigators for the function, location, and product aspects of your engineering, Automation Designer allows you to combine different views of engineering in one application and even in one object.

### Aspect Navigators

How you structure a production system or machine in an engineering application depends on the aspect that you consider.

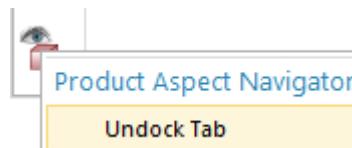
Because each aspect represents a view that may structure objects differently, Automation Designer provides several Aspect Navigators. Each *Aspect Navigator* represents one view of

planning. In each Aspect Navigator, you define the hierarchical structure of the Engineering Objects for the current aspect.

- In the Function Aspect Navigator, you organize Engineering Objects based on their intended function within the production system or machine.
- In the Location Aspect Navigator, you define the spatial relations of the Engineering Objects. Using location-related objects like buildings, floors, cabinets, or racks, the Location Aspect Navigator organizes all objects based on where they will be installed or mounted.
- In the Product Aspect Navigator, you document which hardware devices and software objects are physically needed to fulfill the function.

Now you will configure the location and product aspects.

1. Undock the location and product aspects.

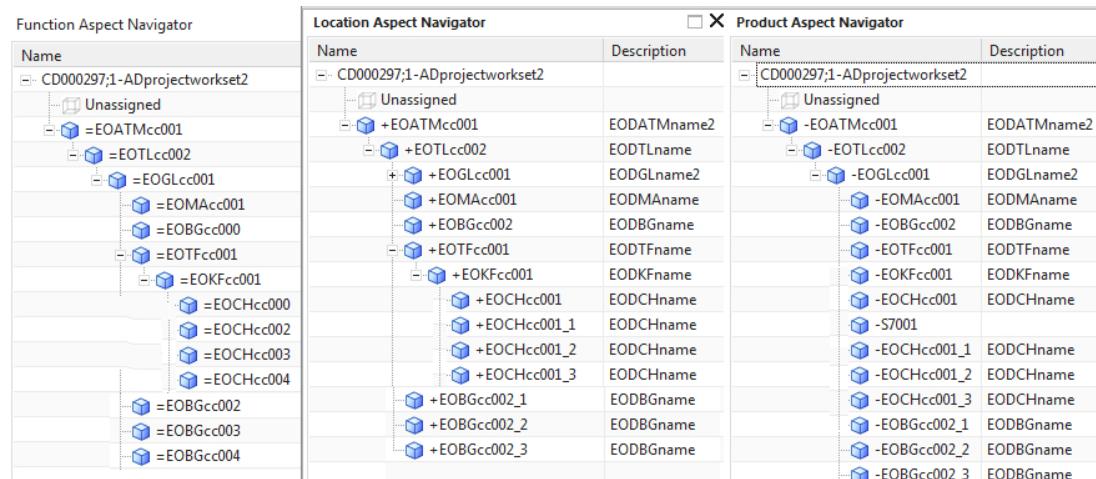


2. Organize (with drag and drop) the Engineering Objects in the Location and Product aspects.

**Location Aspect Navigator**

Name	Description
CD000297;1-ADprojectworkset2	
Unassigned	
+ EOATMcc001	EODATMname2

3. The result should look like this.



*5.5 Location-Product aspects*

# Map Line Designer and Automation Designer

## 6.1

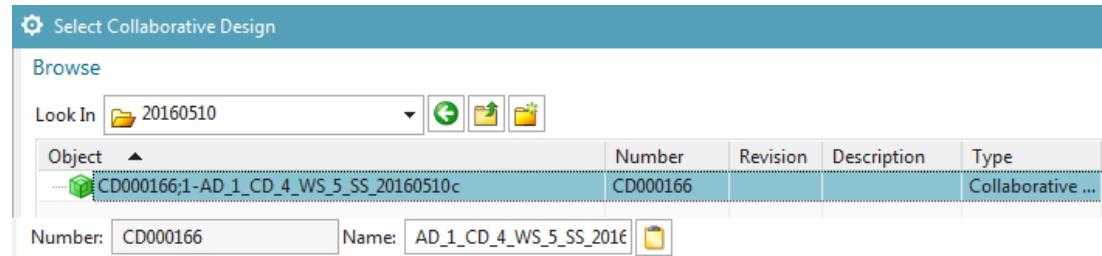
### Link Automation Designer and Line Designer Collaborative Designs

Mappings from mechanical and line design data to automation data eliminate the need for reentering data and make mechanical changes transparent. You can directly use the mechanical and line design data in Automation Designer.

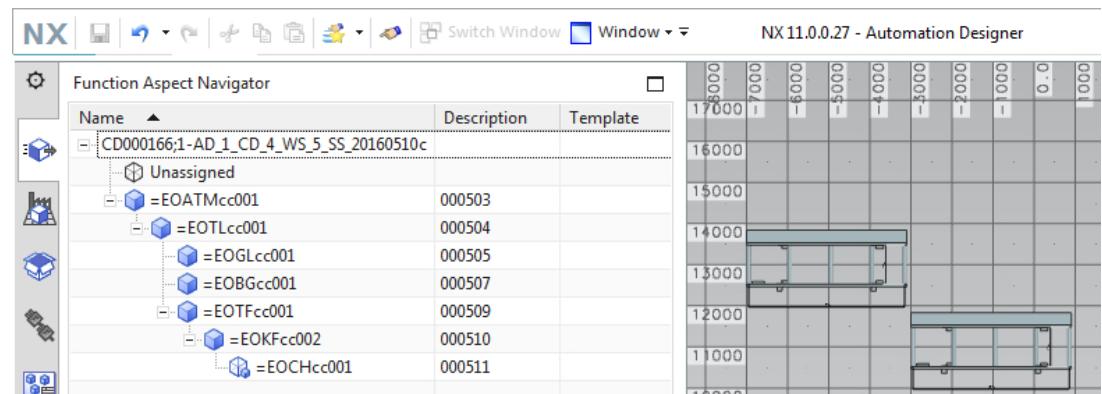
The mapping rule between external type and Automation Designer Type is stored in the database. It is used as the default in all projects and shown as a predefined mapping in the **Type Mapping** dialog box.

The Line Designer project is structured in TeamCenter using a Plant Design. The Automation Designer project is structured in TeamCenter using a Collaborative Design. To connect the two designs you need to link the Plant Design (Line Designer) with the Collaborative Design (Automation Designer). This action needs to be done only once. After this you can map the mechanical layout (Line Designer) to Automation Designer Engineering Objects.

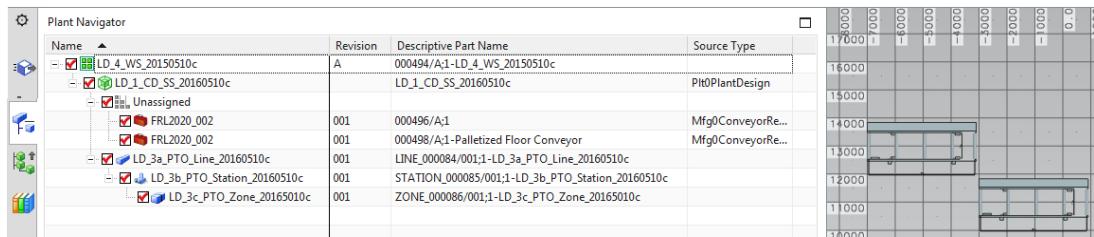
1. Close the Automation Designer project you created previously.
2. Open the Line Designer Collaborative Design.
3. Select **File→All Applications→Automation Designer**.
4. Select the Automation Designer Collaborative Design.



The Automation Designer Collaborative Design is on the left and the Line Designer Collaborative Design on the right.



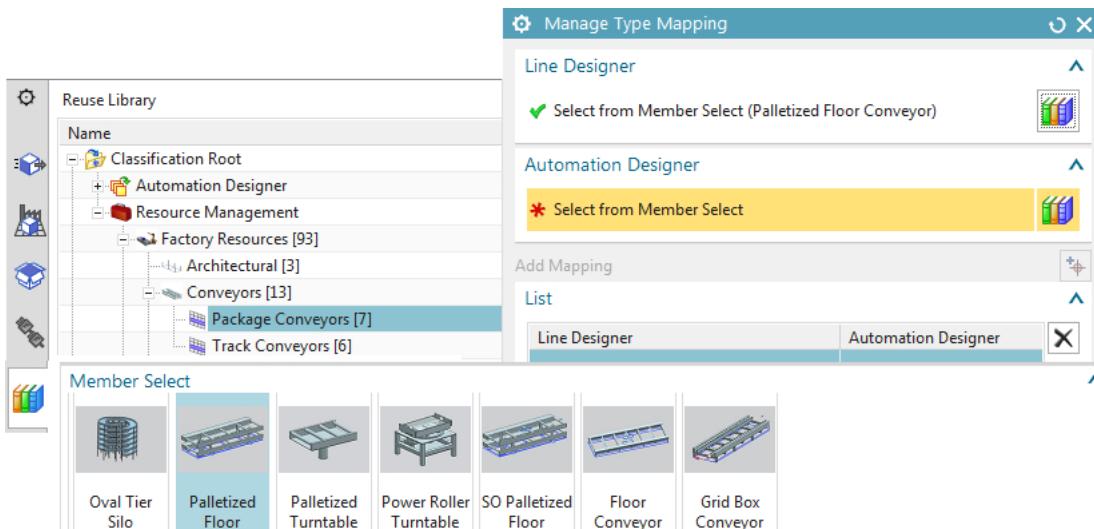
## 6.2 Manage type mapping



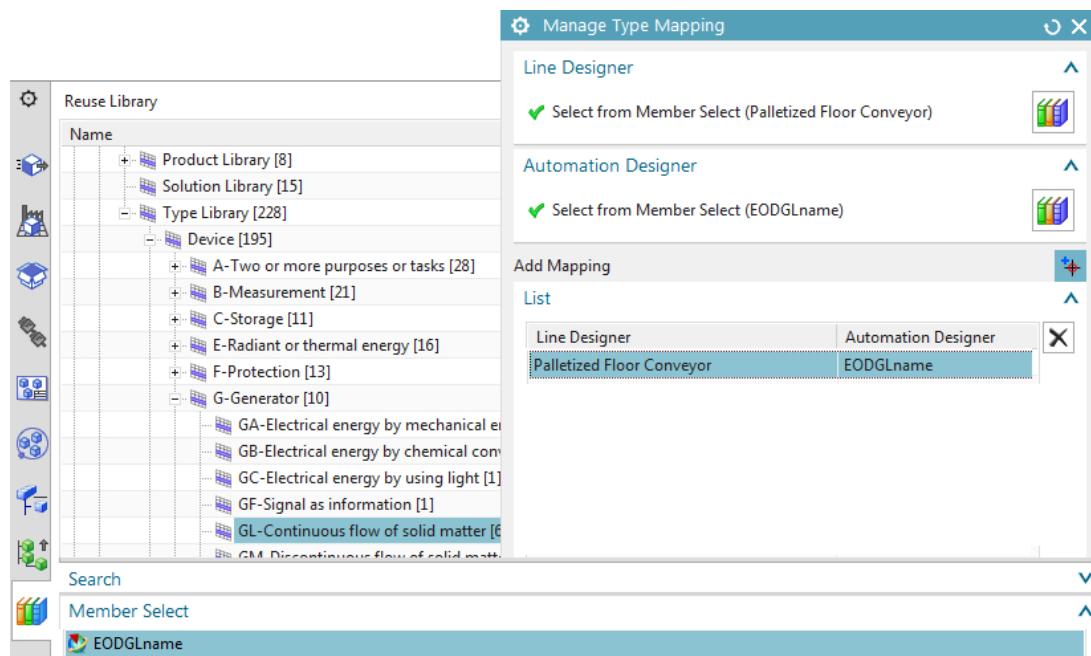
## 6.2 Manage type mapping

You now will define the type mapping. Type mapping determines for which type of Line Designer object what type of Engineering Object will be created when you use **Map to new**.

1. Open the **Manage Type Mapping** dialog.
2. Under **Line Designer** select the conveyor.



2. Under **Automation Designer** select GL. A list of existing mappings may appear.



## 6.3

## Manage object mapping

Use the **Manage Object Mapping** dialog box to map single external objects to single Engineering Objects or templates. Then continue engineering with the mapped Engineering Object or template.

There are 3 ways to map objects:

1. Map to existing
2. Map to new
3. Map to new based on type

### 1. Map to existing

You can map external objects to existing Engineering Objects or templates. You can choose an Engineering Object or template whose type matches the type mapping defined for all projects. Or you choose an Engineering Object or template whose definition deviates from the type mapping. In that case, the type mapping is overridden for this one object mapping.

1. Click **Manage object mapping**. 2 conveyors appear.

2. Select a conveyor.

### 6.3 Manage object mapping

External Name	External Type	Status	RDS	Type
FRL2020_002	Palletized Floor Conveyor			EODGLname
FRL2020_002	Palletized Floor Conveyor			EODGLname

3. Click Map to existing in project.

4. Select GL.

5. Click OK. The following is the result.

External Name	External Type	Status	RDS	Type
FRL2020_002	Palletized Floor Conveyor		=EOATMcc001.EOTLcc0...	
FRL2020_002	Palletized Floor Conveyor			EODGLname

### 2. Map to new

You can map an external object to a new Engineering Object and override the type mapping defined for all projects for this one mapping.

Now try to create a new Engineering Object based on the mapping.

1. Select the unmapped Engineering Object.

2. Click Map to new.

External Name	External Type	Status	RDS	Type
FRL2020_002	Palletized Floor Conveyor		=EOATMcc001.EOTLcc0...	
FRL2020_002	Palletized Floor Conveyor			EODGLname

3. Select the Engineering Object from reuse library (you cannot select GL, because it is mapped).

4. For the parent select TL.

Name	Description
CD000166;1-AD_1_CD_4_WS_5_SS_20160510c	
Unassigned	
=EOATMcc001	000503
= EOTLcc001	000504
= EOGLcc001	000505
= EOBGcc001_1	000507
= EOTFcc001_1	000509
= EOKFcc002	000510
= EOCHcc001	000511
= EOMAcc001	000506

5. Click OK. A new Engineering Object is created and mapped to the conveyor.

### 6.3 Manage object mapping

The screenshot shows the Function Aspect Navigator on the left and the Manage Object Mapping interface on the right. The Function Aspect Navigator lists objects under project CD000166;1-AD\_1\_CD\_4\_WS\_5\_SS\_2016, including Unassigned, =EOATMcc001, =EOTLcc001, =EOGLcc001, =EOBGcc001\_1, =EOTFcc001\_1, =EOKFcc002, =EOCHcc001, =EOMAcc001, and =EOMAcc002. The Manage Object Mapping interface has tabs for Actions (Map to Existing in Project, Map to New, Map to New Based on Type, Unmap) and Object Mapping (Show, Unhidden selected). A table lists object mappings:

External Name	External Type	Status	RDS	Type
FRL2020_002	Palletized Floor Conveyor		=EOATMcc001.EOTLcc0...	
FRL2020_002	Palletized Floor Conveyor		=EOATMcc001.EOTLcc0...	

#### 3. Map to new based on type

If you want to map external objects to new Engineering Objects, you can use the type mappings that were defined as default for all projects.

1. Unmap the previous mapping (select and click Unmap).

The screenshot shows the Manage Object Mapping interface with the 'Map to New Based on Type' tab selected. The table shows two entries for FRL2020\_002, both mapped to Palletized Floor Conveyor with RDS icons.

External Name	External Type	Status	RDS	Type
FRL2020_002	Palletized Floor Conveyor		=EOATMcc001.EOTLcc0...	
FRL2020_002	Palletized Floor Conveyor		=EOATMcc001.EOTLcc0...	

2. Click **Map to new based on type**. GL is automatically selected.

The screenshot shows the 'Function Aspect Navigator' and 'Reuse Library' panes on the left and right respectively, with the main focus on the 'Manage Object Mapping' dialog.

**Function Aspect Navigator:**

- Name: CD000166;1-AD\_1\_CD\_4\_WS\_5\_SS\_20160510c
- Unassigned:
  - =EOGLcc002 (Description: 000346)
  - =EOATMcc001 (Description: 000502)

**Reuse Library:**

- Classification Root
  - Automation Designer
    - Product Library [8]
    - Solution Library [15]

**Manage Object Mapping Dialog:**

**Actions:**

- Map to Existing in Project
- Map to New
- Map to New Based on Type
- Unmap

**Object Mapping:**

Show:

- Unhidden
- Hidden
- Unmapped
- Mapped
- Deleted
- All

External Name	External Type	Status	RDS	Type
FRL2020_002	Palletized Floor Conveyor		=EOATMcc001.EOTLcc0...	
FRL2020_002	Palletized Floor Conveyor		=???.EOGLcc002/+???.E...	EODGLname

*6.3 Manage object mapping*

# Configure (non-template) EPLAN

## 7.1

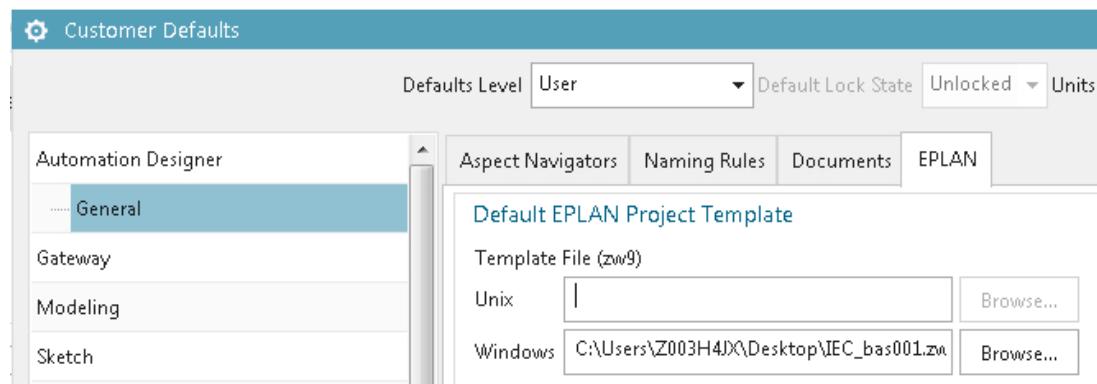
### Import EPLAN project template

You must predefine a Default EPLAN Project Template in Customer Defaults for macro import. If the path is not set to the template, you cannot import any macro. It is recommended to do so before starting a project. Ensure that the used EPLAN project template includes the symbol libraries used in the macros. Otherwise the symbols are not visible, neither on the imported EPLAN macro nor on the created PDF.

1. Select **Menu→Preferences→Customer Defaults**.



2. In **Automation Designer→General**, under tab EPLAN, select the Default EPLAN Project Template for Windows.



3. Click **OK**.

## 7.2

### Add PM250D macro

EPLAN page macros contain full pages. EPLAN window and symbol macros represent cut-outs of a page which can be reused by placing them on the page.

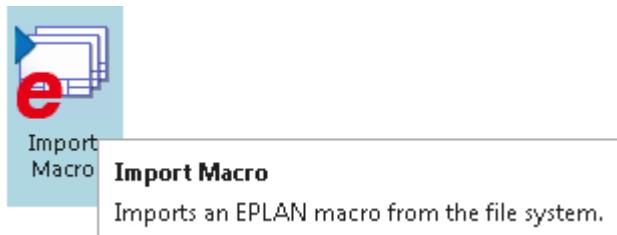
In EPLAN, macro placement is done graphically. In the Automation Designer you want to do the same and place window macros on pages to reuse the configurations.

As the graphics of EPLAN macros are not accessible from Automation Designer, insertion points on the EPLAN macros allows you to place your macros.

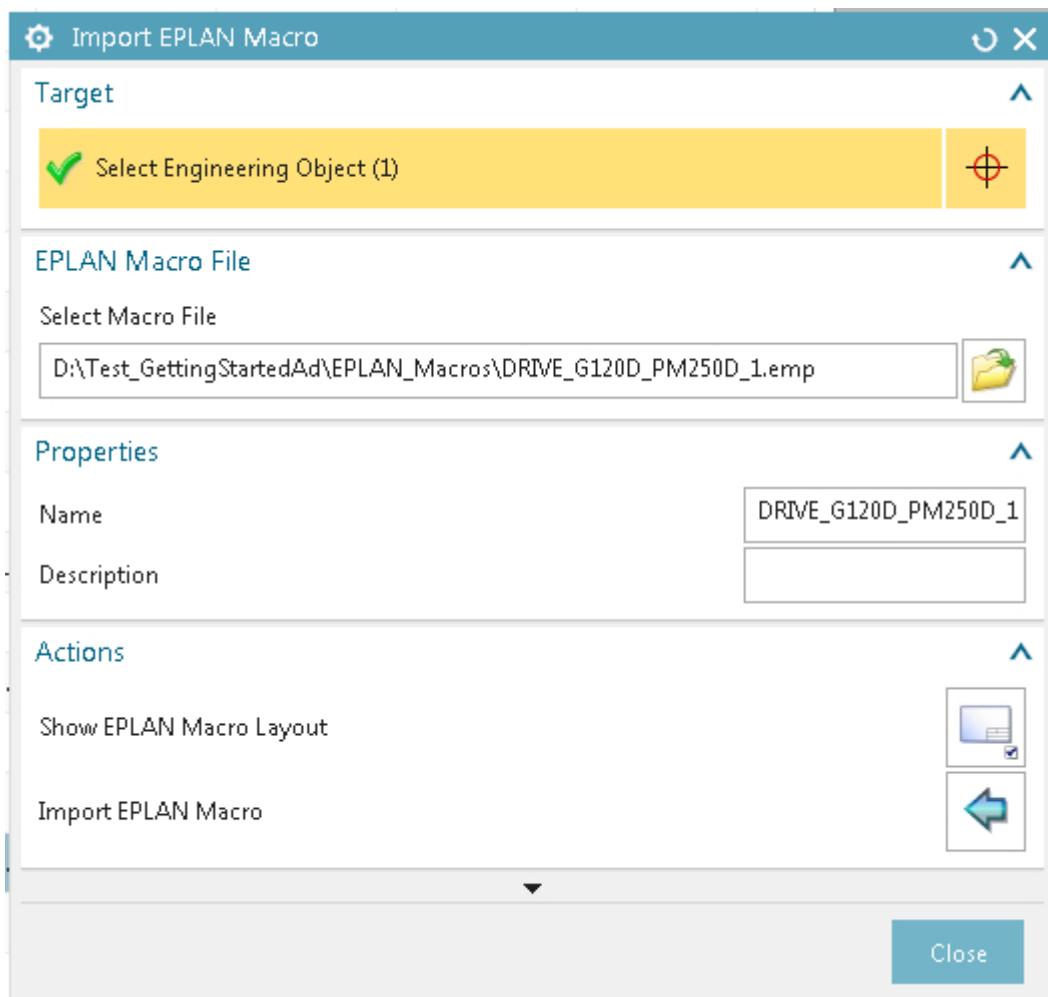
## 7.2 Add PM250D macro

The target insertion point is the socket while the plug is the anchor point of a window or symbol macro. Without a plug, the macro cannot be used in Automation Designer. Sockets can be used on each macro type. A macro can only have one plug, but none, one or multiple sockets.

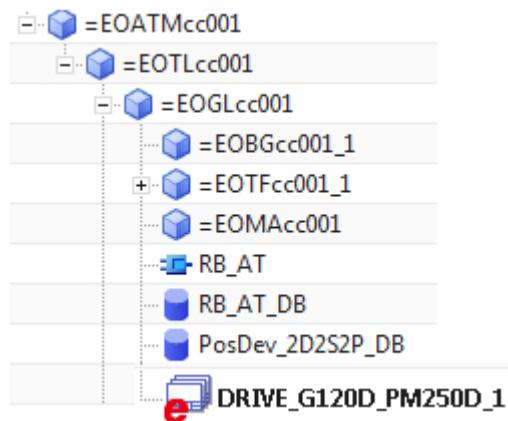
1. Click on **Import Macro**.



2. Import **DRIVE\_G120D\_PM250D\_1.emp** under Engineering Object GL.

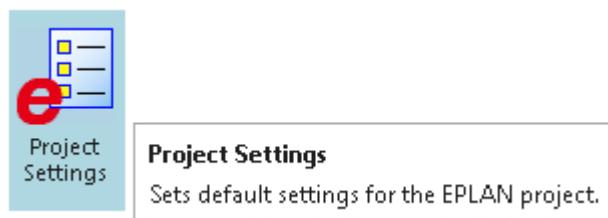


3. Click **Close**. The macro appears in the aspect tree.



## 7.3 Modify ELPAN template

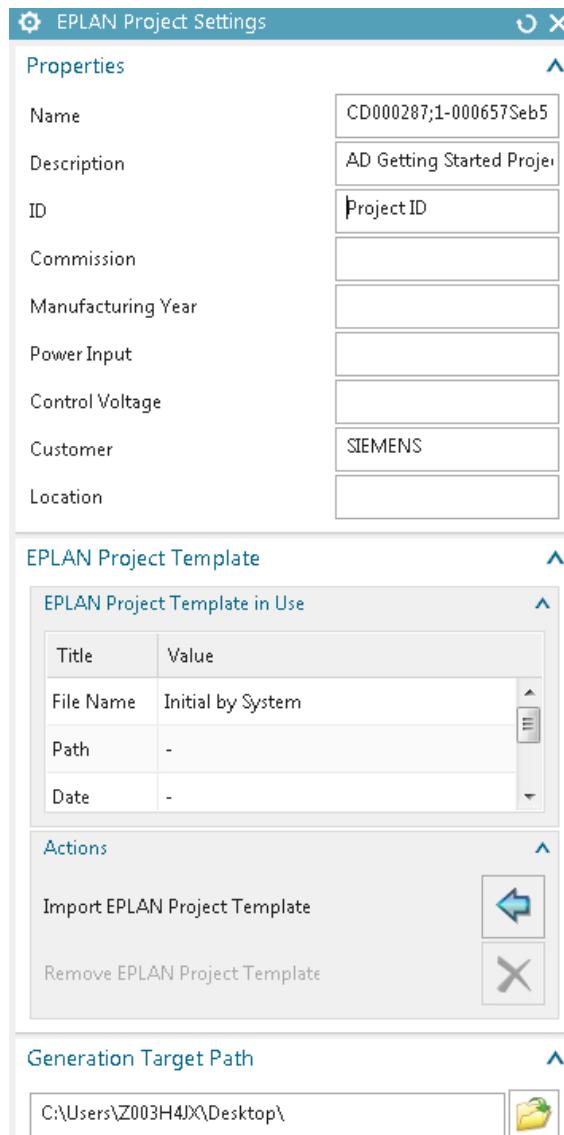
1. Under tab EPLAN, click on **Project Settings**.



2. Set the following values for the EPLAN template:

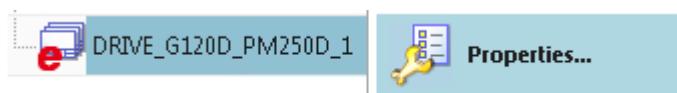
- Description: **AD Getting Started Project**
- ID: **Project ID**
- Customer: **SIEMENS**

#### 7.4 Modify EPLAN macro



## 7.4 Modify EPLAN macro

1. In the Function Aspect, right-click on the EPLAN macro and select **Properties**.



The following shows the default properties.

Properties			
Engineering Object Attributes			
Title/Alias	Value	Type	R...
+  Aspect Function			
-  General			
Class	EPLAN Page Macro	String	
Reference Designation Set	=UL-Machinery.UF-Instrumentation equipment	String	
Type	EPLAN Page Macro	String	
-  Type			
Character Code	EPLAN	String	
Description		String	
Full page name	1	String	
Function		String	
Import Path of EPLAN Macro	D:\Test_GettingStartedAd\EPLAN_Macros	String	
Location		String	
Name of EPLAN Macro	DRIVE_G120D_PM250D_1.emp	String	
Object Name	DRIVE_G120D_PM250D_1	String	
Page Description		String	
Page name	1	String	
Unique Identifier	EPLAN021	String	
Variable: ControlUnitFunctionText		String	
Variable: ControlUnitName		String	
Variable: ControlUnitPartNumber1		String	
Variable: ControlUnitPartNumber2		String	
Variable: MotorCableFunctionText		String	
Variable: MotorCableName		String	
Variable: MotorCablePartNumber1		String	
Variable: MotorCablePartNumber2		String	
Variable: MotorFunctionText		String	
Variable: MotorName		String	
Variable: MotorPartNumber1		String	
Variable: MotorPartNumber2		String	
Variable: PowerModuleFunctionText		String	
Variable: PowerModuleName		String	
Variable: PowerModulePartNumber1		String	
Variable: PowerModulePartNumber2		String	
Variable: PowerSupply24VName		String	
Variable: PowerSupply400VName		String	

2. Modify the following macro properties with the values shown. The properties are EPLAN placeholder variables that are specific for each macro.

Device property	Value	Description	Location in EPLAN macro report
Aspect Function / Name	SINAMICS G120	Name of the device	
Full page name		= Function + Location + "/" + Page name	
Type / Function	EOATMcc001.EOTLcc001.EOGLc c001	Function aspect of the device	Bottom right
Type / Location	EOTLcc001.EOGLcc001	Location aspect of the device	Bottom right

---

## 7.5 Generate

Type / Page Description	Motor, PM250D and frequency converter G120D	Page description	Bottom center
Type / Variable: ControlUnitName	CU250S-2	Name of the control unit	Left-center
Type / Variable: PowerModuleName	PM240	Name of power module	Right-center

## 7.5 Generate

After you have specified the EPLAN placeholder variables, generate the EPLAN project and open it in EPLAN. All placeholder variables show the specified values and are available in EPLAN.

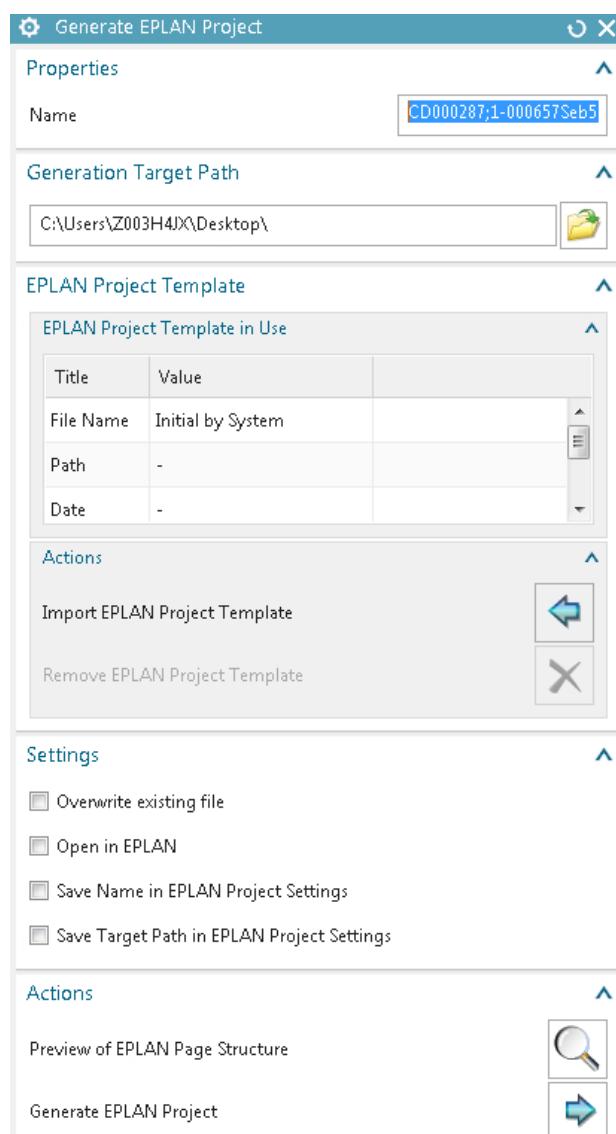
You are only allowed to generate an EPLAN project when you have a valid EPLAN license including EPLAN runtime license for the Automation Designer EPLAN generation. For generating EPLAN projects the EPLAN application is required that is not part of the Automation Designer software.

It is recommended to check the adapted EPLAN project in a preview if the hierarchy and structure of the macros are correct.

1. In the toolbar click **Generate Project**. The “Generate EPLAN Project” page appears.



2. Enter the **Name**, **Generation Target Path**, and check **Open in EPLAN**.
3. Click on **Generate EPLAN Project**.

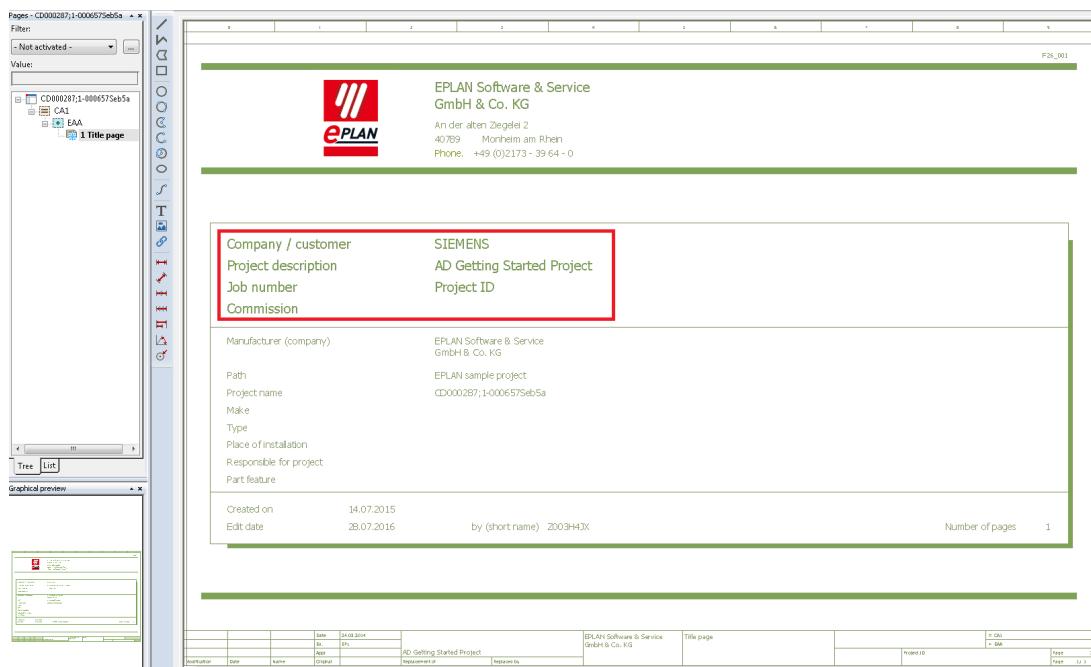


### EPLAN template

Open the title page of your EPLAN project. The template properties are generated on the title and the footer for the pages:

## Configure (non-template) EPLAN

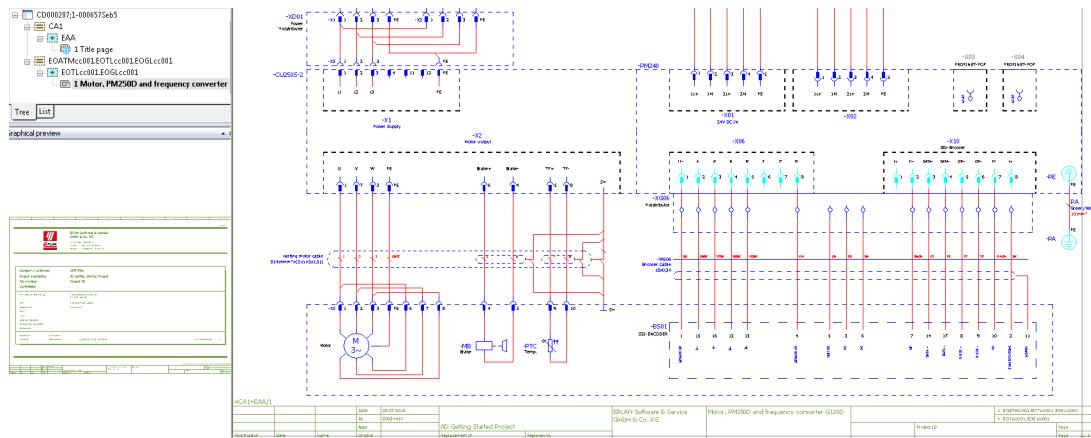
### 7.5 Generate



Date	24.03.2014	EPLAN Software & Service GmbH & Co. KG	
Eq.	EPL	AD Getting Started Project	
Appr			
Original	Replacement of	Replaced by	

### EPLAN macro

The added properties CU250S-2 and PM240 are added to the EPLAN page.



# Configure (non-template) TIA Portal

## 8.1

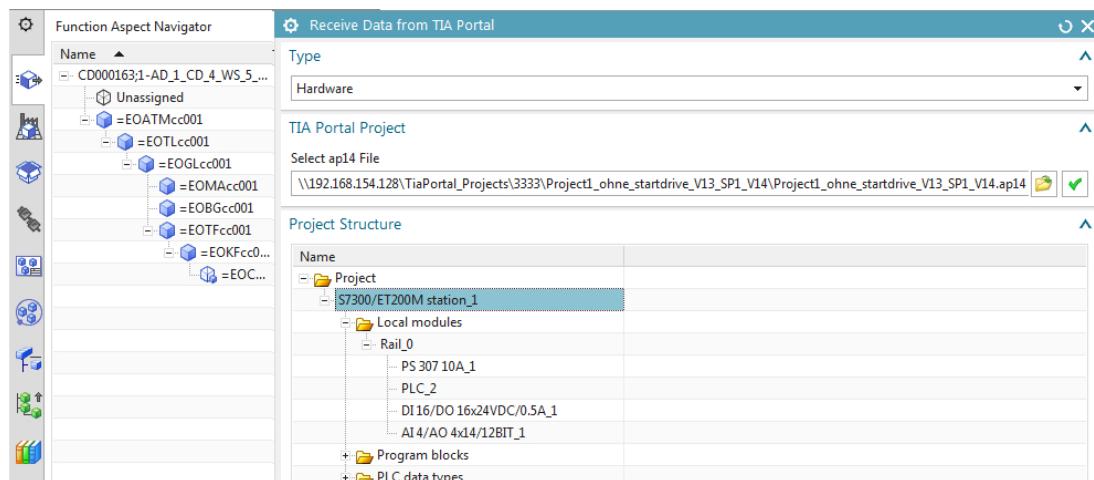
### Receive hardware, software

You can import the hardware configuration, tags, and program blocks from TIA Portal and adapt them so that they react to changes in the Automation Designer project environment. Import them into a project and use them directly or import them into a template and reuse the template.

#### 8.1.1. Receive hardware

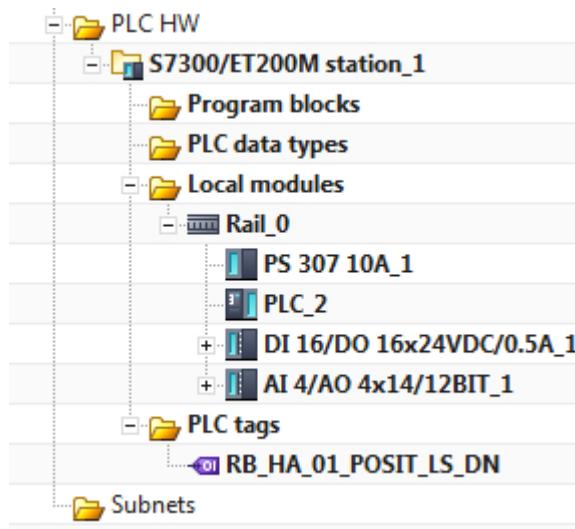
From STEP 7 or WinCC V13 SP1 onward, TIA Portal Openness is included in the delivery of STEP 7 or WinCC in TIA Portal. This enables you to program the applications which automate the engineering in TIA Portal.

1. In tab “Controller Programming” click **Receive Data**.
2. For Type select **Hardware**.
3. Select the .ap14 file.
4. Click the green arrow. The TIA Portal projects in the .ap14 file are displayed.
5. Select the project to import the PLC station with its modules.



6. Click **Receive from TIA Portal**. The station is imported.

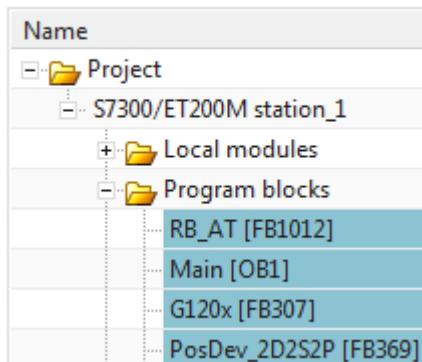
## 8.1 Receive hardware, software



### 8.1.2. Import software

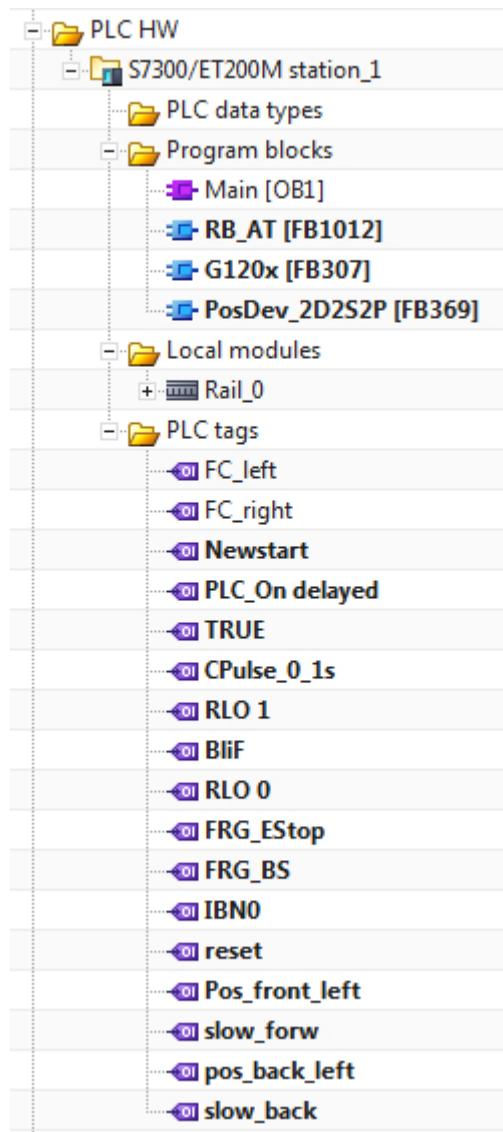
1. For Type select software.
2. Select the following blocks
  - Main [OB1]
  - RB\_AT
  - G120x
  - PosDev\_2D2S2P

#### Project Structure



Note: You could also import the IDBs, but in this Getting Started you import only the Function Blocks (either way is OK).

3. For **Target** select the station you imported.
4. Click **Receive from TIA Portal**. The software and the tags used in the Function Blocks are imported.

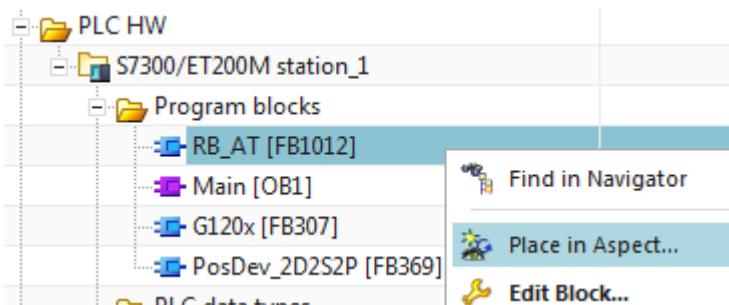


## 8.2 Place the function blocks in aspects and create IDBs

You now need to place the function blocks in the aspects and create IDBs in the aspects. Where you place the function blocks determines the aspect chain that will be used to create unique identifiers (symbolic names). In this Getting Started you focus only on the Function aspect.

1. Right-click on RB\_AT and select **Place in Aspect**.

## 8.2 Place the function blocks in aspects and create IDBs



2. For Placement select GL.

3. Click OK.

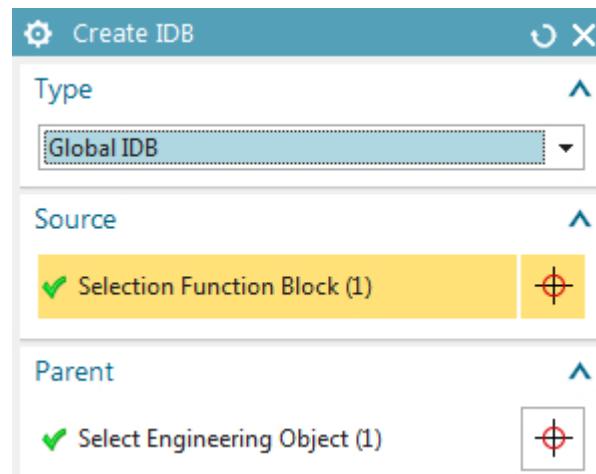
=EOATMcc001	000344
=EOTLcc001	000345
=EOGLcc001	000346
=EOMAcc001	000347
=EOBGcc001	000348
=EOTFcc001	000351
=EOKFcc0...	000352
=EOC...	000353
<b>RB_AT</b>	

4. Right-click on RB\_AT and select **Create IDB...**.



5. The default settings are correct. Click **OK**.

## 8.2 Place the function blocks in aspects and create IDBs



The following shows the result.

Name	Description
CD000163;1-AD_1_CD_4_WS_5_SS_20160509_2	
Unassigned	
PLC HW	
S7300/ET200M station_1	
Program blocks	
RB_AT [FB1012]	
Main [OB1]	
G120x [FB307]	
PosDev_2D2S2P [FB369]	
RB_AT_DB [DB1012]	
PLC data types	
Local modules	
Rail_0	

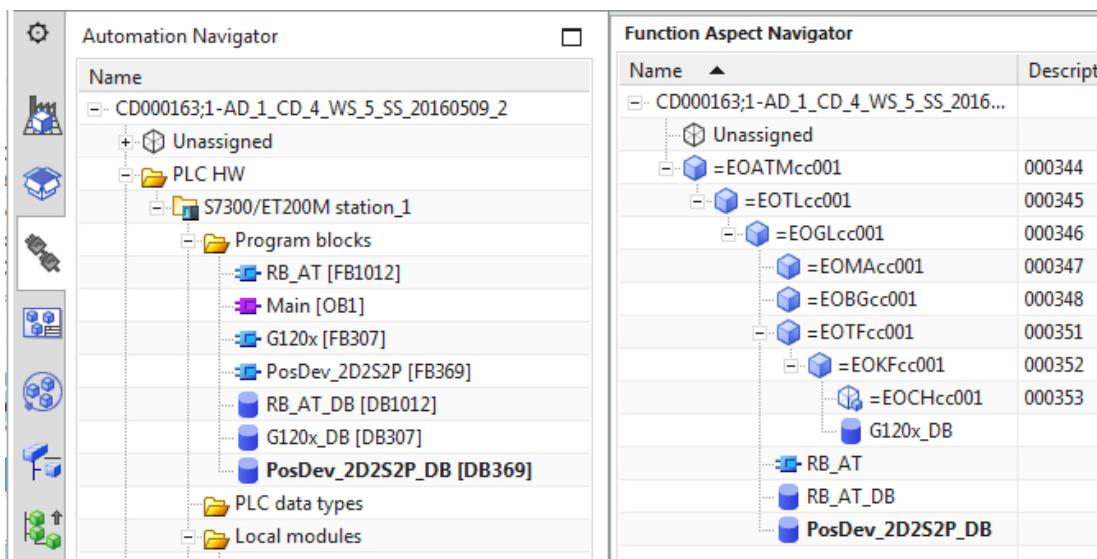
  

Name	Description
Unassigned	
=EOATMcc001	000344
=EOTLcc001	000345
=EOGLcc001	000346
=EOMAcc001	000347
=EOBGcc001	000348
=EOTFcc001	000351
=EOKFcc001	000352
=EOC...	000353
RB_AT	
RB_AT_DB	

6. Create an IDB for G120x under Engineering Object KF.

7. Create an IDB for PosDev under Engineering Object GL.

### 8.3 Add tags



## 8.3 Add tags

For this Getting Started we only add 2 tags.

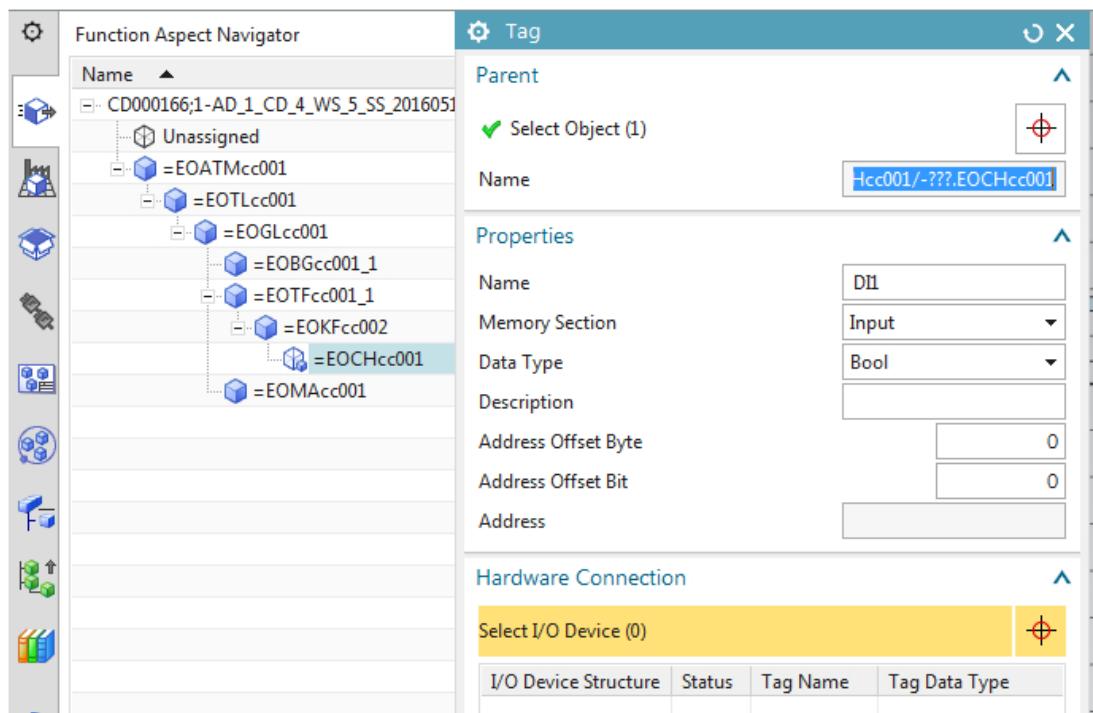
### 8.3.1. DI1

Add the sensor tag boolean input DI1.

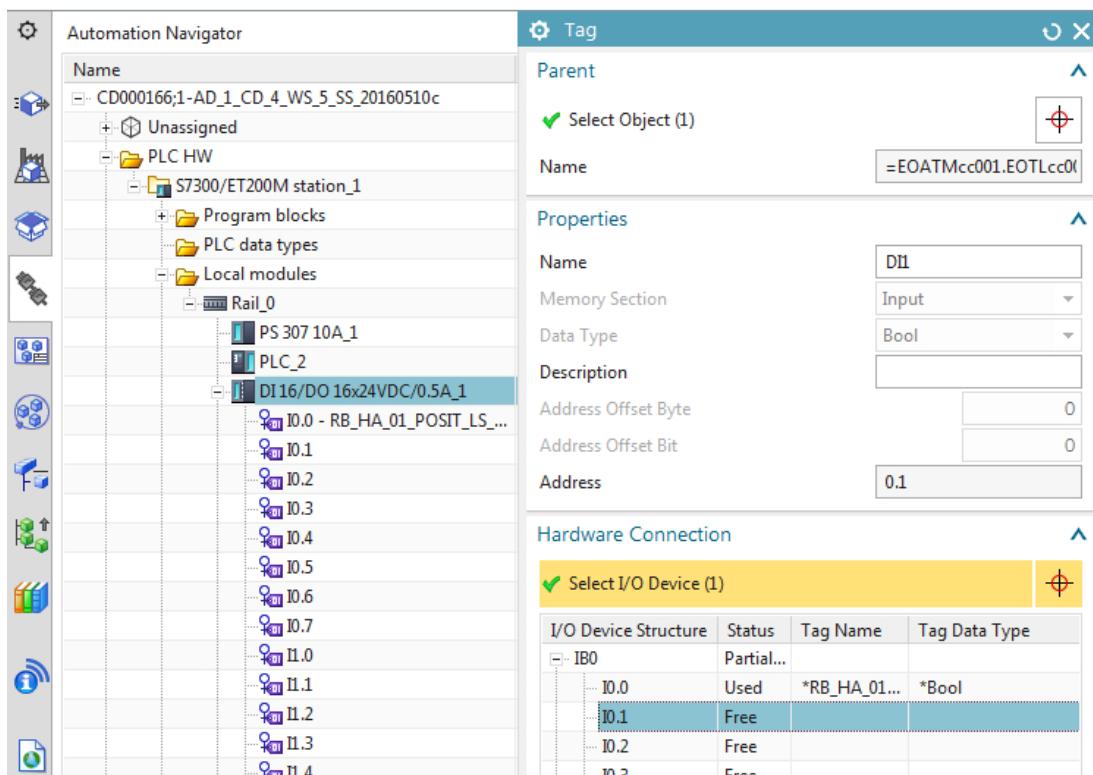
1. Click **Tag**.

2. Specify the tag properties:

- Parent EO = CH
- Name = DI1
- Memory section = Input
- Data Type = Boolean



3. Click on **Select I/O device**.
4. Click on the local module **DI 16 / DO 16**.
5. Select a free input (otherwise the memory area does not match).



6. Click OK.

=EOATMcc001	000503
=EOTLcc001	000504
=EOGLcc001	000505
=EOBGcc001_1	000507
=EOTFcc001_1	000509
=EOKFcc002	000510
=EOCHcc001	000511
DI1	
=EOMAcc001	000506

7. Right-click on DI1 and select **Properties**.

8. For **Interaction Method** select **Traditional**.

9. Select **Symbolic Name**.

10. For **Data Type** select **Value**.

11. For **Value** enter **DI1sn** (DI1 symbolic name). This is the unique ID of the tag (later you will define this using an expression).

PLC Tag Attributes

Title/Alias	Value	Units	T...	Type	R...	D...	I...
Address							
General							
Symbolic Name	DI1sn			String			

Category (optional)

Title/Alias

Data Type

Value  Expression Formula

Value

Accept Edit

### 8.3.2. PID0

Add the tag PID0 (digital input).

1. Click **Tag**.

2. Specify the tag properties:

- Parent EO = KF
- Name = **PID0**

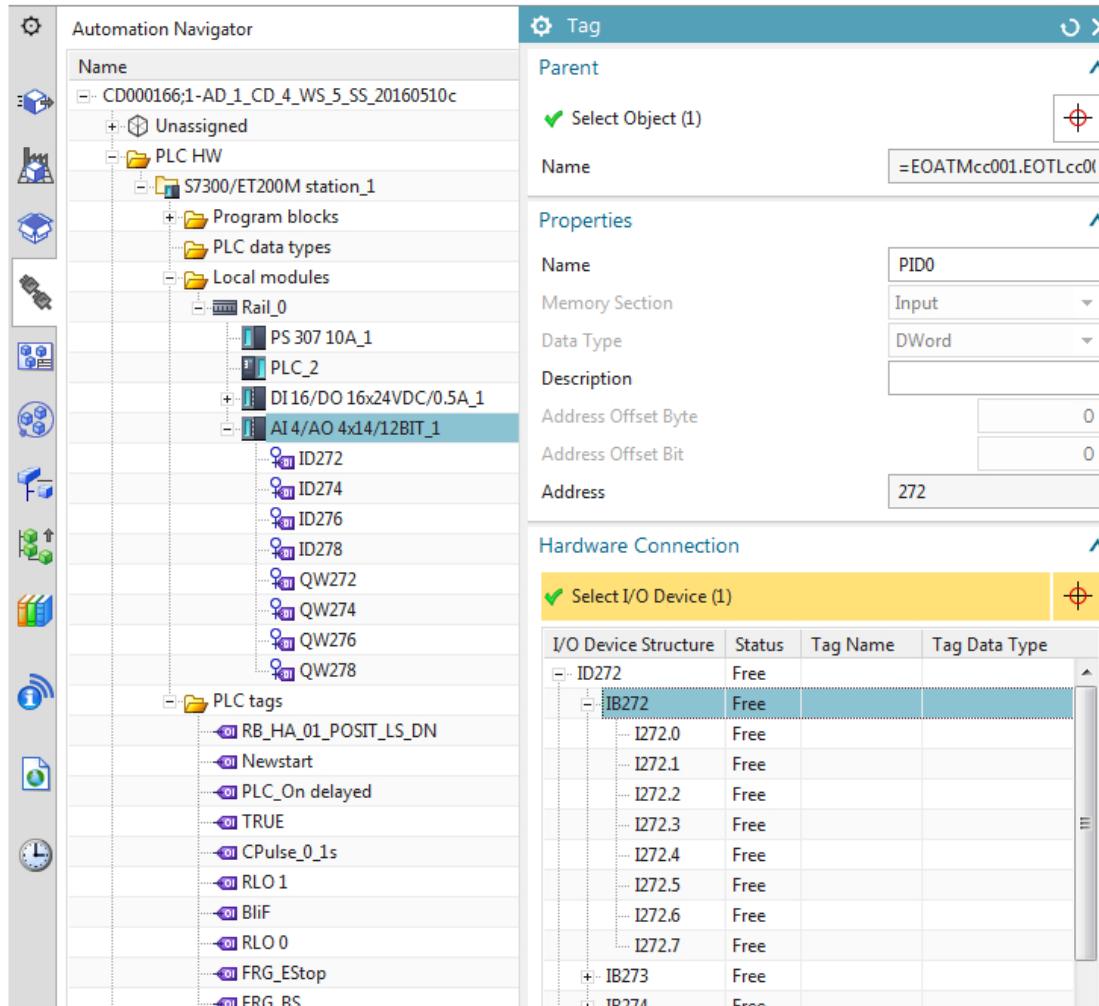
- Memory section = **Input**
- Data Type = **DWord**

The screenshot shows the TIA Portal interface for configuring tags. On the left is the 'Function Aspect Navigator' showing a hierarchy of function blocks and their parameters. On the right is the 'Tag' configuration dialog with several tabs:

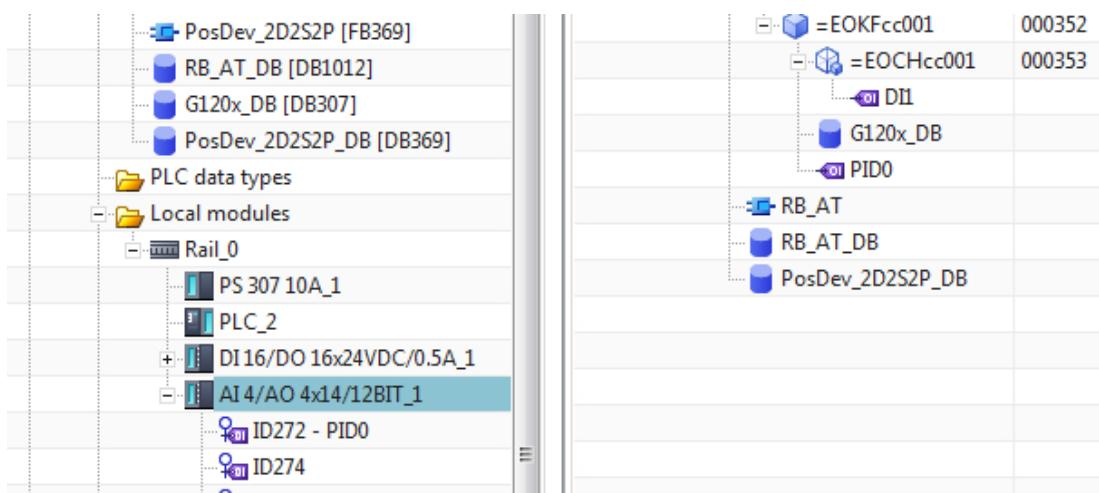
- Parent:** Shows 'Select Object (1)' with a red cross icon.
- Properties:** Set 'Name' to 'PID0', 'Memory Section' to 'Input', and 'Data Type' to 'DWord'.
- Hardware Connection:** Shows a dropdown menu 'Select I/O Device (0)' with a red cross icon.

3. Click on **Select I/O device**.
4. Click on the local module **AI 4 / AO 4x14**.
5. Select a free IO.

## 8.3 Add tags



6. Click OK. The tag is added.

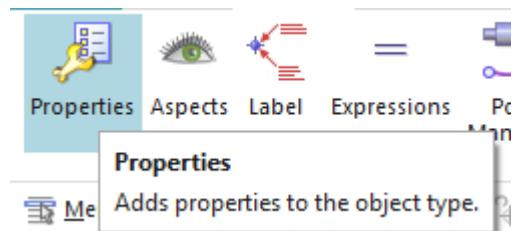


## 8.4

## Create TL constant value

The constant values in the top aspect Engineering Object TL can be referenced in Engineering Objects. This allows you to modify the constant value and the value changes for all Engineering Objects that reference that value.

1. Right-click on **TL01** and select **Edit Type**.
2. Click on **Properties**.



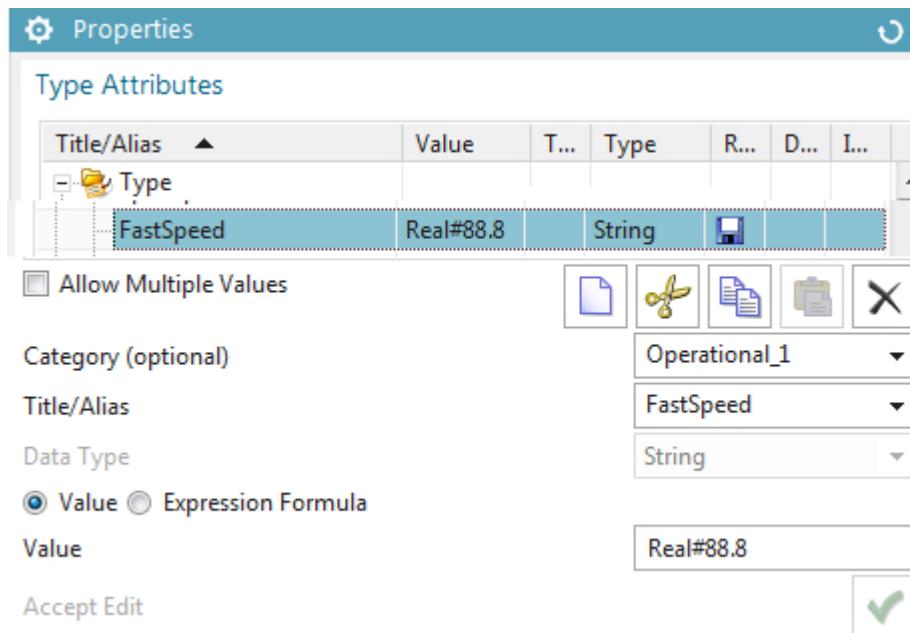
3. Create a property with following :

- Category = **Operational\_1**
- Title/Alias = **Fast\_Speed**
- Data Type = **String**
- Value = **Real#88.8**

Category (optional)	Operational
Title/Alias	Fast_Speed
Data Type	String
<input checked="" type="radio"/> Value <input type="radio"/> Expression Formula	
Value	Real#88.8
Add New Attribute	

4. Click the green arrow. The following shows the result.

## 8.4 Create TL constant value



5. Open the properties for the Engineering Object in the aspect tree.

The screenshot shows the 'Function Aspect Navigator' and the 'Properties' dialog for an engineering object.

**Function Aspect Navigator:**

- Name: CD000166;1-AD\_1\_CD\_4\_WS\_5\_1
- Unassigned
- =EOATMcc001
- =EOTLcc001
- =EOGLcc001
- =EOBGcc001
- =EOTFcc001
- =EOKfcc001
- =EOCcc001
- =PID0
- =EOMAcc001

**Properties Dialog:**

**Select Object:** Select Object (1)

**Context:** Interaction Method: Traditional

**Engineering Object Attributes:**

Title/Alias	Value	Type
FastSpeed	Real#88.8	String

Below the table are other configuration options:

- Category (optional): Operational\_1
- Title/Alias: FastSpeed
- Data Type: String
- Value  Expression Formula
- Value: Real#88.8
- Override Attribute:

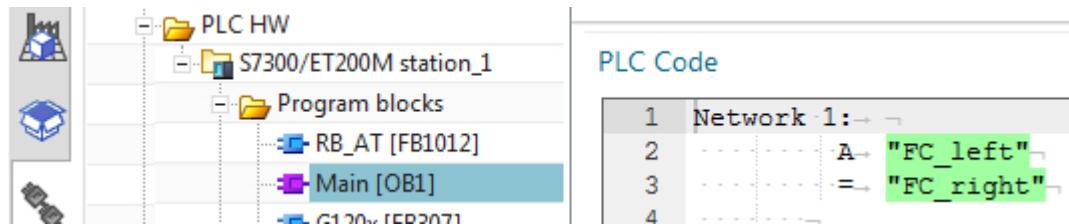
## 8.5 Dynamize software

You will now dynamize the calls (OB1, RB\_AT) and tag references (RB\_AT, PosDev) in the imported SW blocks.

### 8.5.1. OB1->RB\_AT\_DB replace by call

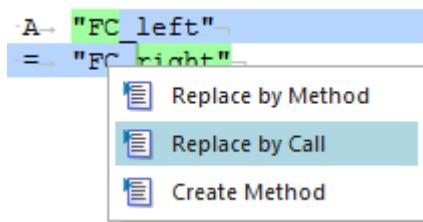
OB1 calls the RB\_AT IDB.

1. Double-click on **OB1**.



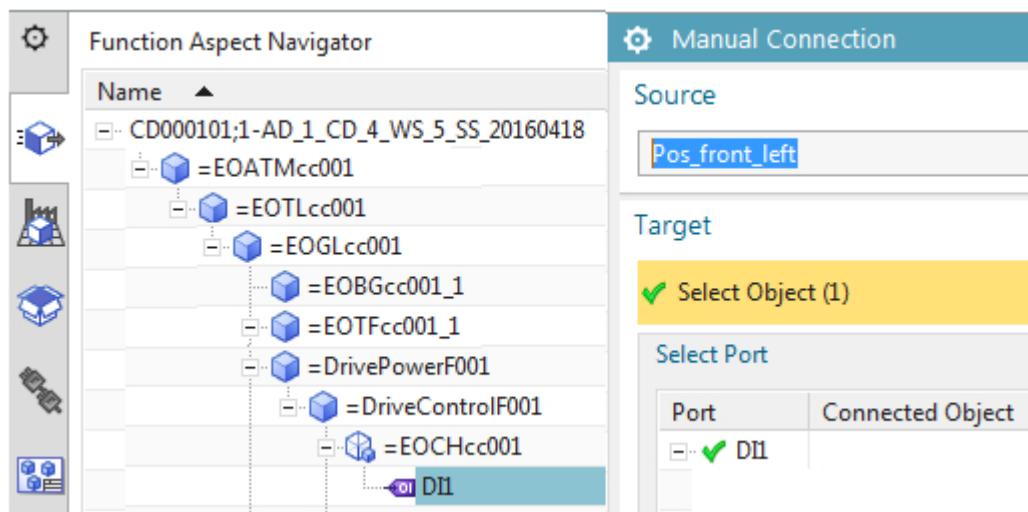
2. Select the lines of OB1 code.

3. Right-click. Select **Replace by Call**.

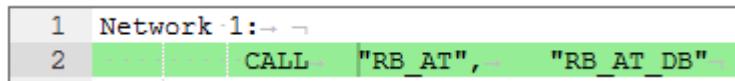


4. For Selection select **Object selection**.

5. For the program block select the RB\_AT\_DB IDB in the Function aspect.



6. Click **OK**. The call contains the symbolic names of the Function Block and IDB.



You can use ports to do the following:

- Define data flow, material flow, or flow of electrical current between objects. Flow ports specify which items flow between the Engineering Object and its environment, and in which direction.
- Plan subnets.
- Define plugs and sockets on EPLAN macros.
- Define a logical connection between two objects that gives these objects access to each others' properties, ancestors, and descendants. This logical connection can be evaluated by expressions, for example to link the property value of one object to a property value of a connected object.

Inside program blocks, you can use ports to, for example, link their global symbols to tags, to insert calls into their PLC code, or to replace operands.

7. Right-click on **OB Main**.

8. Select **Ports Manager**. A list of ports is displayed.

The following diagram shows the OB Main call to the RB\_AT IDB.

- OB004 = OB Main.
- The port on OB004 is named “Caller\_1” and is user defined.
- The port is connect to the port named “DB010” on object DB010 (RB\_AT\_IDB).

Source				
OB004				
Ports				
Port	Connected Obj...	Connected Port	Port Type	Connection Type
User Defined				
Caller_1	DB010	DB010	EO	Caller
				IDB_Proxy

9. List the ports for the RB\_AT IDB.

- DB010 = RB\_AT IDB.
- The port on DB010 is named “DB010” and is system-defined, since the call replacement was initiated from OB Main.
- OB004 = OB Main.

Source					
Ports					
Port	Connected Object	Connected Port	Port Type	Connection Type	
User Defined					
System Defined					
DB010	OB004	Caller_1	EO	IDB_Proxy	Caller

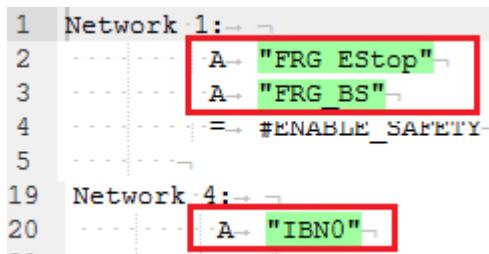
10. List the ports for the RB\_AT FB. This shows the port that was created earlier when you created the IDB.

- FB019 = RB\_AT FB.
- The port on FB019 is named “FB019” and is system-defined.
- DB010 = RB\_AT IDB and the connected port is named “RB\_AT”.

Source					
Ports					
Port	Connected Object	Connected Port	Port Type	Connection Type	
User Defined					
System Defined					
FB019	DB010	RB_AT	EO	FB_Proxy	FB

### 8.5.2. RB\_AT manual connect to RB\_AT AUTOMATION tags (FRG\_EStop)

The RB\_AT automation tags are already connected.



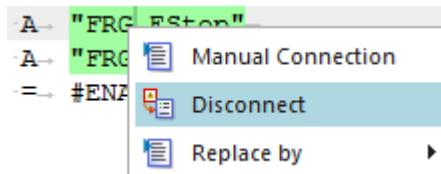
The following shows the ports manager for RB\_AT.

Port	Connected Object	Connected Port	Port Type	Tag	Undirected	1	Tag_Proxy
FRG_EStop	FRG_EStop	FRG_EStop	EO	Tag_Proxy	Undirected	N	Tag, Any, Operand
FRG_BS	FRG_BS	FRG_BS	EO	Tag	Undirected	1	Tag_Proxy
				Tag_Proxy	Undirected	N	Tag, Any, Operand

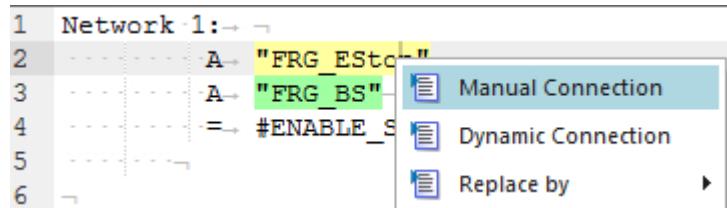
The following shows you how to make this connection.

1. Disconnect.

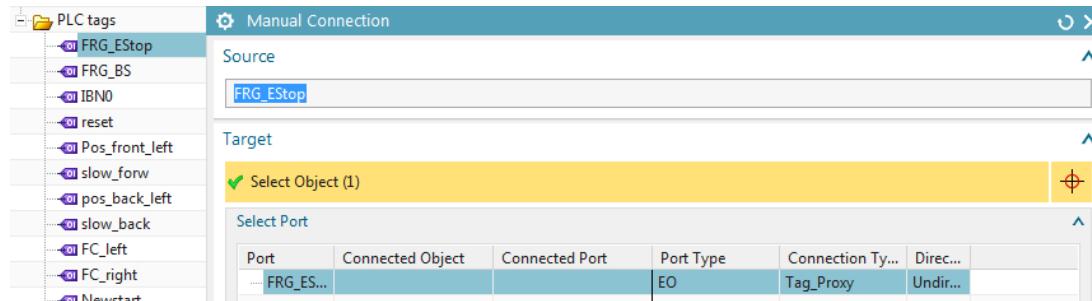
## 8.5 Dynamize software



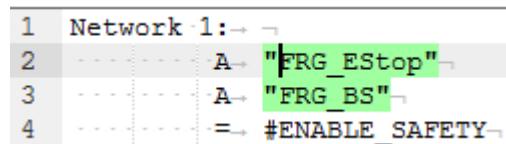
2. Select Manual Connection.



3. Select FRG\_EStop.



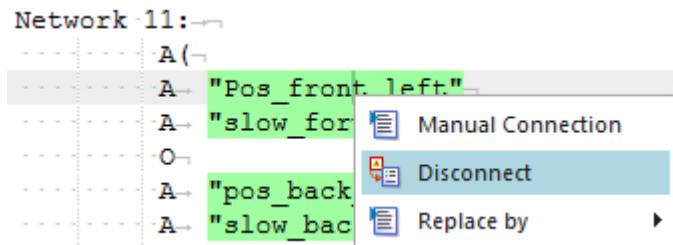
4. Click OK.



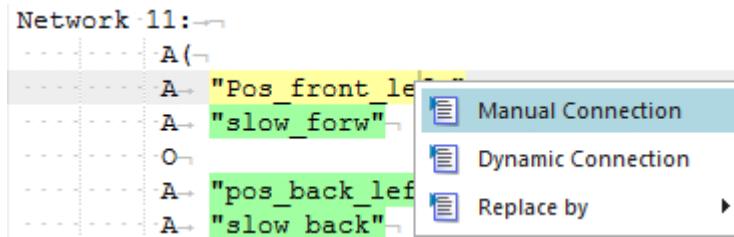
### 8.5.3. RB\_AT manual connect to CH DI tags

This section shows you how to make a connection to the channel tag.

1. Disconnect the existing connection for **Pos\_front\_left**.



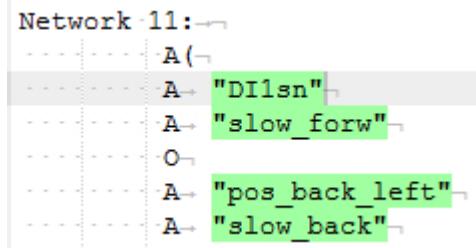
2. Right-click on **Pos\_front\_left** and select **Manual Connection**.



3. For the target object select DI1.

The screenshot shows the Function Aspect Navigator on the left and the Manual Connection dialog on the right. The Function Aspect Navigator lists various objects like Unassigned, \_001, \_004, ConveyorF001, MotorF001, SensorF001, DrivePowerF001, DriveControlF001, EOCHcc001, DI1, and G120x. The Manual Connection dialog has 'Pos\_front\_left' selected in the Source field and 'Select Object (1)' in the Target field. A table titled 'Select Port' shows a connection from Port DI1 to Connected Object S7-300-Station\_2, with Port Type INTERNAL, Connection Type Tag\_Proxy, and Direction Undir...

The following shows the result.



The following shows the RB\_AT FB port.

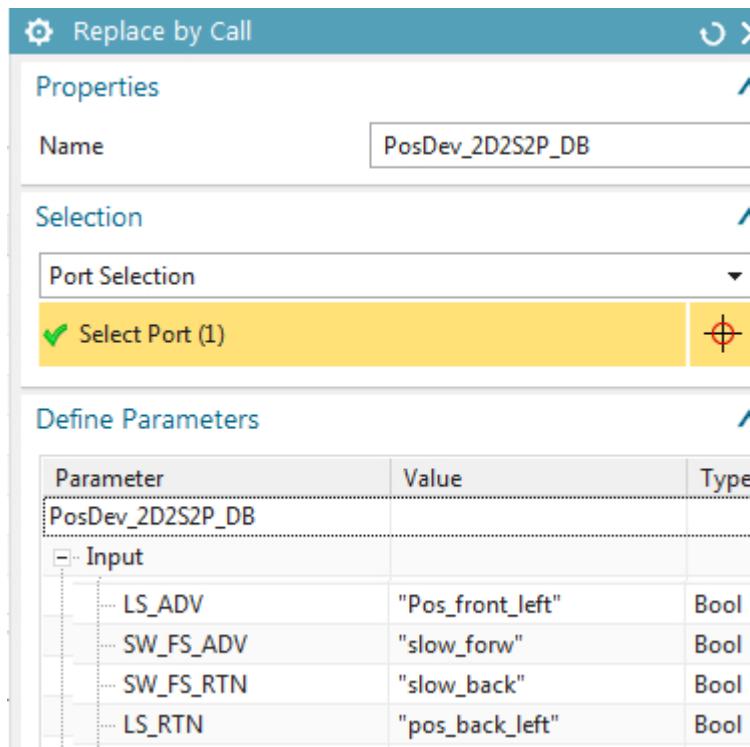
Port	Connected Object	Connected Port	Port Type	Connection Ty...	Dir...
DI1	S7-300-Station_2	I12.0	INTERNAL	Tag_Proxy	Undir...

#### 8.5.4. RB\_AT->PosDev replace by call

You now want to create your first replace by call. You use this to create a call from RB\_AT to Pos\_Dev. You will also reference the DI tag in a call.

1. Select the call to **PosDev** code.
2. Right-click and select **Edit**. Note the value of LS\_ADV.

## 8.5 Dynamize software



3. For Replace Parameter by select Symbolic Reference.

4. For Object Selection select DI1.

Parameter	Value	Type
PosDev_2D2S2P_DB		
Input		
LS_ADV	DI1sn	Bool

5. Click OK.

```

CALL "PosDev_2D2S2P", "PosDev_2D2S2P_DB"
... LS_ADV := "DI1sn"
... SW_FS_ADV := "slow_forw"
... SW_FS_RTN := "slow_back"
LS_RTN := "pos_back_left"

```

### 8.5.5. RB\_AT->G120x replace by call

You now want to create your second replace by call. You use this to create a call from RB\_AT to G120x. You will also reference the PID0 tag and the constant value you created earlier.

1. For G120x change the value for INPUT\_ADDR to PID0.

The screenshot shows the Function Aspect Navigator on the left and the 'Replace by Call' dialog box on the right.

**Function Aspect Navigator:**

- Name: CD000166;1-AD\_1\_CD\_4\_WS\_5\_SS\_201601
- Unassigned
- =EOATMcc001
  - =EOTLcc001
    - =EOGLcc001
      - =EOBGcc001\_1
      - =EOTFcc001\_1
        - =EOKFcc002
          - =EOCHcc001
            - oi D11
            - G120x\_DB
            - oi PID0
  - =EOMAcc001
  - RB\_AT
  - RB\_AT\_DB
  - PosDev\_2D2S2P\_DB

**Replace by Call Dialog Box:**

**Properties:**

- Name: G120x\_DB

**Selection:**

- Port Selection
- Select Port (1)

**Define Parameters:**

Parameter	Value	Type
G120x_DB		
Input		
INPUT_ADDR	PID0	DWord
I_M		DWord
A_F		DWord
FAST_SPEED	Real#88.8	Real
SLOW_SPEED	REAL#20.0	Real
FWD		Bool

**Replace Parameter by:**

Symbolic Reference

**Object Selection:**

- Select Object (1)

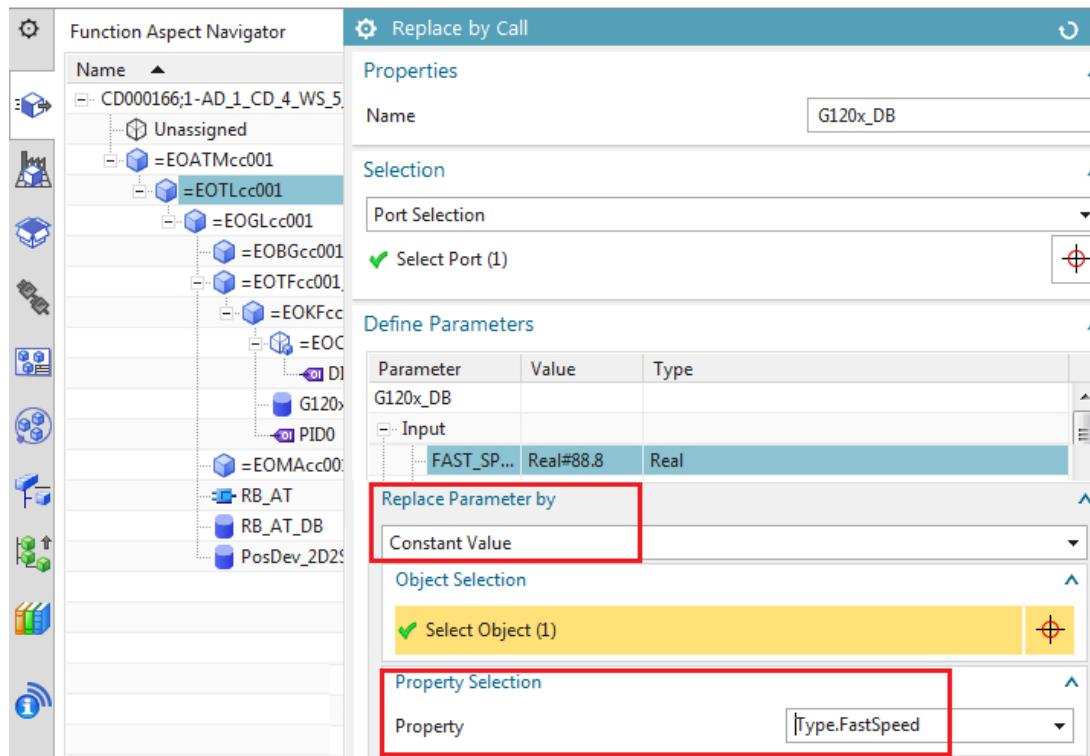
**Expression:**

**Return Value:**

**Break Expression:**

2. Change the value of FAST\_SPEED to Fast\_Speed.

## 8.5 Dynamize software



The following shows the result.

```
CALL-> "G120x",-> "G120x_DB"
INPUT_ADDR := | "PID0sn"
FAST_SPEED := -- Real#88.8
```

#### 8.5.6. PosDev manual connect to Automation tags

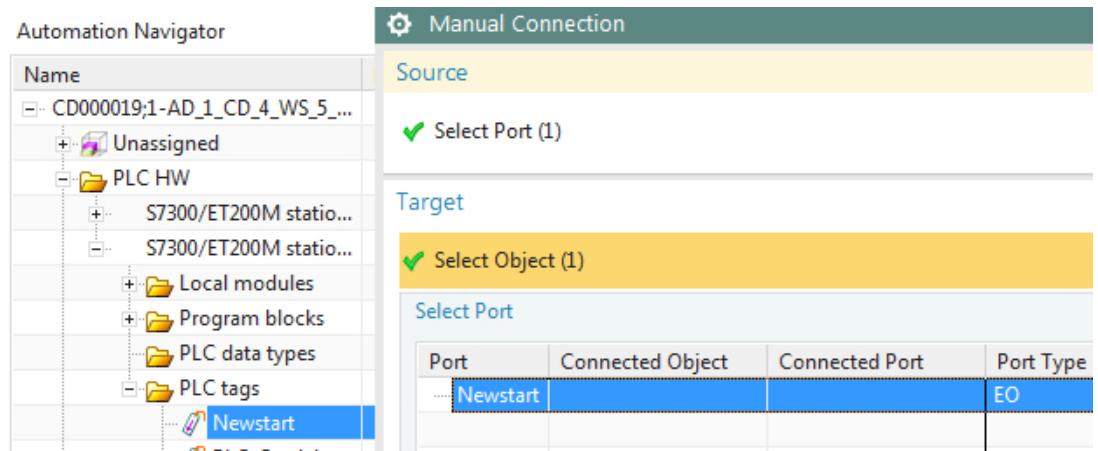
The PosDev automation tags should already be connected.

```
5 Network 2:->
6   A-> "Newstart"
7   R-> #TM_STARTUP-
8   R-> #EN_FAST-
9
10 Network 3:->
11   A-> #ERR_RESET-
12   FP-> #Err_Reset_P-
13   ON-> "PLC_On delayed"
```

				Tag	Undirected	1
	✓ Newstart	Newstart	Newstart	EO	Tag_Proxy	Undirected
					Tag	Undirected
	✓ PLC_On delayed	PLC_On delayed	PLC_On delayed	EO	Tag_Proxy	Undirected

If they are not connected, the do the following:

1. Disconnect.
2. Select **Manual Connection**.
3. Select **Newstart**.



4. Click **OK**.
5. Click **OK**. The tag reference has been connected to the tag.

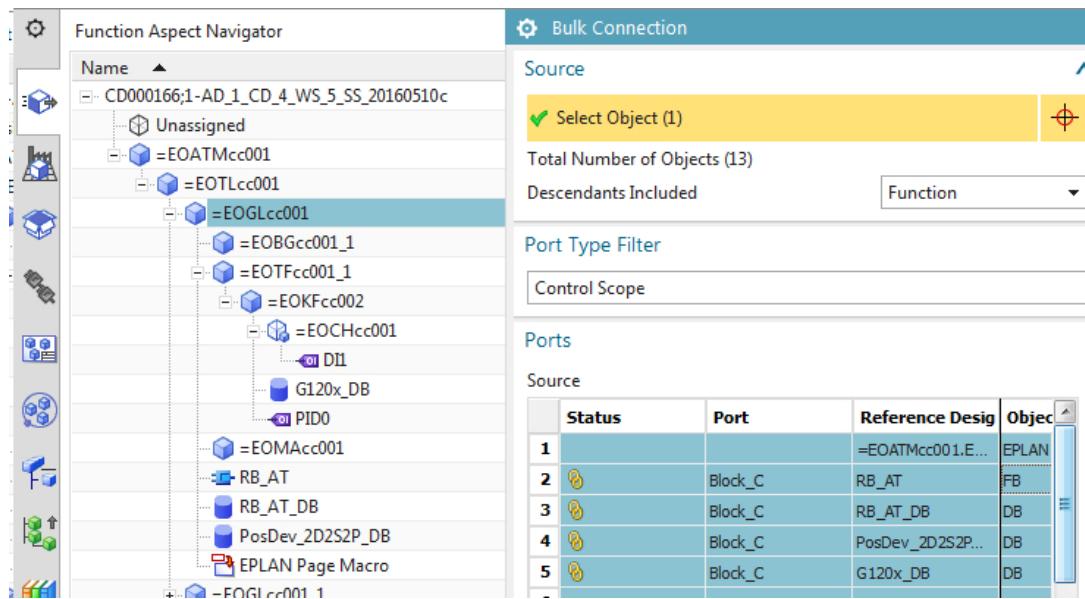
Configurations		PLC Code
Name	Value	
Global Symbols		
Tags		
Newstart	Newstart	1 Network 1: → ... TAR1 → #SAVE_AR1
PLC_On delay	PLC_On delay	3 ... TAR2 → #SAVE_AR2
TRI IF	TRI IF	4
		5 Network 2: → ... A → "Newstart"
		6 ... n #mu smanm
		7

## 8.6 Assign software to hardware

You have created the software blocks. But you still need to assign these blocks to your hardware in the automation tab. After this you can export to TIA Portal.

1. Select **Bulk Connection**.
2. For **Source** select **GL**.
3. Select the ports.

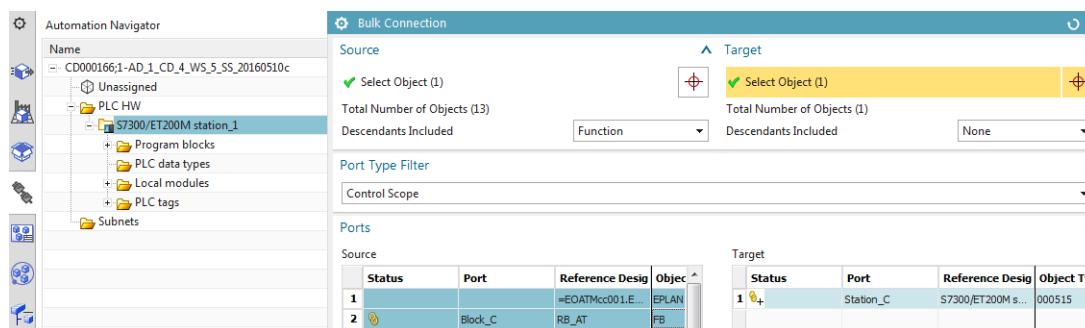
## 8.6 Assign software to hardware



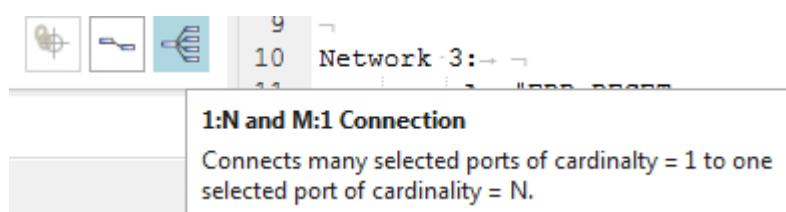
4. Under Target select Select Object.

5. Select the station.

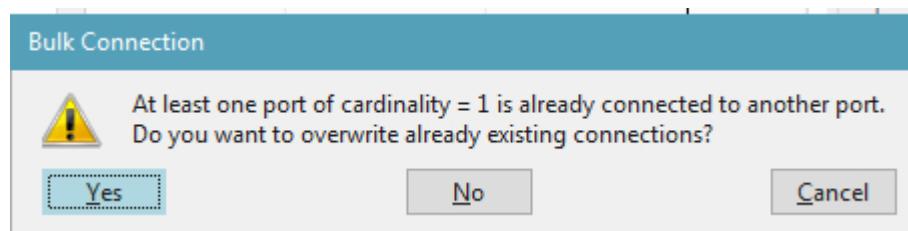
6. Select the target ports.



7. Click on 1:N and M:1 Connection.



8. The bulk connection message appears. Click OK.



The following shows the result.

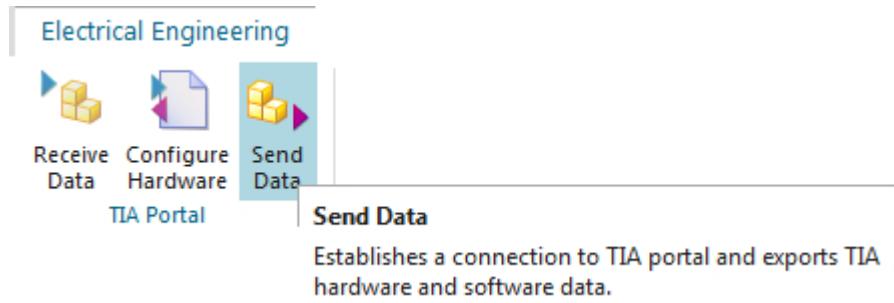
## 8.7

## Send data to TIA Portal

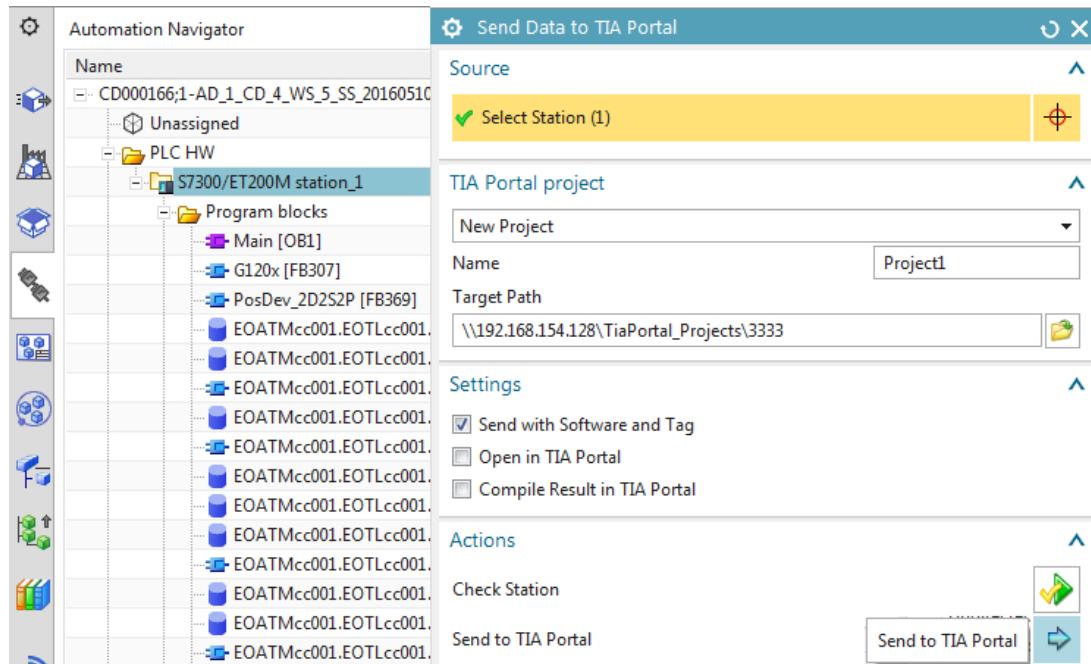
You now send the data to TIA portal.

1. Click **Send Data**.

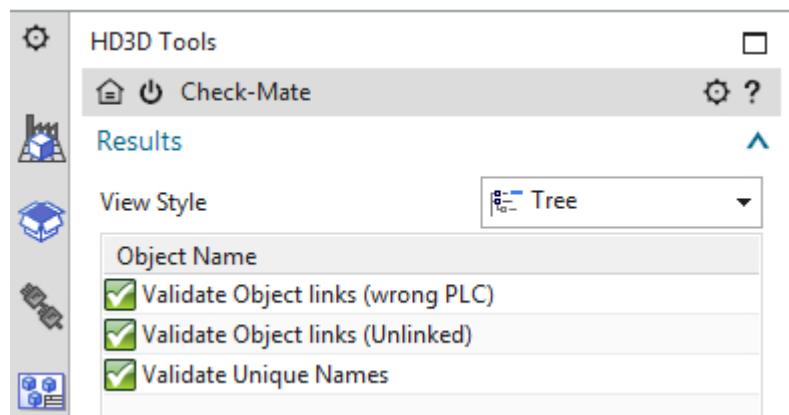
## 8.7 Send data to TIA Portal



2. Select the station.
3. Select **New Project** and enter the project name.
4. Select the target path.
5. Check **Send with Software and Tag**.



6. Run Checkmate to check that all software and tag variables are unique.



7. Click **Send to TIA portal**. A project is created in TIA Portal.



# Template-related concepts

*Templates* are reusable, ready-made solutions that reduce the complexity of engineering decisions to choosing between prepared solutions. They allow you to take an engineering solution from one project and to reuse it in the same project and in other projects.

## 9.1

### Getting aspect chain ID of parent EO using expressions

You need to get the aspect chain ID of the parent Engineering Object for EPLAN macro properties and TIA Portal software block and tag symbolic names.

#### EPLAN

Previously you simply entered text for the names of macro device properties. Entering the values manually is not a good idea because

1. Moving the parent Engineering Object would invalidate the aspect.
2. Instantiating a template or copying a conveyor with same values would result in EPLAN reports with duplicate names.

The solution is to use expressions to make the value of the **Function** property depend on the Engineering Object (GL) the macro was placed under in the aspect tree. Thus the property value is unique. This is not required to generate a report, but is required to avoid confusion when actually running the wiring on the factory floor. The following shows the expression.

	Name	Formula	Value	Units	Dim	Type	Source
1	p0	subString(p2,2,1000)	EOATMcc001.EOTLcc001.EOGLcc001			String	
2				mm	Len	Number ▾	
3	p2	(Attribute)	=EOATMcc001.EOTLcc001.EOGLcc001			String	(EOGLcc001

The resulting property value is the following.

Function EOATMcc 001.EOTLcc001.EOGLcc002

#### TIA Portal (software blocks and tags)

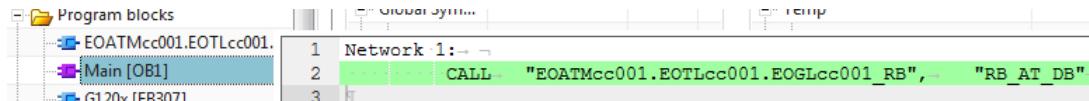
Previously you created only one conveyor, so all software block and tag names were globally unique. You did not need to use the aspect chain. But if you add or copy a conveyor with the same values, the symbolic names of software blocks and tags will be repeated (which will generate an error in TIA Portal).

The solution for the software is to use expressions to make the symbolic name of RB\_AT FB depend on the Engineering Object (GL) the Function Block was placed under in the aspect tree. Thus the symbolic name is globally unique (as required in TIA Portal). The following shows the expression.

## 9.2 Getting aspect chain ID of non-parent: 2 Ports + link

	Name	Formula	Value	Type
1	p0	subString(p4,2,1000)+"_RB"	"EOATMcc001.EOTLcc001.EOGLcc001_RB"	String

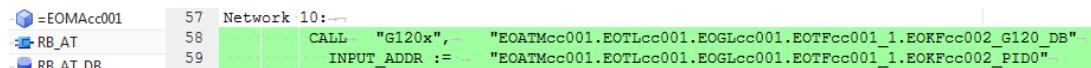
The resulting symbolic name for the Function Block is the following.



The solution for tags is to use expressions to make the symbolic name of the tags (PID0 and DI1) depend on the Engineering Objects (KF and CH) they were placed under in the aspect tree. Thus the symbolic names are globally unique (required in TIA Portal). The following shows the expression for PID0.

	Name	Formula	Value	Type
1	p0	subString(p2,2,1000)+"_PID0"	"EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_PID0"	String

The resulting symbolic name for PID0 is the following.

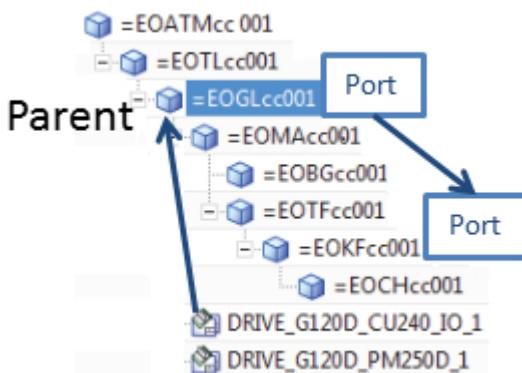


## 9.2

## Getting aspect chain ID of non-parent: 2 Ports + link

Sometimes you need to get the aspect chain ID of an Engineering Object that is not a direct parent. You can't use expressions to access a non-parent Engineering Objects.

The solution is shown in the following diagram. The macros access "neighbor" Engineering Object KF using a link between a port on the parent GL and a KF port.



To configure this you do the following:

1. Create a port for parent Engineering Object GL.
2. Create a port for target Engineering Object KF.
3. Link the ports.

## 9.2 Getting aspect chain ID of non-parent: 2 Ports + link

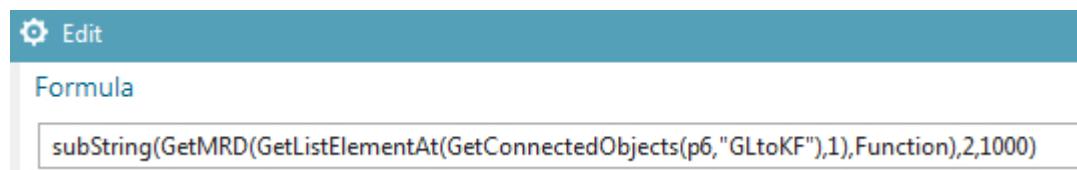
4. Reference the parent Engineering Object port name in a property expression.

**EPLAN**

Since the Engineering Objects representing the devices are not the parent Engineering Objects of the macro, you do not have a link to the Engineering Objects to retrieve their Multi Reference Designation.

You must use expressions and ports to get the aspect ID of KF. The steps are

1. Create ports GLtoKF and KFtoGL port (GL is parent of the macro).
2. Link the ports.
3. Create an expression for the macro that returns the KF aspect chain (p6 is "Object name" of GL).

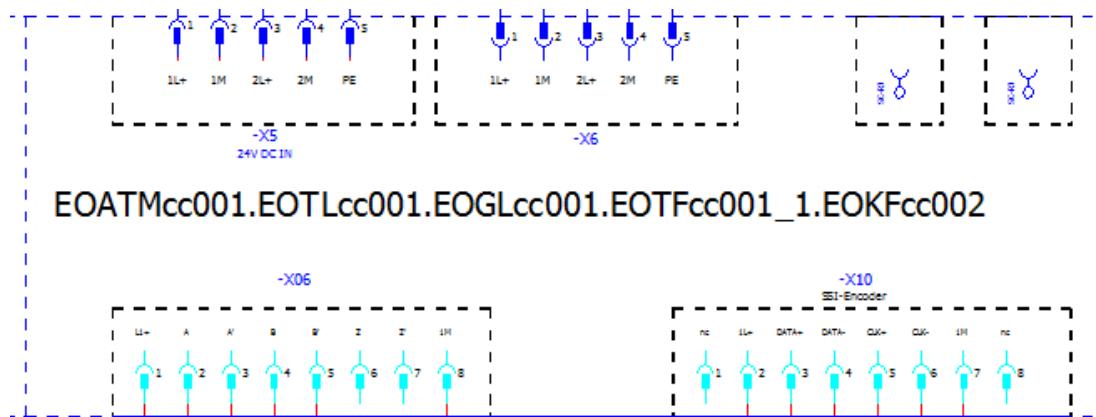


```
subString(GetMRD(GetListElementAt(GetConnectedObjects(p6,"GLtoKF"),1),Function),2,1000)
```

The following shows the resulting property value.

Variable: PowerModuleFunctionText	EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002
-----------------------------------	--

The following shows the result in the generated macro report.

**TIA Portal (software blocks and tags)**

In this Getting Started you don't have an example where a software block uses a port to access the aspect ID of a non-parent Engineering Object to use as the symbolic name. Normally you would not do this.

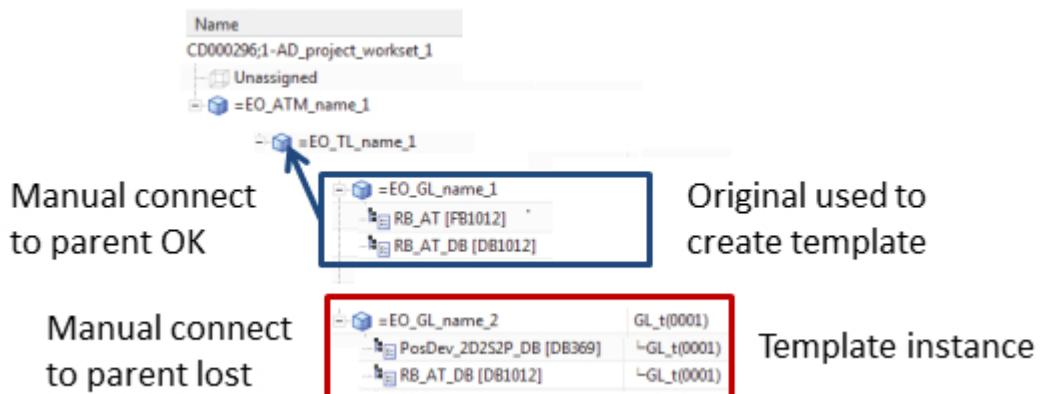
For tags you might want to have the symbolic name for a tag set to the aspect chain of a non-parent Engineering Object. For example, for the DI input tag for the boolean data from the light sensors could use the aspect chain ID of the physical sensor as the symbolic name of the tag. This Getting Started does not include such an example.

## 9.3 Getting aspect chain ID outside template with dynamic connection (software only)

Sometimes you need to get the aspect chain ID of an Engineering Object that will be outside an instantiated template. You cannot use a simply manual connection.

### Problem

The following diagram shows the Engineering Objects added to the aspect chain for a conveyor (top half). These Engineering Objects (GL and below) were then used to create a template. The template was then instantiated to create the aspect chain for a second conveyor (bottom half). These EO's (GL and below) were then used to create a template. The template was then instantiated to create the aspect chain for a second conveyor (bottom half).



The problem is if a software block (or tag) in the aspect chain of the first conveyor had a link to an Engineering Object outside of the conveyor aspect chain, then this link information would not be valid if you instantiated the template. The link in the template needs to be defined with expressions that specify the relative location of the parent outside of the template.

This is also true for macros, but macros don't usually need access to an Engineering Object outside of the conveyor aspect chain.

### Solution

With a dynamic expression the template instance automatically locates the required parent EO TL01.



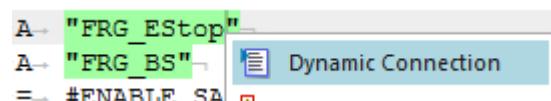
To configure the dynamic connection.

1. Create the TL port.
2. Create a manual connection from TL to the tag. The tag does not require a port.
3. Create the dynamic connection from RBAT FB to the tag via the TL port using the following expressions (P6 is RB\_AT object name).

	Name	Formula	Value	Type
2	aaa	nth(3,GetAncestors(p6,Function))	"EOTLcc001"	String
3	bbb	First(GetConnectedObjects(aaa,"TLtoFRGEStop"))	"ST001.Tag65"	String
5	p6	🔒 (Attribute)	"FB001"	String

4. Copy this GL or create a template and instantiate the template.

The result is a dynamic connection to FRG\_Estop.



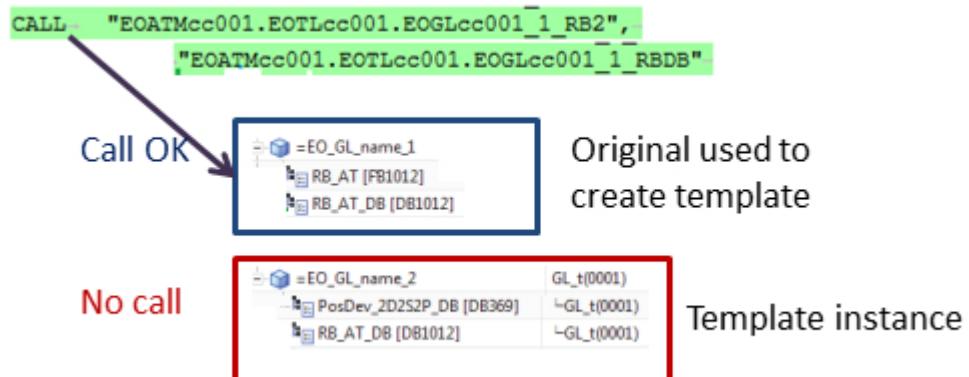
## 9.4

## Automatic generation of calls for inserted software

After you have copied GL or instantiated the template, OB Main needs to have a call to the copy or new instance.

### Problem

When you copy a conveyor or instantiate a conveyor template then OB main will not have a call to the RB\_AT DB in the new conveyor.



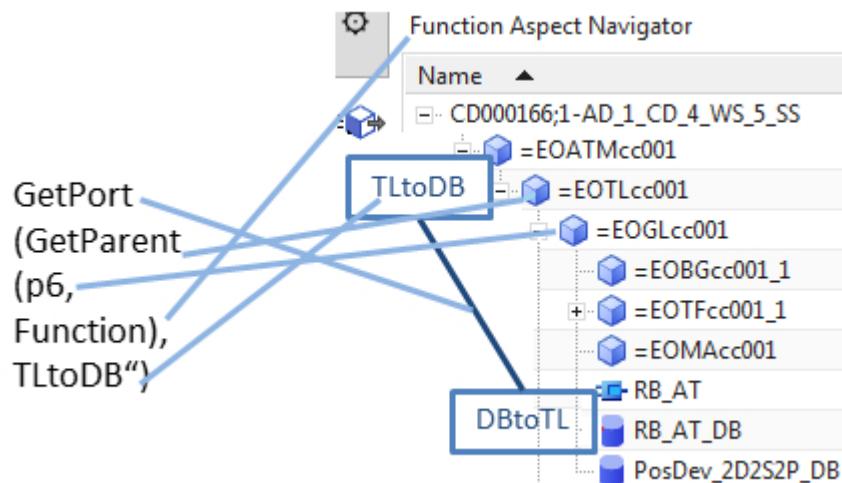
### Solution

You need to enable OB Main to automatically add a call to an added GL. To do this:

1. Define ports TLtoDB (in TL) and DBtoTL (in RBAT IDB).
2. In RBAT DB create a connection between the ports with the following expressions (p6 is the GL object name).

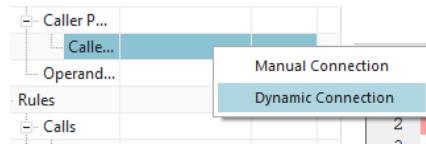
	↑ Name	Formula	Value	Type
3	ccc	GetPort(ddd,"TLtoDB")	"EOTLcc001.Port2"	String
4	ddd	GetParent(p6,Function)	"EOTLcc001"	String
8	p6	🔒 (Attribute)	"EOGLcc001"	String

This diagram shows the expression components.



3. Change the OB main -> RBAT DB call port (p1 is the TL object name) with the following expressions.

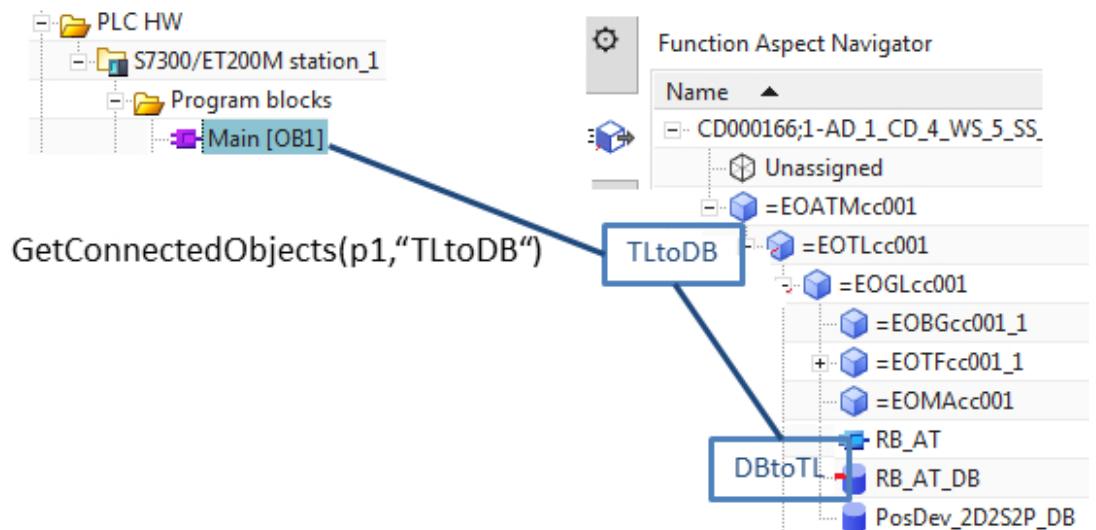
## 9.4 Automatic generation of calls for inserted software



The screenshot shows a table with two rows. Row 2 contains a call named 'bbb' with the formula 'GetConnectedObjects(p1,"TLtoDB")'. Row 3 contains an operand named 'p1' with the value 'EOTLcc001'. The 'Calls' column shows a connection status.

	Name	Formula	Value	Type	Source
2	bbb	GetConnectedObjects(p1,"TLtoDB")	{"DB001"}	List	
3	p1	(Attribute)	"EOTLcc001"	String	(EOTLcc001::Engine)

The following diagram shows the connections.

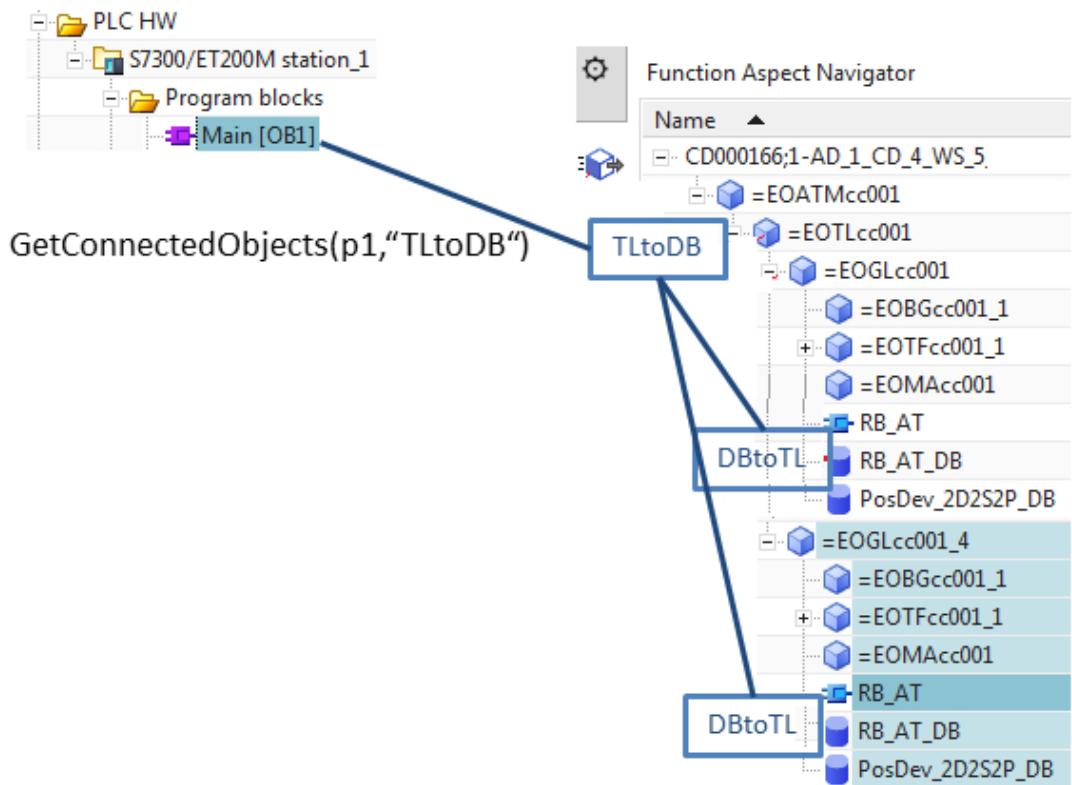


4. When you copy GL or instantiate the template, the OB main calls will automatically update.

```
CALL "EOATMcc001.EOTLcc001.EOGLcc001_1_RB2", "EOATMcc001.EOTLcc001.EOGLcc001_1_RBDB"
CALL "EOATMcc001.EOTLcc001.EOGLcc001_RB", "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

The following diagram shows the connections.

9.4 Automatic generation of calls for inserted software



# Configure template-ready EPLAN

The EPLAN configuration you created earlier needs to be modified in order to work with templates. Now you will configure a template-ready project for EPLAN by creating the following.

- 10.1. Function expression
- 10.2. KF01.name ports, link, and expression
- 10.3. Test

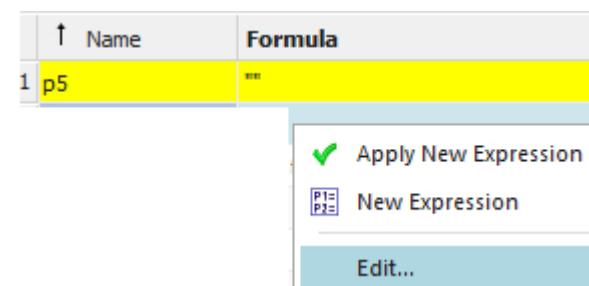
## 10.1 Function expression

This section shows a simple example of how to use the parent Engineering Object aspect tree for the value of an EPLAN property.

1. First add the expression for the property Function of the macro. Open the properties for the macro. Under **Type** select **Function**. Select **Expression Formula**. Right-click on the drop-down list and select **Formula...**

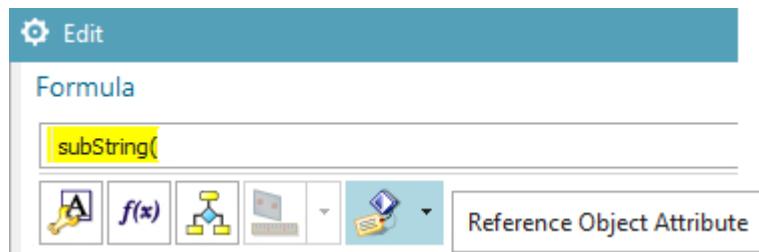


2. Click **Edit**.



3. Enter "subString(" .

## 10.1 Function expression



4. For Reference Object Attribute select in the Function aspect the Engineering Object GL Multi Reference Designation.

Title/Alias	Value	Units	Dim	Type	R...	D...	I...
Designated	True			Boolean	<input checked="" type="checkbox"/>		
Designation	EOGLcc001			String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Multi-level Reference Designation	=EOATMcc001.EO...			String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

5. Add the formula “,2,1000)”. This specifies to create a substring from character 2 that includes the first 1000 characters.

6. Click OK.

	Name	Formula	Value	Units	Dim	Type	Source
1	p0	subString(p2,2,1000)	"EOATMcc001.EOTLcc001.EOGLcc001"			String	
2				mm	Len	Number	
3	p2	(Attribute)	"=EOATMcc001.EOTLcc001.EOGLcc001"			String	(EOGLcc001)

The following shows the result.

Function EOATMcc 001.EOTLcc001.EOGLcc002

The following diagram shows the value for “Function” in a generated report.



## 10.2 KF01.name ports, link, and expression

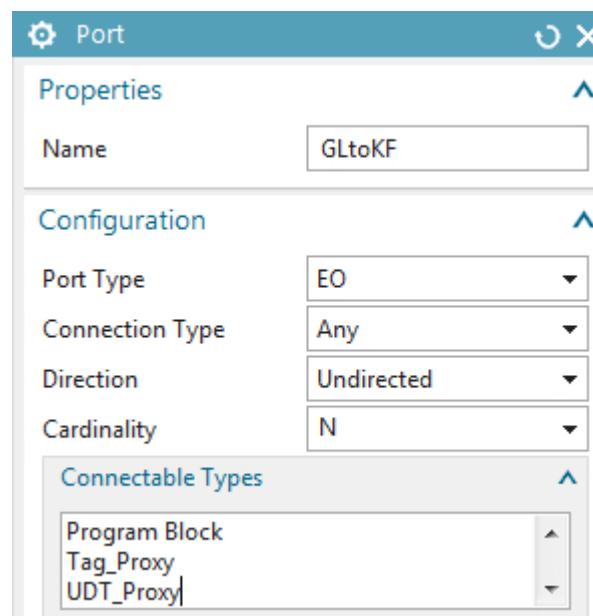
The value for KF01.Function text is taken from the corresponding Engineering Object KF (not the parent Engineering Object). Therefore you must create and connect ports for the parent Engineering Object and Engineering Object KF. You do this using ports, connections, and expressions.

### Ports

1. Right-click on EO GL and select **Create Port....**



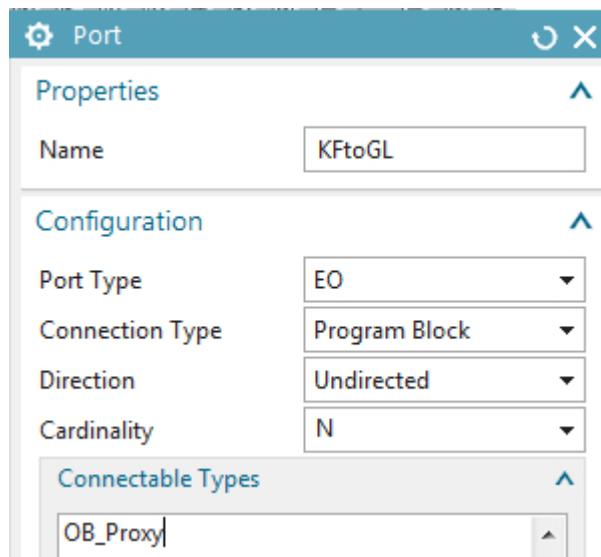
2. Enter the GLtoKF port information.



3. Click **OK**.

4. Right-click on Engineering Object KF and select **Create Port....**

5. Enter the KFtoGL port information. For Connection Type do not use **Any**.



6. Click OK.

#### Connection (link)

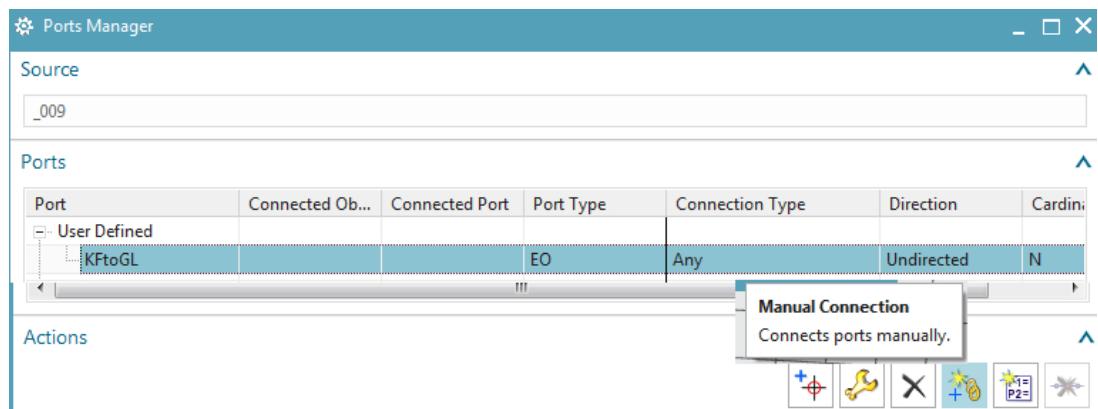
You connect a source port to a target port by creating a manual or dynamic connection.

- A *manual connection* is a connection for which you manually select the target port.
- A *dynamic connection* is a connection that you link to a navigation expression. The navigation expression returns the target port.

You now need to link the 2 ports with a manual connection, since this connection will be within the template you create later. The link can be created starting from either Engineering Object.

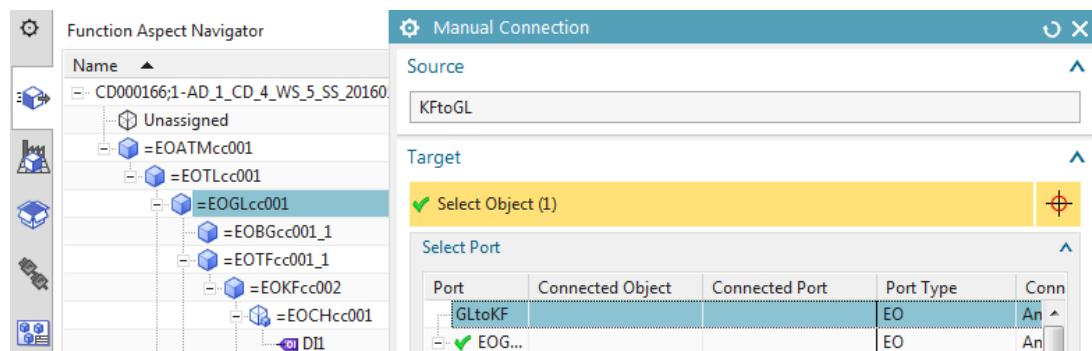
1. Right-click on Engineering Object KF and select **Ports manager....**

2. Right-click on the port and select **Manual connection**.



3. Select the target Engineering Object GL.

4. Select the port GLtoKF.



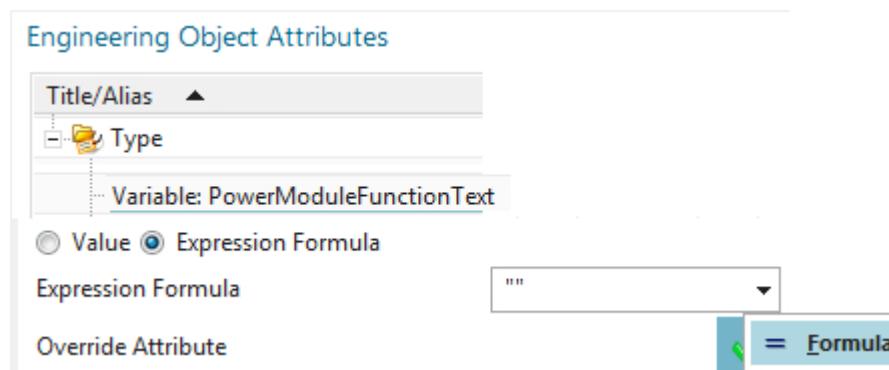
5. Click **OK**. The ports are linked.

Source							
EOKFc002							
Port	Connected Obj...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
<input checked="" type="checkbox"/> KFtoGL	EOGLcc001	GLtoKF	EO	Program Block	Undirected	N	OB_Proxy, FB_Proxy, FC_Proxy, Device Function
				Any	Undirected	N	OB_Proxy, Device Function

## Expression

This section shows how to use an expression to get the Function aspect Multi Reference Designation of an Engineering Object that is not a parent of the macro and assign to a macro variable.

1. Open the properties for the EPLAN macro.
2. For **Apply to** select **EPLAN page**.
3. Click on Power Module Function Text.
4. Click **Expression Formula**.
5. Click on the arrow for **Expression Formula**. A drop-down list appears.

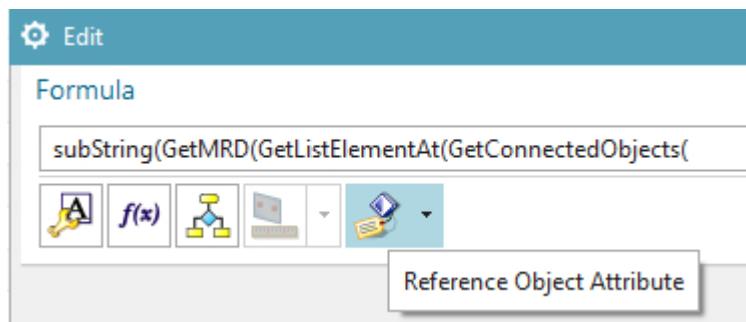


6. Click **Formula**. The Expressions dialog appears.
7. Under **Formula** right-click and select **Edit**.

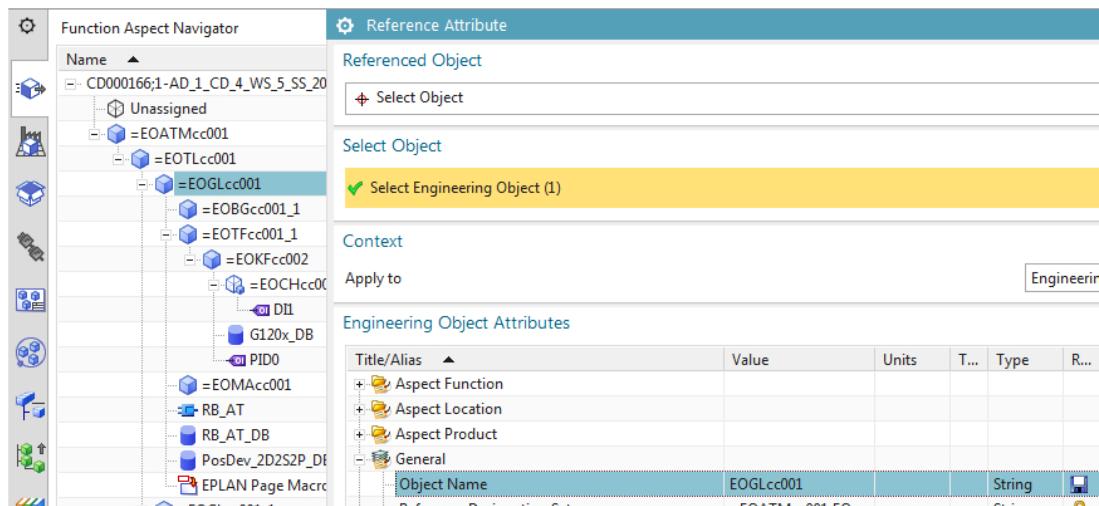
	Name	Formula
1	p4	"
2		New Expression
3	p0	subString(p2,2,1000)

8. Enter “subString(GetMRD(GetListElementAt(GetConnectedObjects(“.

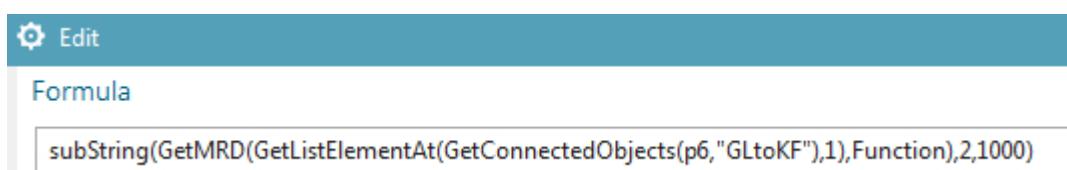
9. Click on **Reference Object Attribute**.



10. Select the conveyor **Object name**.



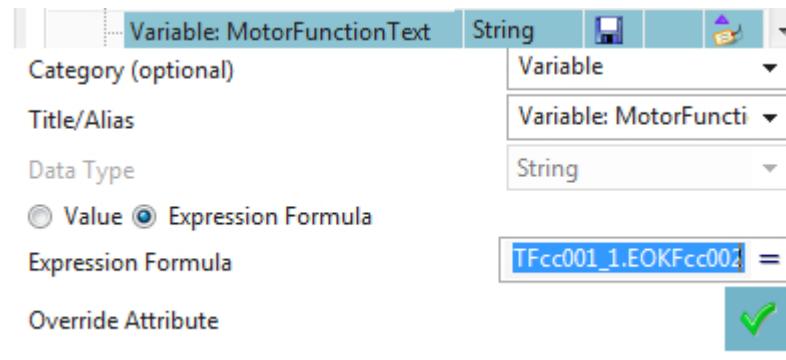
11. Complete expression with “,”GLtoKF”),1),Function),2,1000)”. This expression gets the connected objects at port GLtoKF, gets the list elements at that port, then gets the MRD, then returns the substring (without the leading “=” character).



12. Click OK.

Name	Formula	Value	Units	Dimensionality	Type	Source
p4	subString(Get#ID(GetListElementAt(GetConnectedObjects(p6,"Global"),1),Function),2,1000)	"EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002"			String	
			mm	Length	Number	
p0	subString(p2,2,1000)	"EOATMcc001.EOTLcc001.EOGLcc001"			String	(EPLAN Page Mac)
p2	\$(Attribute)	"=EOATMcc001.EOTLcc001.EOGLcc001"			String	(EOGLcc001;Fun)
p6	\$(Attribute)	"EOGLcc001"			String	(EOGLcc001;Eng)

13. Click OK.



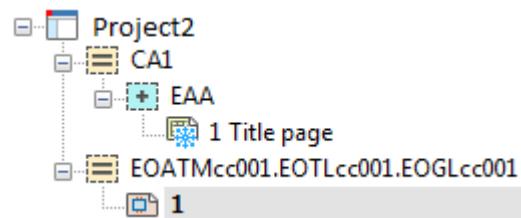
14. Click the green arrow. The following shows the result.

Variable: PowerModuleFunctionText      EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001\_1.EOKFcc002

15. Click OK.

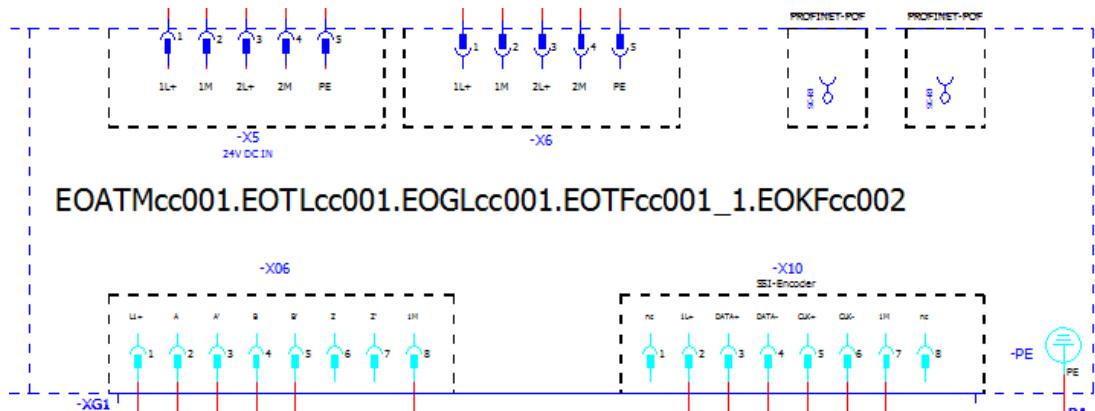
## 10.3 Generate

1. Generate EPLAN.



Note the property for KF "MotorFunctionText".

10.3 Generate



# Configure template-ready TIA Portal

You've tested basic SW generation in chapter 8 "Configure (non-template) TIA". Now you want to create a template the GL conveyor from chapter 8.

But you must modify the conveyor before creating the template. If you copy and paste another GL under TL, many SW and tag ID's for the 2 GL's will be the same. You need to make the symbolic names dependent on the aspect chain.

## 11.1 Configure symbolic names

In TIA Portal, tags and program blocks need unique symbolic names. In Automation Designer, the symbolic names of tags and program blocks are not unique by default. Create unique symbolic names before you send the tags and program blocks of a PLC station to TIA Portal.

When project engineers insert the template in a project, the symbolic name of each tag displays the function Multi Reference Designation of the drive, followed by a suffix with the tag's name. If the function Multi Reference Designation of the drive changes, Automation Designer updates the symbolic names automatically.

Automation Designer is delivered with a set of predefined validations, for example Validate Unique Names, which checks the uniqueness of symbolic names. When users carry out certain actions, Automation Designer runs these validations automatically. For example, before you export a PLC station to TIA Portal, you can check the entire PLC station. This check includes the Validate Unique Names validation.

You need to configure the following symbolic names

- 11.1.1. RB\_AT FB symbolic name
- 11.1.2. RB\_AT IDB symbolic name
- 11.1.3. PosDev\_DB symbolic name
- 11.1.4. G120 IDB symbolic name
- 11.1.5. PID0 symbolic name
- 11.1.6. DI1 symbolic name

### 11.1.1. RB\_AT FB symbolic name

The RB\_AT FB must have a symbolic name that is unique when a template is instantiated. You do this by using an expression to assign a name based on the Function aspect Multi Reference Designation of RB\_AT.

1. Open RB\_AT properties.
2. Select **Symbolic Name**.
3. Click **Expression Formula**.
4. Right-click on the drop-down box and select **Formula**.

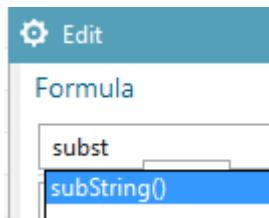
## 11.1 Configure symbolic names



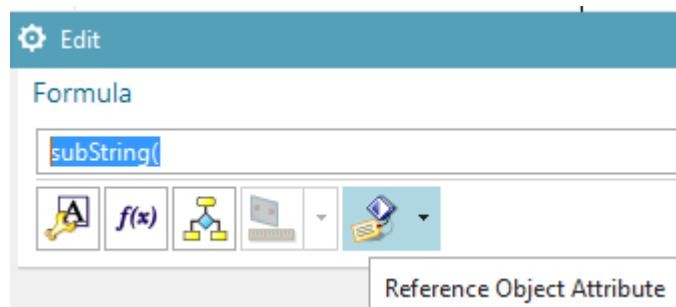
5. Right click in column **Formula**.

	Name	Formula
1	p0	""
2		<input type="text" value="P1= P2= New Expression"/> Edit...

6. Enter “subst” and accept suggestion.



7. Click **Reference Object Attribute**.



8. Select the GL Function aspect **Multi Reference Designation**.

P4 = GL Function Multi Reference Designation.

9. Complete the expression.

`subString(p4,2,1000)+"_RB"`

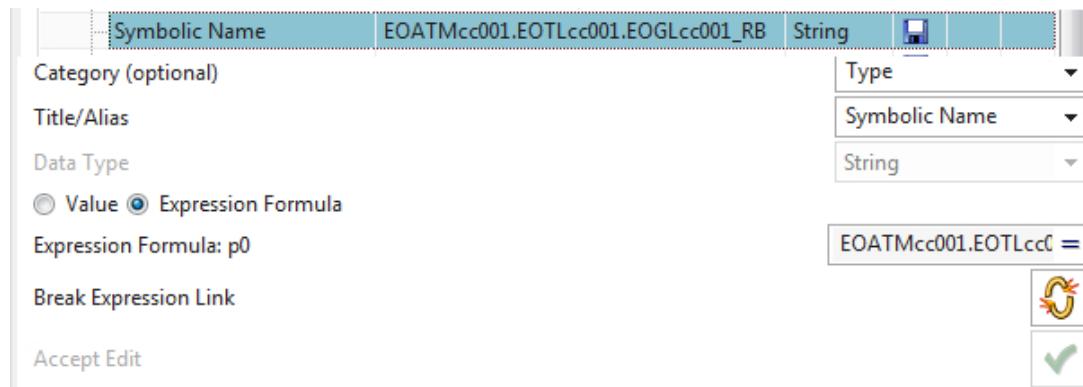
10. Click **OK**. The following shows the result.

	Name	Formula	Value	Type
1	p0	subString(p4,2,1000)+"_RB"	"EOATMcc001.EOTLcc001.EOGLcc001_RB"	String

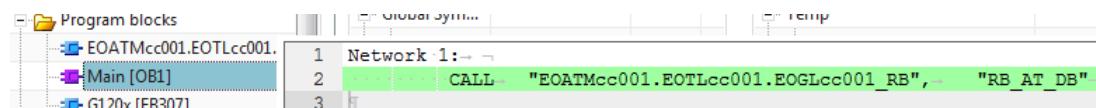
11. Click **OK**. The following shows the result.

12. Click the green arrow.

## 11.1 Configure symbolic names



13. Click **OK**. RB\_AT FB now has a globally unique name.



### 11.1.2. RB\_AT IDB symbolic name

The RB\_AT IDB must have a symbolic name that is unique when a template is instantiated. You do this by using an expression to assign a name based on the Function aspect Multi Reference Designation of RB\_AT IDB.

1. Modify the RB\_AT IDB symbolic name.

	Name	Formula	Value	Type
1	p0	subString(p2,2,1000)+"_RBDB"	"EOATMcc001.EOTLcc001.EOGLcc001_RBDB"	String

The following shows the result.

```
CALL "EOATMcc001.EOTLcc001.EOGLcc001_RB", "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

### 11.1.3. PosDev\_DB symbolic name

1. Modify the PosDev\_DB symbolic name.

	Name	Formula	Value	Type
1	p0	subString(p2,2,1000)+"_PD_DB"	"EOATMcc001.EOTLcc001.EOGLcc001_PD_DB"	String

The following shows the result.

```
=EUMAcc001 44 Network 9:-
=RB_AT 45 CALL "PosDev_2D2S2P", "EOATMcc001.EOTLcc001.EOGLcc001_PD_DB"
```

### 11.1.4. G120 IDB symbolic name

The RB\_AT IDB must have a symbolic name that is unique when a template is instantiated. You do this by using an expression to assign a name based on the Function aspect Multi Reference Designation of RB\_AT IDB.

1. Modify the symbolic name.

The screenshot shows the Function Aspect Navigator on the left and the Reference Attribute dialog on the right. In the Navigator, a tag named 'PID0' is selected under the 'EOGLcc001\_1' aspect. The Reference Attribute dialog shows the 'Select Object' step completed with 'Select Engineering Object (1)'. In the 'Engineering Object Attributes' table, the 'Multi-level Reference Designation' is set to '=EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001\_1.EOKFcc002\_G120\_DB' and the 'Name' is set to 'EOKFcc002'. Below this, a table shows the formula for the tag: 'subString(p2,2,1000)+"\_G120\_DB"' with a value of 'EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001\_1.EOKFcc002\_G120\_DB'.

Name	Formula	Value	Type
p0	subString(p2,2,1000)+"_G120_DB"	"EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_G120_DB"	String

The following shows the result.

```

57 Network 10:-
58 CALL "G120x", "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_G120_DB"

```

### 11.1.5. PID0 symbolic name

The PID0 tag must have a symbolic name that is unique when a template is instantiated. You do this by using an expression to assign a name based on the Function aspect Multi Reference Designation of the tag.

1. Modify the symbolic name.

The screenshot shows the Function Aspect Navigator and Reference Attribute dialog. The tag 'PID0' is selected in the Navigator. The Reference Attribute dialog shows the 'Select Object' step completed. In the 'Engineering Object Attributes' table, the 'Multi-level Reference Designation' is set to '=EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001\_1.EOKFcc002\_PID0' and the 'Name' is set to 'EOKFcc002'. Below this, a table shows the formula for the tag: 'subString(p2,2,1000)+"\_PID0"' with a value of 'EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001\_1.EOKFcc002\_PID0'.

Name	Formula	Value	Type
p0	subString(p2,2,1000)+"_PID0"	"EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_PID0"	String

The following shows the result.

```

57 Network 10:-
58 CALL "G120x", "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_G120_DB"
59 INPUT_ADDR := "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_PID0"

```

### 11.1.6. DI1 symbolic name

The DI1 tag must have a symbolic name that is unique when a template is instantiated. You do this by using an expression to assign a name based on the Function aspect Multi Reference Designation of the tag.

1. Modify the symbolic name.

## 11.2 FRGEStop dynamic connection

Name	Formula	Value	Type
1 p0	subString(p2,2,1000)+"_DI1"	"EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002.EOCHcc001_DI1"	String

The following shows the result.

## 11.2 FRGEStop dynamic connection

You can make manual connections or dynamic connections. Create dynamic connections if you are working in the template environment and the connected object is outside of the template scope, or if you are working in the project environment, intend to turn your engineering into a template, and the connected object is outside the future template scope.

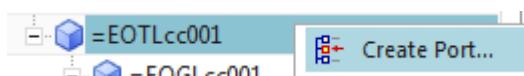
An inserted GL template needs to connect to the FRGEStop tag (in the automation tab). A simple manual connection will not work. You must create a dynamic connection.

A *dynamic connection* is a connection that you link to a navigation expression. The navigation expression returns the target port.

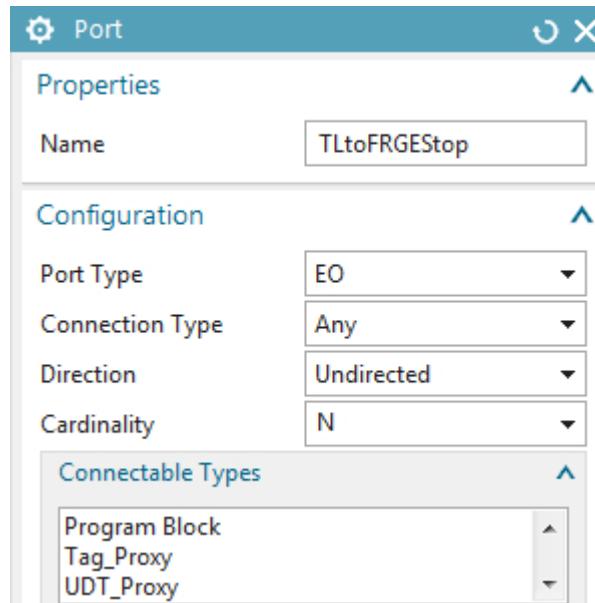
Use a dynamic connection to add a connection to an object outside of the template's scope. For example, to define the material flow between the conveyor and its successor, you must use a dynamic connection, since the successor and its ports are not available in the template environment. When the template is inserted in a project, the navigation expression of the dynamic connection is evaluated and returns the port of the conveyor that shall be the successor. Automation Designer then creates a connection between the ports of the conveyor at which you defined the dynamic connection and the successor. The *external structure* represents the template's interface. It consists of the horizontal relations of the template members to objects outside the template. In the template definition, template engineers use dynamic connections to define these relations. When a project engineer inserts the template in a project, Automation Designer evaluates the navigation expressions behind each dynamic connection and connects the template members to other objects from the project.

First create a port from TL to the tag.

1. Right click on TL and select **Create Port**.



2. Enter the following.



3. Click OK. The port is created.

4. Create a manual connection from TL to the tag. The tag does not require a port. Note that a manual connection is good enough, since TL is not in the template.

Port	Connected Obj...	Connected Port	Port Type
TLtoFRGEStop			EO
Port_1			

Port	Connected Obj...	Connected Port	Port Type	Conn
FRG_ES...	Tag_P	EOTLcc001	EO	

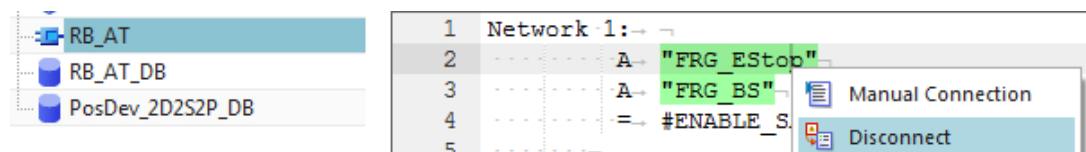
The following shows the result.

## 11.2 FRGEStop dynamic connection

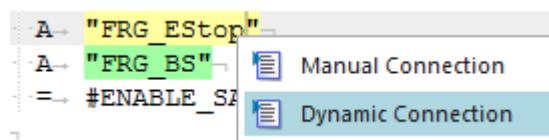
Port	Connected Obj...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable typ...
- User Defined							
- TLtoFRGEStop		FRG_EStop	EO	Any	Undirected	N	OB_Proxy, Device...

Now create the dynamic connection from RBAT FB to the tag via the TL port.

5. Disconnect the manual connection.



6. Create a dynamic connection.



7. Add an expression to get TL (P6 is RB\_AT FB object name).

Title/Alias	Value	Units	T...	Type	R...
Aspect Function					
General					
Family				String	
Object Name	FB001			String	

8. Add an expression to get the tag.

Name	Formula	Value
bbb	First(GetConnectedObjects(aaa,"TLtoFRGEStop"))	"ST001.Tag65"
aaa	nth(3,GetAncestors(p6,Function))	"EOTLcc001"

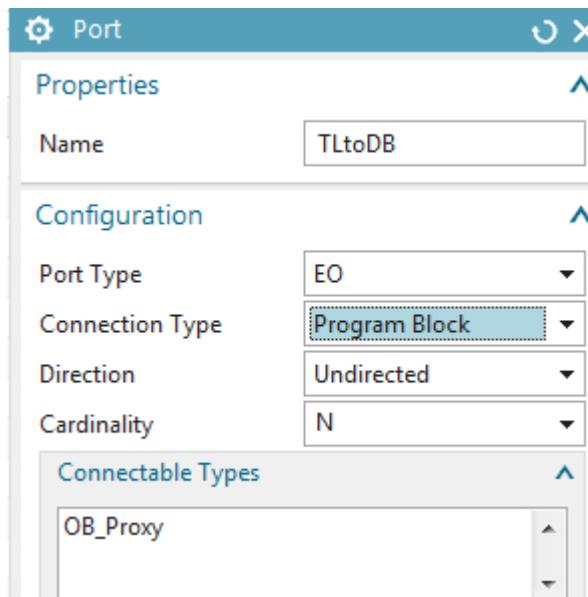
9. Click OK. FRG\_Estop is green because the connection was established.



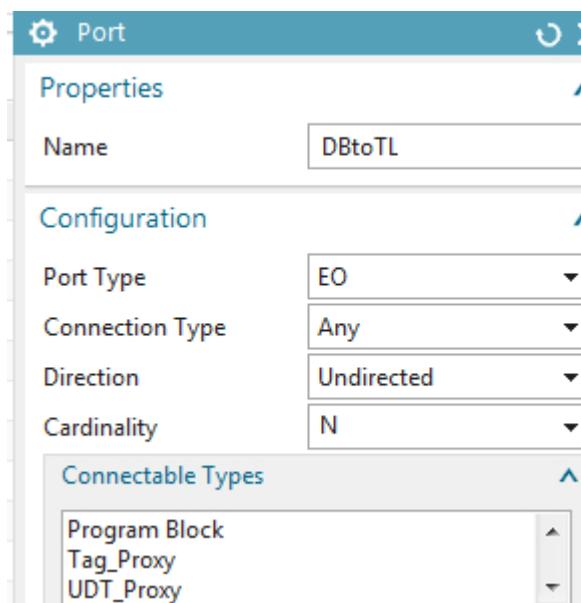
## 11.3 OB Main calls

Now you need to enable OB Main to automatically add a call to an added GL.

1. Create a port on TL named TLtoDB with Connection type = **Program Block** (or the type for the DBtoTL port is program block; you can not have both as **Any**).

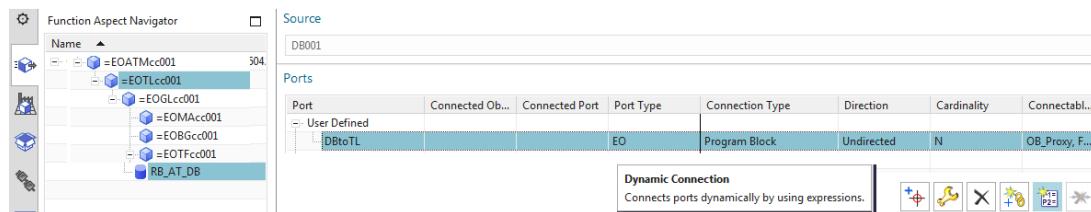


2. Create a port on RBAT IDB.



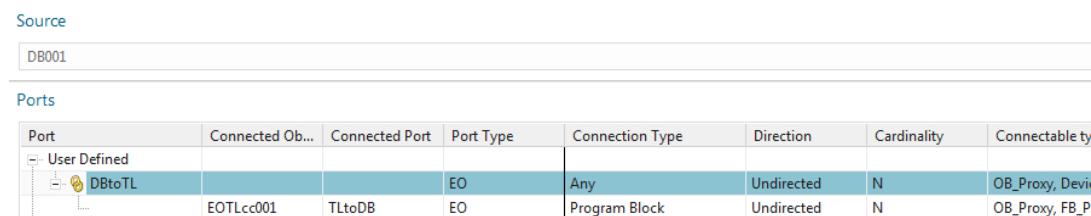
3. Create a dynamic connection by doing a `getPort` from RB\_AT IDB (p6 = conveyor Object name).

## 11.3 OB Main calls



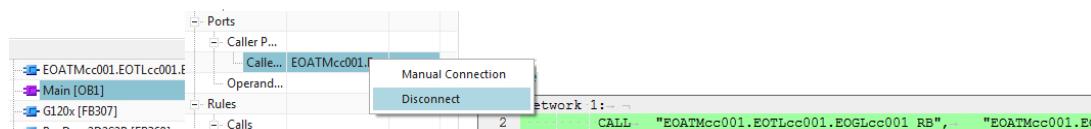
	Name	Formula	Value	Type
1		""	""	String
3	ccc	GetPort(ddd,"TLtoDB")	"EOTLcc001.Port2"	String
4	ddd	GetParent(p6,Function)	"EOTLcc001"	String
8	p6	🔒 (Attribute)	"EOGLcc001"	String

The following shows the result.

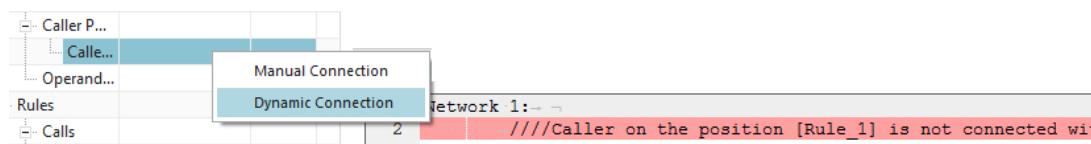


Now change the OB main to RBAT DB call port.

### 4. Disconnect the existing caller port.

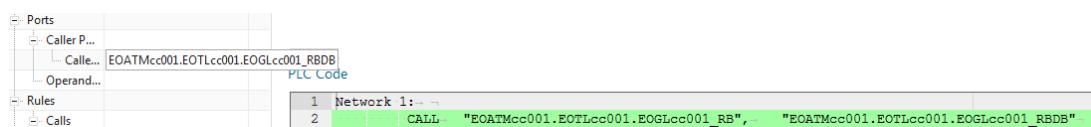


### 5. Add a dynamic connection (P1=TL Object Name).



	Name	Formula	Value	Type	Source
1		""	""	String	
2	bbb	GetConnectedObjects(p1,"TLtoDB")	{"DB001"}	List	
3	p1	🔒 (Attribute)	"EOTLcc001"	String	(EOTLcc001::Engineer...

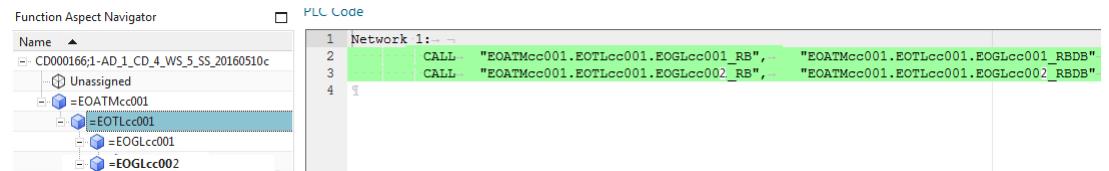
The following shows the resulting OB main call.



## 11.4 Generate

Now you need to check if everything has been setup correctly by simply copying a conveyor.

1. Right-click on GL.
2. Select **Copy**.
3. Right-click on TL.
4. Select **Paste**. The conveyor is added and OB main is updated.



The screenshot shows the Function Aspect Navigator on the left and the PLC Code editor on the right. In the Function Aspect Navigator, under the 'Unassigned' category, there is a node named '=EOTLcc001'. In the PLC Code editor, Network 1 contains the following code:

```
1 Network 1:-->
2     CALL "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
3     CALL "EOATMcc001.EOTLcc001.EOGLcc002_RB",    "EOATMcc001.EOTLcc001.EOGLcc002_RBDB"
4
```

**11.4 Generate**

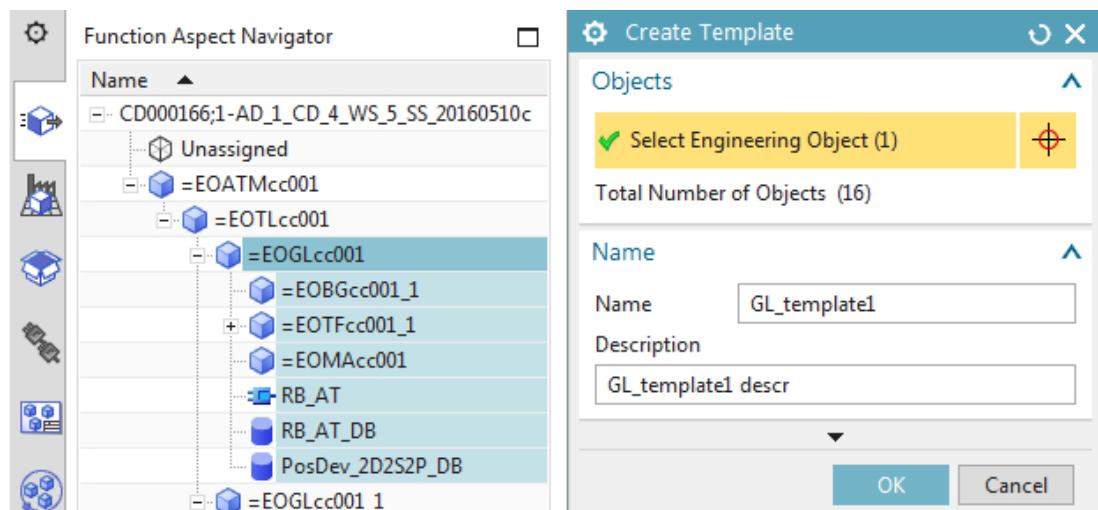
# Create/instantiate template

The goal of Automation Designer is to create templates that allow you to easily create software and EPLAN for items in the factory that are very repetitive. For example, a factory with 1000 conveyors that are more or less identical. Rather than having to think of unique global names for elements in the conveyors, you simply add an instance of a template to the aspect tree, and then modify the name of the top Engineering Object in the instance. Then you generate TIA Portal and EPLAN. This dramatically streamlines and simplifies your workflow.

This chapter shows how to create and instantiate a basic template.

## 12.1 Create template

1. Select GL01. Click **System Design→Create Template**.
2. For **Name** enter **GL\_Template**.



3. Click **OK**.



4. Click **OK**.
5. For **Classification** select **Software / block**.

---

## 12.1 Create template

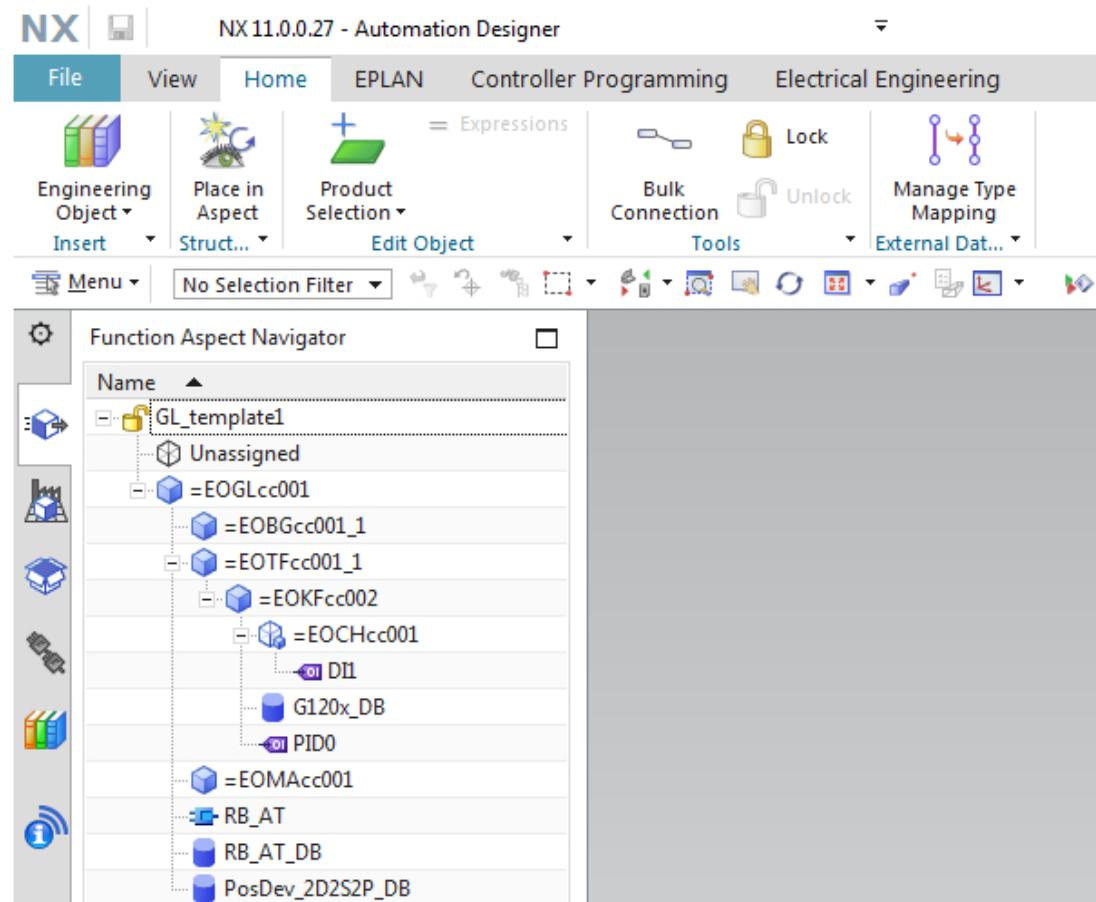
Classification Class

Name
+ Device
+ Devicefunction
+ EPLAN Macro
+ PLC
- Software
- Block
- DB
- FB
- FC
- OB

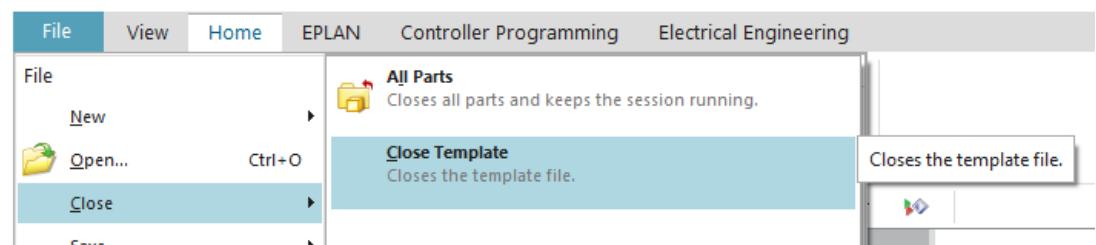
Properties

	Name	Value
1	AutoNumber	
2	Character Code	

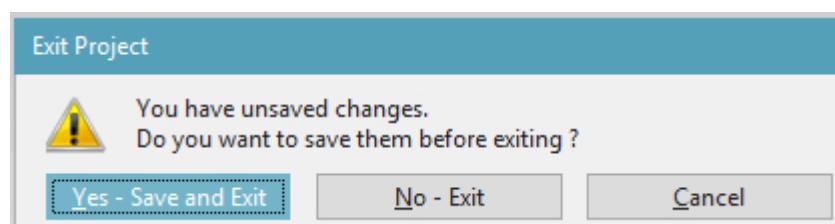
6. Click OK. You are now in the template editor.

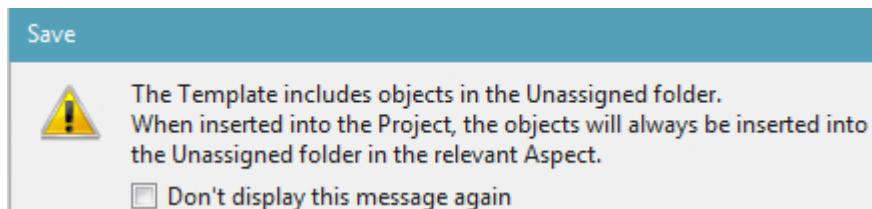


7. Choose File→Close→Close template.



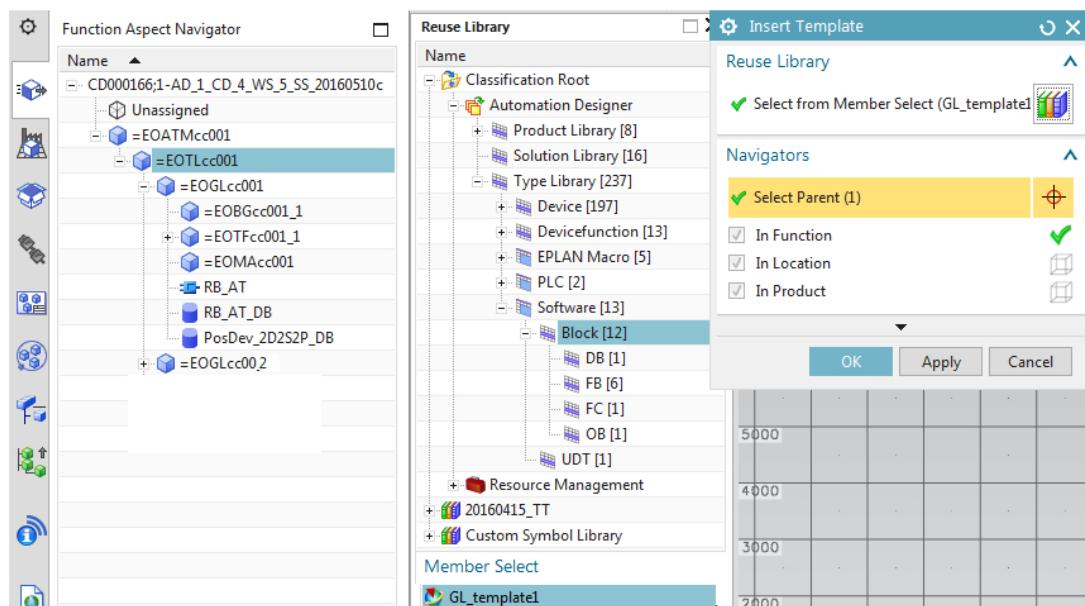
8. Choose Yes – Save and Exit.



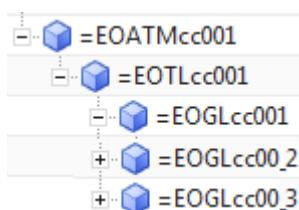


## 12.2 Insert template

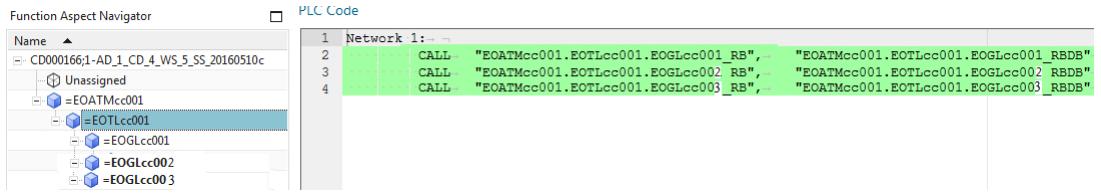
1. Undock the Reuse Library.
2. Drag & drop the template. The **Insert Template** dialog appears.
3. For **Parent** select **Engineering Object TL**.



4. Click **OK**. The template instance is added.
5. Change the name of the GL Engineering Object in the instance. This name must be unique, but the names of the Engineering Objects below GS will match those in the other conveyor.



The following shows the result for OB Main.



## 12.3 Synchronize changes

If no conveyors are added in Line Designer, then you would not need to instantiate any new GL templates in Automation Designer. If in Automation Designer you make changes (without deleting or adding conveyors), then you would not have to synchronize changes between Line Designer and Automation Designer. You would only have to send to TIA Portal and generate EPLAN reports.

If a conveyor is added in Line Designer, then you would need to add a new GL template instance in Automation Designer. This is a very simple process, and is a very simple example of the major improvements in efficiency possible with Automation Designer. The following describes the steps:

1. A new conveyor is added in Line Designer.
2. In Automation Designer perform **Map to new based on type**, which creates a new template instance. The template is mapped to the Line Designer conveyor.
3. Modify the **Symbolic Name** of the Engineering Object GL in the new instance.
3. Generate EPLAN.
4. Generate TIA Portal: (1) **Bulk Connect** the software, (2) run **Checkmate**, and (3) perform **Generate TIA Portal**.

