terry.taylor.ext@siemens.com
2016-07-29

Automation Designer Getting Started

Minimal (distilled) version

This doc at \\debonkl0c19\ADNX\Teams\Documentation\10_Meetings.

About this doc

This doc describes step-by-step basic Automation Designer with TIA Portal STEP 7 (TIA Portal) and EPLAN.

1. Assumes everything (Automation Designer, TIA, EPLAN) already configured (for setup details look at other docs/mp4’s in \10_Meetings).
2. focuses only on basics.
3. On my PC takes only 2-3 hours to complete.

TOC

2. Concepts

20160620 TERRY: maybe put concepts chapter first?

Text from user guide is in this red color and Cambria font.

Automation Designer is intended for electrical engineers and automation engineers from the discrete industry who work in teams on multi-disciplinary projects. By focusing on reusability, rule-based engineering, and cross-discipline collaboration and data integration, Automation Designer improves the consistency of such projects, reduces the time and cost invested in them, and increases the productivity of the entire production engineering process.

In Automation Designer, you design the electrical and automation solution for a production system or machine, eventually enabling the generation of its electrical schematics and PLC program. Automation Designer supports you in these tasks by facilitating engineering solution reuse and by automating the creation of the engineering data and documentation.
Automation Designer tightly integrates with the engineering data from the line and machine design. By using and enriching data from other engineering disciplines, Automation Designer reaches its full potential.

This section describes the following concepts:

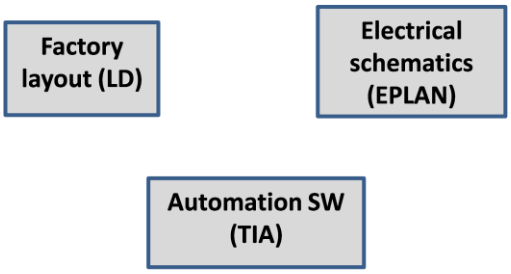
2.1. Traditional engineering versus integrated engineering
2.2. Reuse library
2.3. TeamCenter (4GD) details

2.1. Traditional engineering versus integrated engineering

1. Traditional engineering

Traditional engineering workflows include

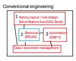
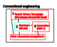
1. Factory layout and line design using Line Designer to design a production line.
2. Electrical engineering using EPLAN to generate schematics for the production line.
3. Automation using TIA Portal to generate PLC software and tags for specific PLC hardware.



Factory layout (LD)

Electrical schematics (EPLAN)

Automation SW (TIA)

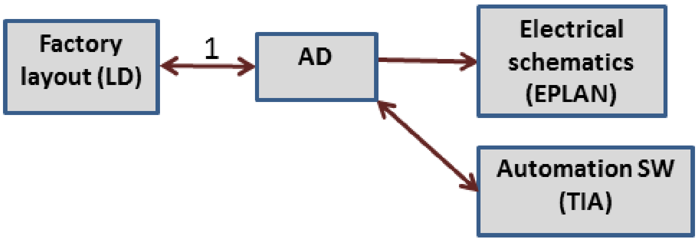
02c_21  

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

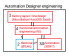
1. The Line Designer, EPLAN, and TIA Portal designers must manually synchronize their configurations.
2. TIA Portal software and tag names have no relationship to EPLAN schematic variables.
3. EPLAN and TIA Portal components that repeat (such as conveyors) must be created individually.

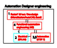
2. Integrated engineering with Automation Designer

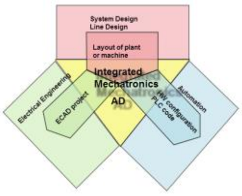
Automation Designer solves the problems above by linking to the above tools to provide centralized functional automation engineering.



02c_22







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).

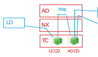
01_04Same source for macro variables and TIA Portal software3.

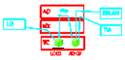
3. Use templates to quickly create TIA Portal software and EPLAN reports for common components (conveyors in this Getting Started). Instantiated templates are automatically assigned unique names as specified in naming rules.

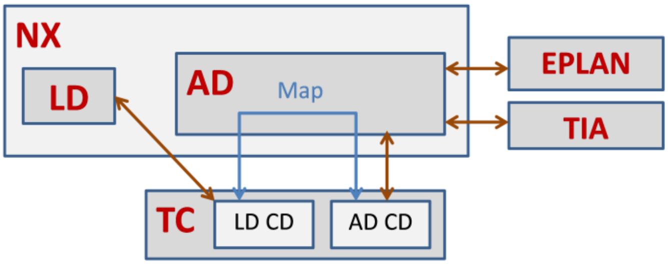
01_05 Instantiating multiple instances

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.

01_02AD as the central development tool







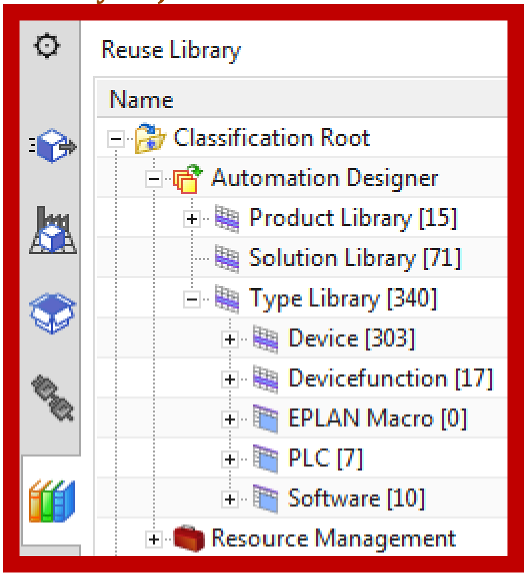
02c_23 (01_02bbb.png)

2.2. Reuse library

TERRY: this is all I found in the user guide about Reuse library

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

Library objects



Reuse Library

Name

Classification Root
Automation Designer
Product Library [15]
Solution Library [71]
Type Library [340]
Device [303]
Devicefunction [17]
EPLAN Macro [0]
PLC [7]
Software [10]
Resource Management

02c_01

The Reuse Library provides the following objects.

- Types

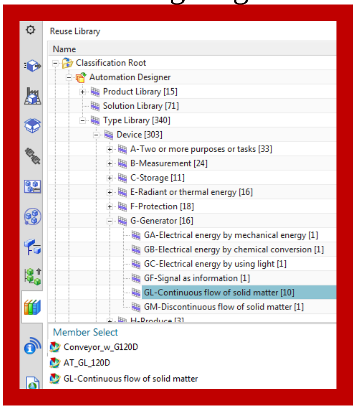
  Types are prototypes for Engineering Objects. They are made available in the Type Libary, where they are structured based on their classification classes. Objects with the same Type have the same characteristics.
  Availability in the Reuse Library: **Classification Root→Automation Designer→Type Library→Device** 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.

1. Engineering Objects

The following diagram shows Engineering Objects in the Reuse Library.



Reuse Library

Name

Classification Root
Automation Designer
Product Library [15]
Solution Library [71]
Type Library [340]
Device [303]
A-Two or more purposes or tasks [33]
B-Measurement [24]
C-Storage [11]
E-Radiant or thermal energy [16]
F-Protection [18]
G-Generator [16]
GA-Electrical energy by mechanical energy [1]
GB-Electrical energy by chemical conversion [1]
GC-Electrical energy by using light [1]
GF-Signal as information [1]
GL-Continuous flow of solid matter [10]
GM-Discontinuous flow of solid matter [1]
H-Produce [3]

Member Select
Conveyor_w_G120D
AT_GL_120D
GL-Continuous flow of solid matter

02c_02

Engineering Objects

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 so 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.

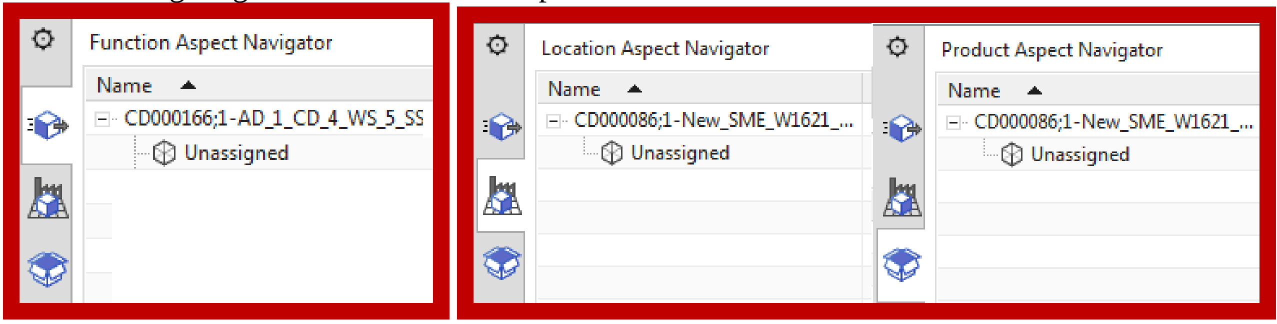
Project engineers can use each Type in their projects as many times as required, by inserting Engineering Objects with this Type. The Engineering Object will have the following data and defaults, as defined by its Type:

- 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.

2. Aspects

The following diagram shows the the 3 aspects.



Function Aspect Navigator

Name

CD000166;1-AD_1_CD_4_WS_5_SS

Unassigned

Location Aspect Navigator

Name

CD000086;1-New_SME_W1621_...

Unassigned

Product Aspect Navigator

Name

CD000086;1-New_SME_W1621_...

Unassigned

02c_03

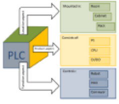
The IEC 81346 standard describes principles for structuring and naming objects and their associated information in industrial systems, installations, equipment, and industrial products. The goal of these principles is to handle the large sets of information that are available in these systems efficiently.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

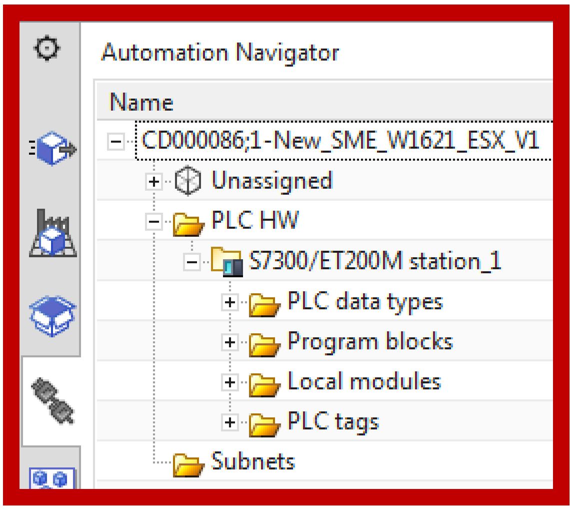
IEC 81346, part 1, calls these different views *aspects*. It defines the following aspects:

- 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).



3. AD automation tab



02c_04

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.

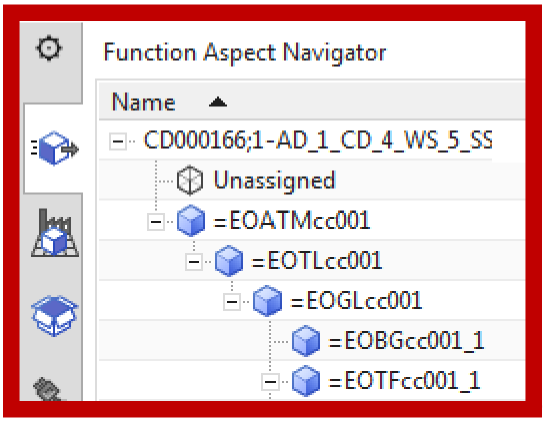
- If a program block is related to a specific Engineering Object, you can place it in one of the aspects that this Engineering Object has. It is then available in the corresponding Aspect Navigator.
- If a tag is related to a specific Engineering Object and you assign the tag to the Engineering Object,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

4. 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.



Function Aspect Navigator

Name ▲

```
⊟ CD000166;1-AD_1_CD_4_WS_5_SS
    Unassigned
  ⊟ =EOATMcc001
    ⊟ =EOTLcc001
      ⊟ =EOGLcc001
          =EOBGcc001_1
        ⊟ =EOTFcc001_1
```

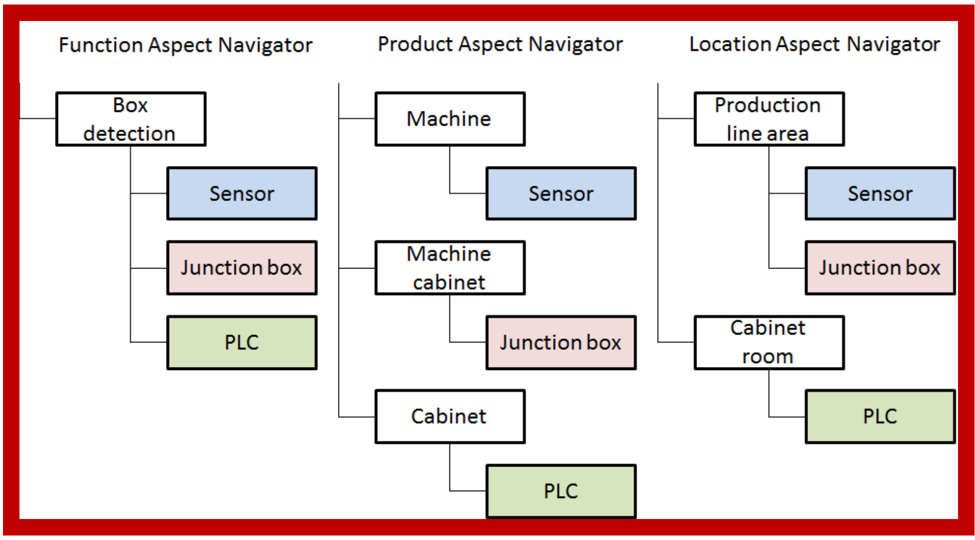
02c_05

One Engineering Object for independent aspect structures

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.

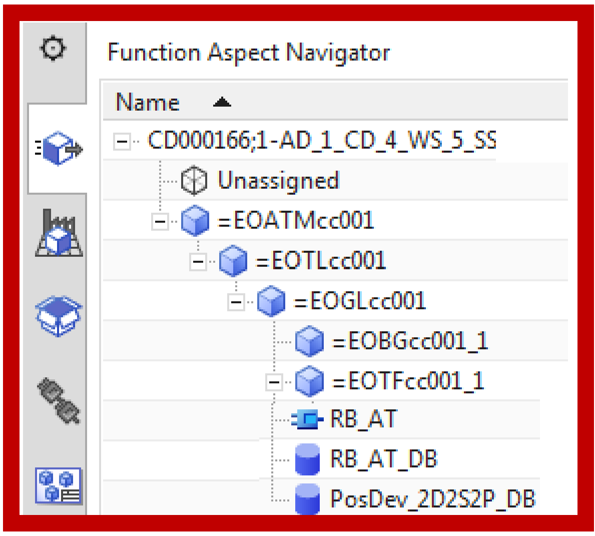


02c_06

5. TIA Portal software in aspects

Software placed in the aspects under an Engineering Objecrt. This aspect chain above this Engineering Object is use to determine the the symbolic name (unique identifier) of the Engineering Objecrt. This name is then used to

- 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.



Function Aspect Navigator

Name ▲

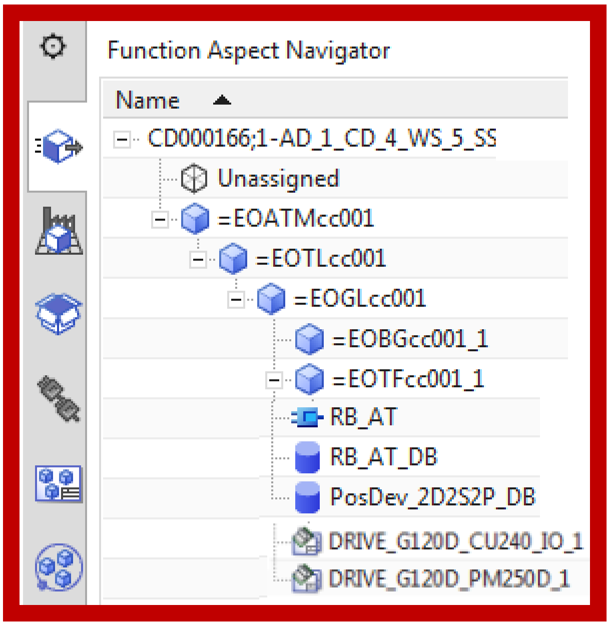
- CD000166;1-AD_1_CD_4_WS_5_SS
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
          - RB_AT
          - RB_AT_DB
          - PosDev_2D2S2P_DB

02c_07

6. macros in aspects

Software and macros are placed in the aspects under an Engineering Objecrt. This aspect chain above this Engineering Object is use to determine the the symbolic name (unique identifier) of the Engineering Objecrt. This name is then used to

- In EPLAN reports (using symbolic names for variables).



Function Aspect Navigator

Name ▲

- CD000166;1-AD_1_CD_4_WS_5_SS
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
          - RB_AT
          - RB_AT_DB
          - PosDev_2D2S2P_DB
        - DRIVE_G120D_CU240_IO_1
        - DRIVE_G120D_PM250D_1

02c_08

7. 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.

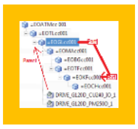
Objects have vertical relations and horizontal relations to other objects.
• *Vertical relations* define parent-child relations in the Aspect Navigators.
• *Horizontal relations* are connections between ports. They do not define parent-child relations. They can connect objects from different navigators or in the same navigator.

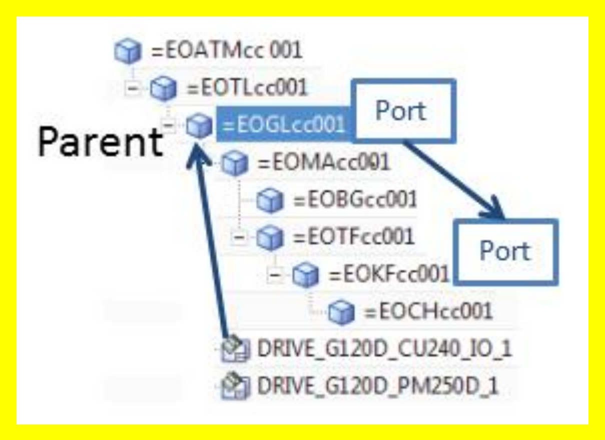
Ports are a means of connecting objects. They are available for Engineering Objects and tags. You connect the port of one object, the source port, to the port of another object, the target port. This creates a bidirectional connection. *Connections* represent port-based relations between objects.

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 conected 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.





02c_09

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.

8. 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.

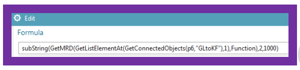
~~**Manual and dynamic connections**~~

~~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.

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

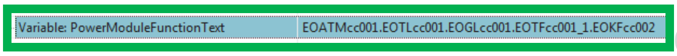


Formula

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

09b_11

The resulting property value:



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

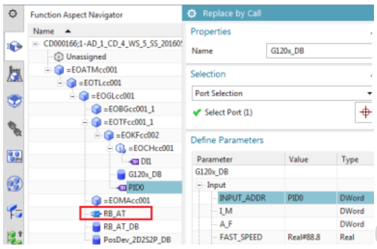
09b_13

xxxx2.3.4. Software dynamization

Imported software blocks sometimes require dynamization. Dynamization involves redefining the calls and call parameters and references to tags.

The following example is from this Getting Started, and shows how a call in RB_AT is dynamized:

1. A call is replaced with a “Replace by call” to G120x IDB.
2. Call parameters INPUT_ADDR and FAST_SPEED are redefined.

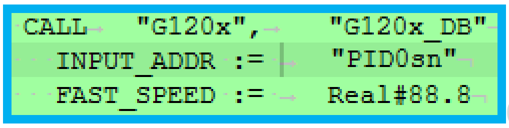


Name: G120x_DB

Parameter	Value	Type
G120x_DB		
Input		
INPUT_ADDR	PID0	DWord
I_M		DWord
A_F		DWord
FAST_SPEED	Real#88.8	Real

08b_43

The following is the result.



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

08b_45

9. 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*.

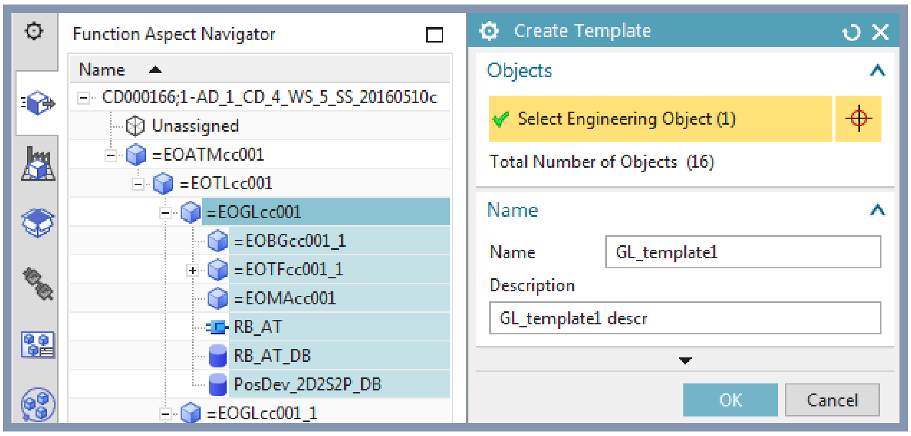
A *template definition* is the blueprint for an engineering solution that project engineers want to reuse in their projects. Template engineers create template definitions in the template environment, where they build the content of the template. This process is called template creation.
The template 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. Creating and instantiating
2. Dynamic connections
3. Workflows

1. Creating and instantiating

The following diagram shows the Function aspect with 2 existing GL conveyor Engineering Objects. The second Engineering Object is simply a copy of the first. A template is about to be created from the first GL Engineering Object.



Function Aspect Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_SS_20160510c
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
        - =EOMAcc001
        - RB_AT
        - RB_AT_DB
        - PosDev_2D2S2P_DB
      - =EOGLcc001_1

Create Template

Objects

✓ Select Engineering Object (1)

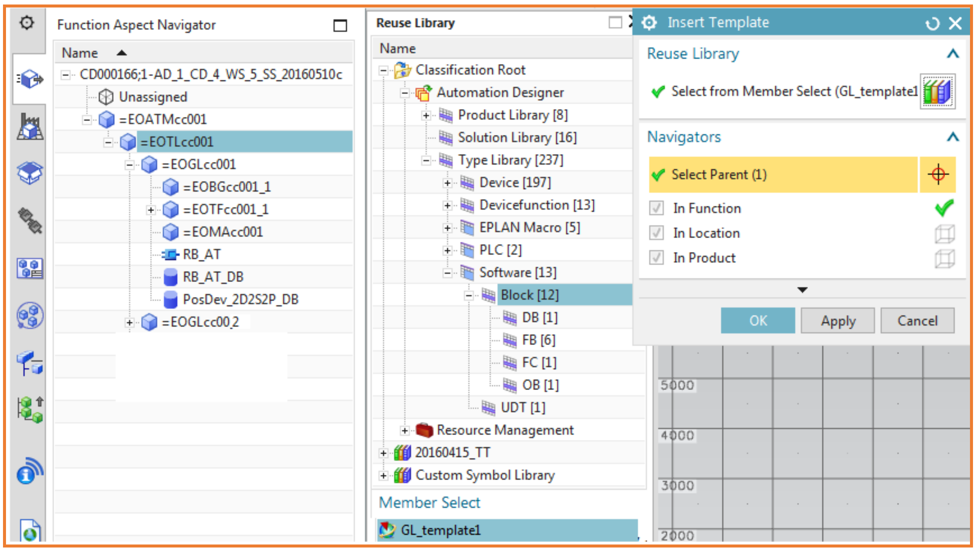
Total Number of Objects (16)

Name

Name: GL_template1

Description: GL_template1 descr

In the following diagram an instance of this template is about to be added to the Function aspect.



Function Aspect Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_SS_20160510c
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
        - =EOMAcc001
        - RB_AT
        - RB_AT_DB
        - PosDev_2D2S2P_DB
      - =EOGLcc002

Reuse Library

Name

- Classification Root
  - Automation Designer
    - Product Library [8]
    - Solution Library [16]
    - Type Library [237]
      - Device [197]
      - Devicefunction [13]
      - EPLAN Macro [5]
      - PLC [2]
      - Software [13]
        - Block [12]
          - DB [1]
          - FB [6]
          - FC [1]
          - OB [1]
        - UDT [1]
  - Resource Management
- 20160415_TT
- Custom Symbol Library

Member Select

GL_template1

Insert Template

Reuse Library

✓ Select from Member Select (GL_template1

Navigators

✓ Select Parent (1)

☑ In Function ✓

☑ In Location

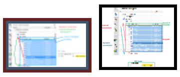
☑ In Product

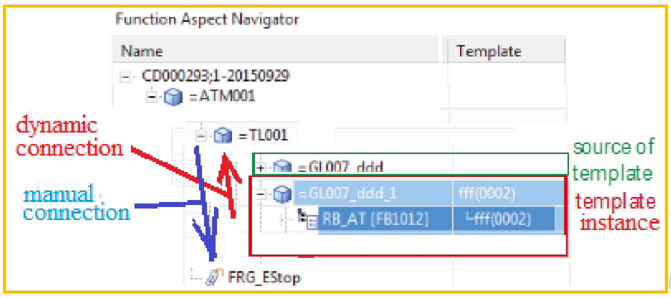
2. Dynamic connections

An import aspect of templates is the dynamic connection.

1. Access by software inside the template of tags outside of the template.

The following diagram shows how with a dynamic expression (red arrow to TL01) the template instance automatically locates the required parent Engineering Object TL. TL has a manual link (blue) to the tag FRG_EStop. Thus the instantiated template can link to the tag (as long as the template instance has TL as parent Engineering Object).



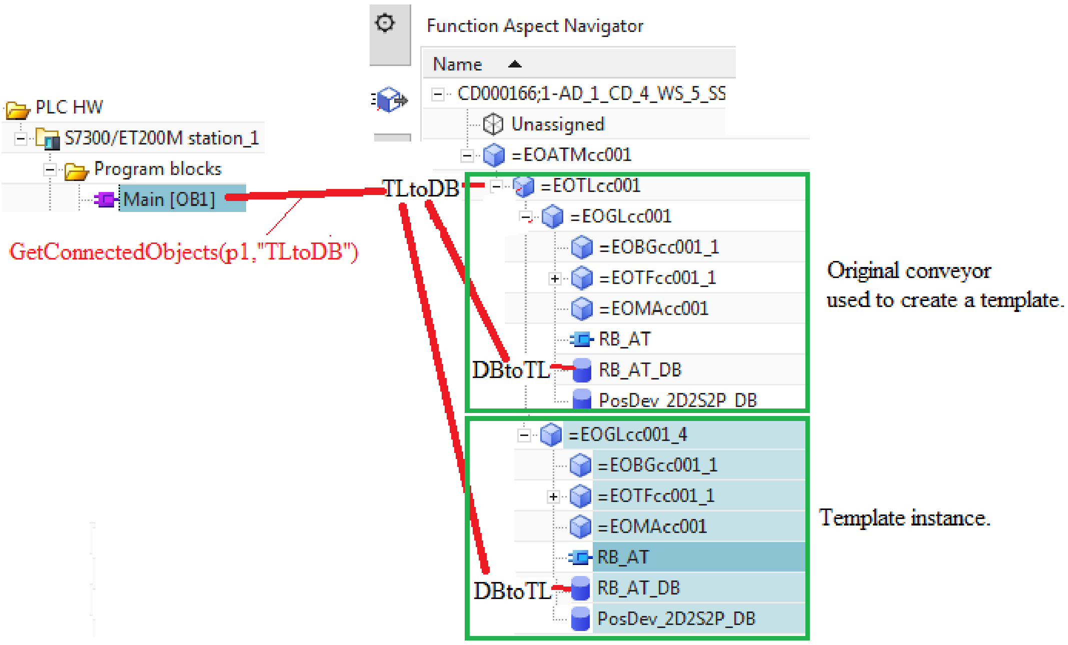


09b_16 xxxxxxxx

2. Automated insertion of calls in software blocks outside an added template instance to software blocks inside the instance.

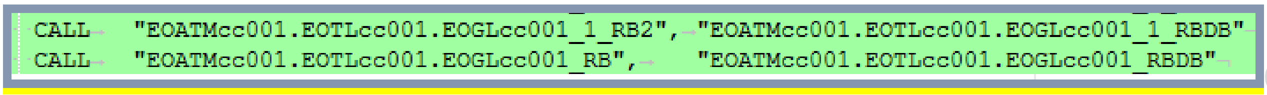
For example, in this Getting Started you have the following example. You have one GL conveyor and instantiate a second from a GL template. The OB main call auto updates, adding a call to the new conveyor.

The following shows how this is configured. “TLtoDB” and “DBtoTL” are ports, and “GetConnectedObjects(p1,”TltoDB”)” is the expression that for Main OB1 and automatically adds a call to new instances.



09b_27

This is the result.



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

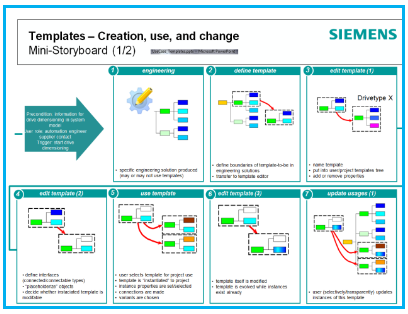
09b_28

3. Workflows

The following diagrams explain the workflows with templates.

See ReqSpec_AD@NX_Templates.docx

20160311 \\debonkl0c19\ADNX\Teams\PRM\Topics\Templates\Material\UseCase_Templates.pptx



Templates – Creation, use, and change
Mini-Storyboard (1/2)

SIEMENS

Precondition: information for drive dimensioning in system model
User role: automation engineer supplier contact
Trigger: start drive dimensioning

1 engineering
- specific engineering solution produced (may or may not use templates)

2 define template
- define boundaries of template-to-be in engineering solutions
- transfer to template editor

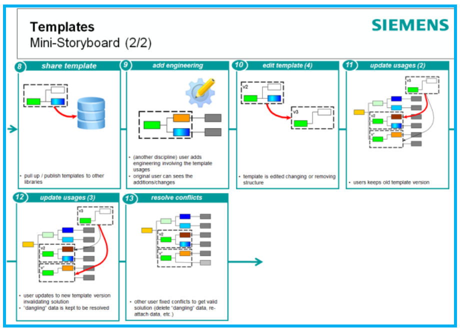
3 edit template (1)
- name template
- put into user/project templates tree
- add or remove properties

4 edit template (2)
- define interfaces (connected/connectable types)
- "placeholderize" objects
- decide whether instaciated template is modifable

5 use template
- user selects template for project use
- template is "instantiated" to project
- instance properties are set/selected
- connections are made
- variants are chosen

6 edit template (3)
- template itself is modified
- template is evolved while instances exist already

7 update usages (1)
- user (selectively/transparently) updates instances of this template



Templates
Mini-Storyboard (2/2)

SIEMENS

8 share template
- pull up / publish templates to other libraries

9 add engineering
- (another discipline) user adds engineering involving the template usages
- original user can sees the additions/changes

10 edit template (4)
- template is edited changing or removing structure

11 update usages (2)
- users keeps old template version

12 update usages (3)
- user updates to new template version invalidating solution
- "dangling" data is kept to be resolved

13 resolve conflicts
- other user fixed conflicts to get valid solution (delete "dangling" data, re-attach data, etc.)

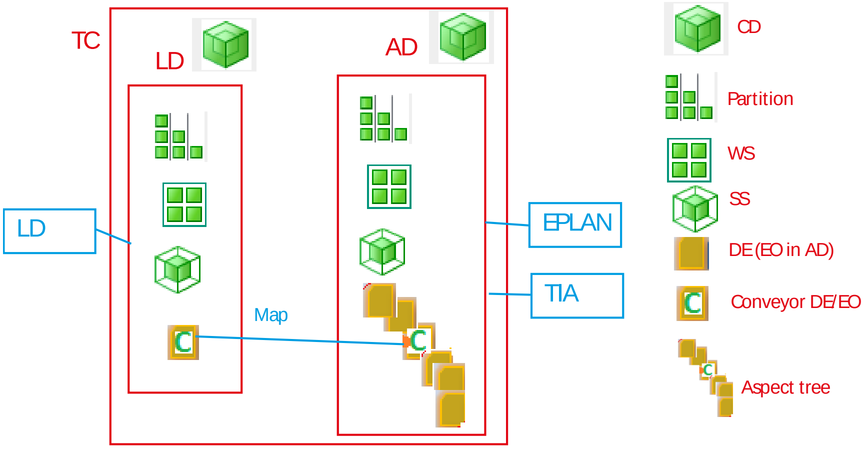
2.3. TeamCenter (4GD) details

~~In part 1 (chapters 3-4) you create Line Designer/Automation Designer 4GD (4th Generation Design) components.~~

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

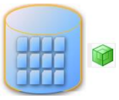
1. Collaborative Design
2. Partition scheme
3. Partition
4. Workset
5. Subset
6. Design element

01_12 4GD components



1. Collaborative Design

01_13 Collaborative Design



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 Collaborative Design is the container object in TeamCenter of all the design data that defines a product or a class of products.

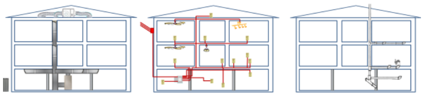
2. Partition scheme

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.

01_14 Partition scheme

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

01_15 Functional partition scheme



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.

01_16 Spatial partition scheme

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

01_17 Physical partition scheme

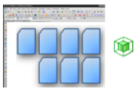
3. Partition

01_18 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 Design Element. Design Elements can be place 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.

4. Workset

01_19 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.

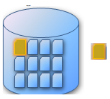
5. Subset

01_20 Subset

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.

6. Design element

01_21 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 Design Elements. A Design Element can reference an NX part or assembly model, or other types of geometry.

xxxx2. 4GD example workflow

An example application would be a Collaborative Design of a house in which multiple designers design the cabinets, appliances, plumbing, and wiring in the kitchen. This is the workflow:

1. Collaborative Design. An administrator creates the Collaborative Design in TeamCenter.

2. Partition scheme and partition. An admin creates the partition scheme and partition objects in TeamCenter. The partition schemes might include a functional partition scheme for systems, and a physical partition scheme for rooms.The functional PS includes partition objects for plumbing, heating and wiring.The physical PS includes partition objects for the different rooms of the house.

3. Workset and subset. The project leader creates a new workset in NX, adds a subset with a recipe for selecting all Design Elements in the kitchen, and assigns it to the responsible designers.

1. Workflow overview

This overview discusses the workflows in this Getting Started,

1.0. Prerequisites
1.1. Workflow from the tools perspective
1.2. Workflow

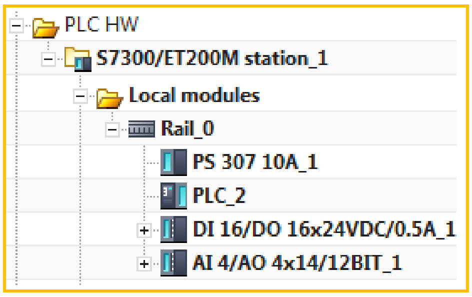
1.0. Prerequisites

This Getting Started assumes you have the following already configured:

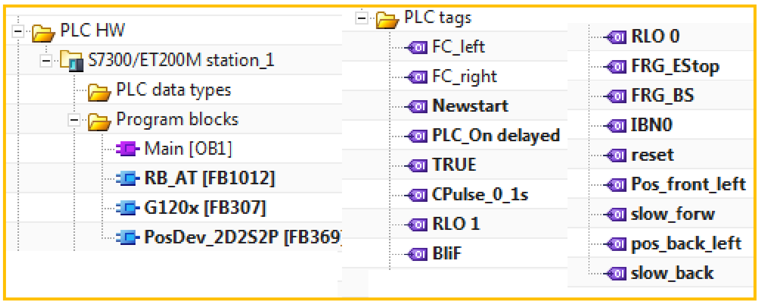
1. NX, TeamCenter and Line Designer with
   - Conveyor (any part is OK).
   - The following in the reuse library Classification Root
     - Device / A ->1 purpose or task / AT
     - Device / U-Keep
     - Device / G-Generator / GL-Continuous flow
     - Device / M-Motor / MA-Electromagnetic
     - Device / 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:



- PLC HW
  - S7300/ET200M station_1
    - Local modules
      - Rail_0
        - PS 307 10A_1
        - PLC_2
        - DI 16/DO 16x24VDC/0.5A_1
        - AI 4/AO 4x14/12BIT_1



- PLC HW
  - S7300/ET200M station_1
    - PLC data types
    - Program blocks
      - Main [OB1]
      - RB_AT [FB1012]
      - G120x [FB307]
      - PosDev_2D2S2P [FB369

- PLC tags
  - FC_left
  - FC_right
  - Newstart
  - PLC_On delayed
  - TRUE
  - CPulse_0_1s
  - RLO 1
  - BliF
  - RLO 0
  - FRG_EStop
  - FRG_BS
  - IBN0
  - reset
  - Pos_front_left
  - slow_forw
  - pos_back_left
  - slow_back

01c_21,22

1.1. Workflow from the tools perspective

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

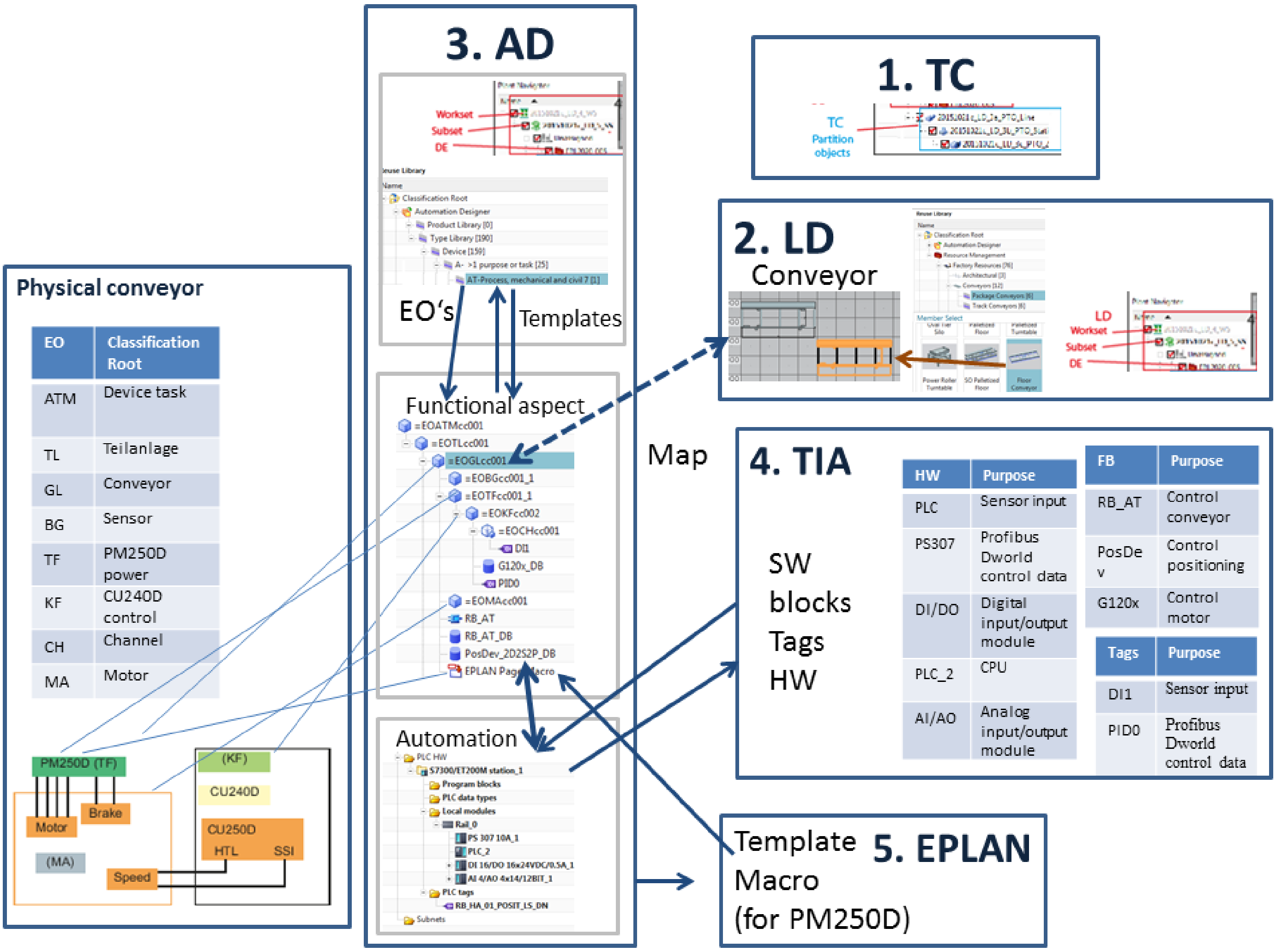
TeamCenter: Create a Line Designer 4GD Collaborative Design.

Line Designer: Create a 4GD workset, subset, and add two conveyor Design Elements.

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.

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.

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.



Physical conveyor

EO	Classification Root
ATM	Device task
TL	Teilanlage
GL	Conveyor
BG	Sensor
TF	PM250D power
KF	CU240D control
CH	Channel
MA	Motor

4. TIA

HW	Purpose
PLC	Sensor input
PS307	Profibus Dworld control data
DI/DO	Digital input/output module
PLC_2	CPU
AI/AO	Analog input/output module

FB	Purpose
RB_AT	Control conveyor
PosDe v	Control positioning
G120x	Control motor

Tags	Purpose
DI1	Sensor input
PID0	Profibus Dworld control data

1.2. Workflow

The following describes what you do in this Getting Started.

~~The next chapter introduces Automation Designer concepts.~~

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

1 Automation Designer is based on the NX framework and uses Teamcenter as the data backbone. This allows seamless data exchange from Line Designer, another Nxbased and Teamcenter-based solution.
2 Line Designer objects are an integrated part of the Automation Designer system design. The automation engineer sees the line design as the mechanical engineer sees it.
3 Objects from the Line Designer layout are used in Automation Designer during system design and further enriched during electrical and automation engineering.

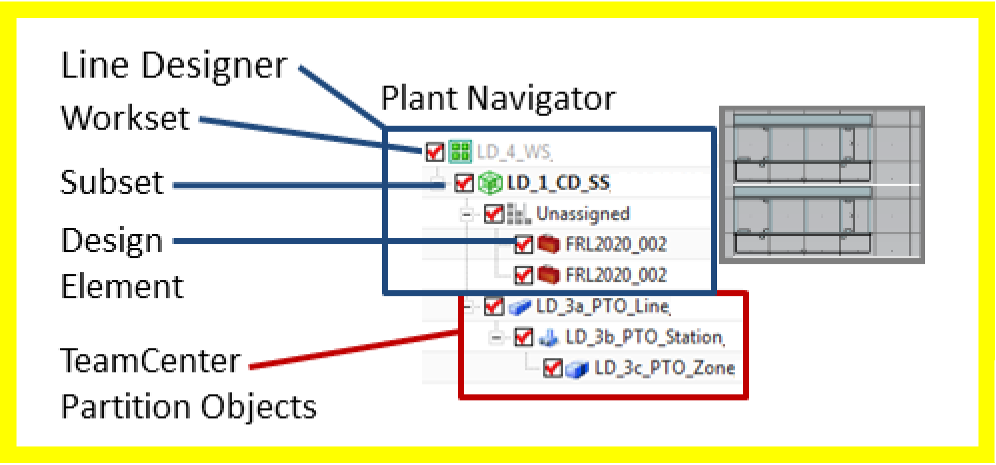
Create Line Designer Collaborative Design, create Line Designer workset, subset and Design Elements.

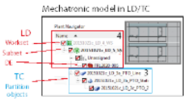
A project is the container that stores the objects you need to carry out the electrical and automation engineering for a production system or machine. In Automation Designer this container is called *project*, in Teamcenter it is called *collaborative design object*. For every Automation Designer project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





02_01c

Create Automation Designer workset, subset and Engineering Objects.

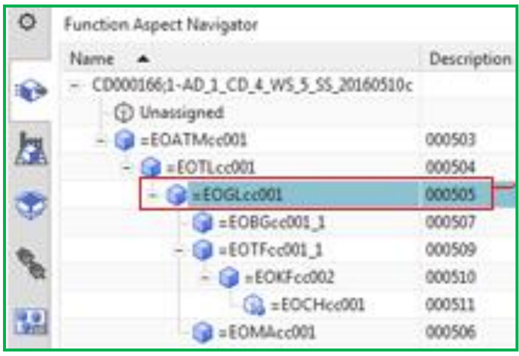
To work in a project, you need a workset. The *workset* is your work context and gives you access to the data that was saved in the workset's project. In a workset, you do the following:

- 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.



Function Aspect Navigator

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

01_02c

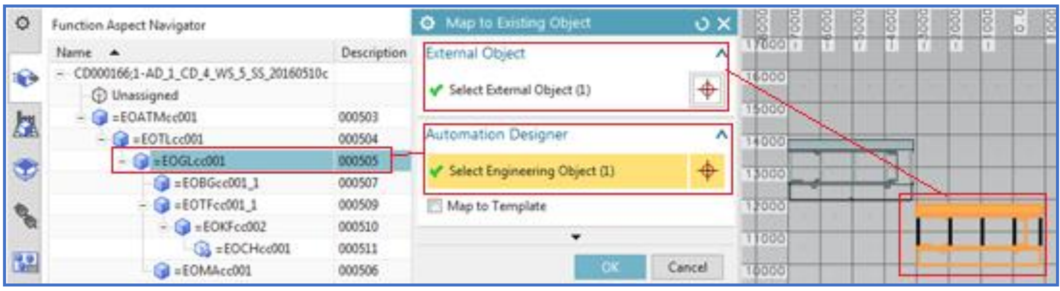
Part 2: Mapping Line Designer-Automation Designer, generating EPLAN and TIA Portal (without templates)

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.



Function Aspect Navigator

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

Map to Existing Object

External Object

✓ Select External Object (1)

Automation Designer

✓ Select Engineering Object (1)

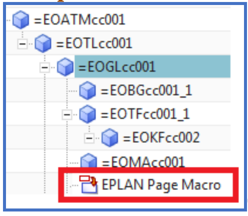
□ Map to Template

01c_03

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.

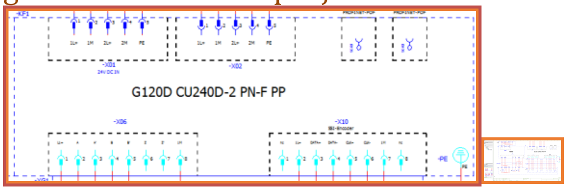
(1) Import EPLAN template and macro. You can import EPLAN project templates and macros into Automation Designer. Import macros into a project and use them directly or import them into a template and reuse the template.



- =EOATMcc001
  - =EOTLcc001
    - =EOGLcc001
      - =EOBGcc001_1
      - =EOTFcc001_1
        - =EOKFcc002
      - =EOMAcc001
      - EPLAN Page Macro

01c_04

(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.



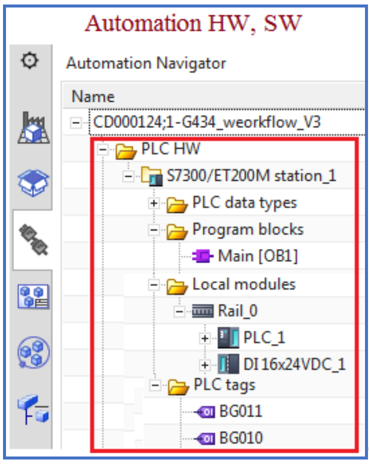
01c_05

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 created IDB's and tags.



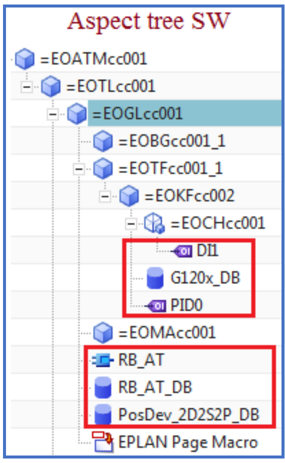
Automation HW, SW

Automation Navigator

Name

- CD000124;1-G434_weorkflow_V3
  - PLC HW
    - S7300/ET200M station_1
      - PLC data types
      - Program blocks
        - Main [OB1]
      - Local modules
        - Rail_0
          - PLC_1
          - DI 16x24VDC_1
      - PLC tags
        - BG011
        - BG010

01c_06



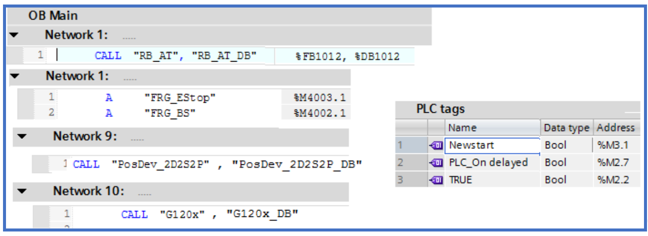
Aspect tree SW

- =EOATMcc001
  - =EOTLcc001
    - =EOGLcc001
      - =EOBGcc001_1
      - =EOTFcc001_1
        - =EOKFcc002
          - =EOCHcc001
            - DI1
            - G120x_DB
            - PID0
      - =EOMAcc001
      - RB_AT
      - RB_AT_DB
      - PosDev_2D2S2P_DB
      - EPLAN Page Macro

01c_07

(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.



```
OB Main
Network 1:
    CALL  "RB_AT", "RB_AT_DB"        %FB1012, %DB1012
Network 1:
    A     "FRG_EStop"                %M4003.1
    A     "FRG_BS"                   %M4002.1
Network 9:
    CALL  "PosDev_2D2S2P" , "PosDev_2D2S2P_DB"
Network 10:
    CALL  "G120x" , "G120x_DB"
```

PLC tags

	Name	Data type	Address
1	Newstart	Bool	%M3.1
2	PLC_On delayed	Bool	%M2.7
3	TRUE	Bool	%M2.2

01c_08

Part 3: Generating EPLAN and TIA Portal with templates

Template-related concepts.

You can insert objects individually or by 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:

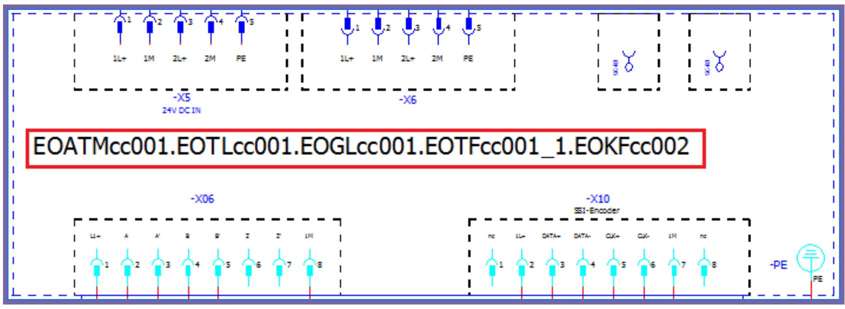
- The conveyor, a motor, a frequency converter, sensors, and signal converters
- EPLAN macros for preparing the generation of the electrical schematics for this conveyor
- Program blocks whose code controls the conveyor, motor, frequency converter, and sensors

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.

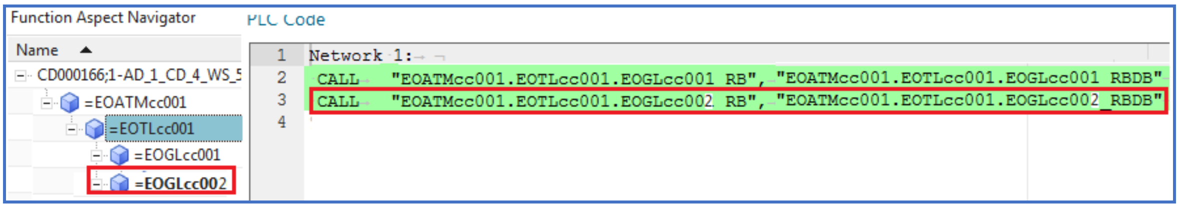


EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002

01c_09

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.

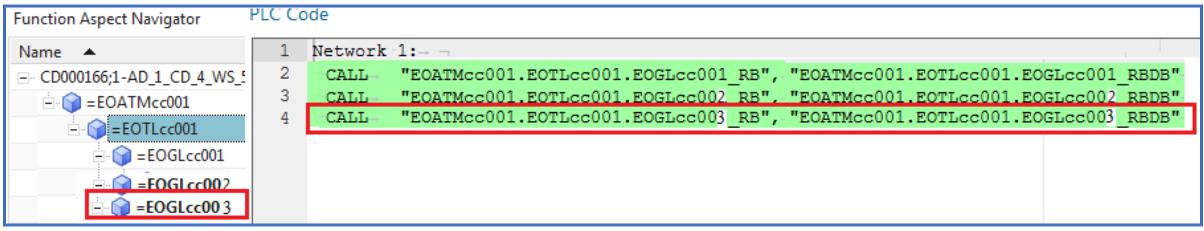


```
Network 1:
  CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB", "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
  CALL   "EOATMcc001.EOTLcc001.EOGLcc002_RB", "EOATMcc001.EOTLcc001.EOGLcc002_RBDB"
```

01c_10

Create/instantiate template.

The following shows an instantiated template.



```
Network 1:
  CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB", "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
  CALL   "EOATMcc001.EOTLcc001.EOGLcc002_RB", "EOATMcc001.EOTLcc001.EOGLcc002_RBDB"
  CALL   "EOATMcc001.EOTLcc001.EOGLcc003_RB", "EOATMcc001.EOTLcc001.EOGLcc003_RBDB"
```

01c_11

Part 1. Create Line Designer/Automation Designer TeamCenter components

This part shows how to create the TeamCenter components for Line Designer and Automation Designer.

3. TeamCenter: Create Line Designer Collaborative Design

TeamCenter is used as the database for your Line Designer project. So you have to create a TeamCenter Collaborative Design. If you make a mistake in the following steps, then start over. TeamCenter is confusing and error-prone.

3.1. Create plant design Collaborative Design

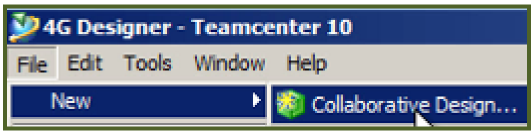
3.2. Create partition scheme

3.3. Create partition objects

3.1. Create plant design Collaborative Design

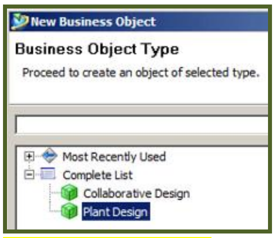
A project is the container that stores the objects you need to carry out the electrical and automation engineering for a production system or machine. In Automation Designer this container is called *project*, in Teamcenter it is called *collaborative design object*. For every Automation Designer project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

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



03b_01

2. Select **Plant Design**.



New Business Object

Business Object Type

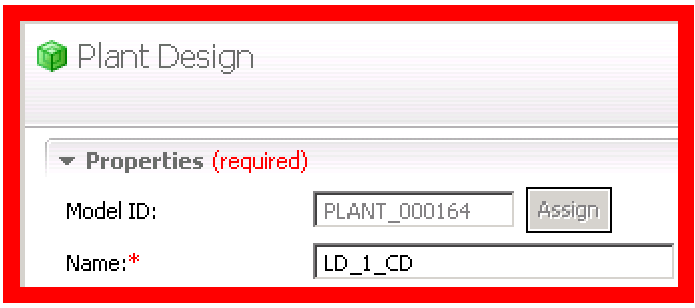
Proceed to create an object of selected type.

03b_02

3. Click **Assign**.

4. For **Name** enter "LD_1_CD " (Line Designer Collaborative Design).

these names make it easier to follow in TeamCenter what is being created. Normally I add the date and/or initials to the name, such as LD_1_CD_TT_20160509



Plant Design

Properties (required)

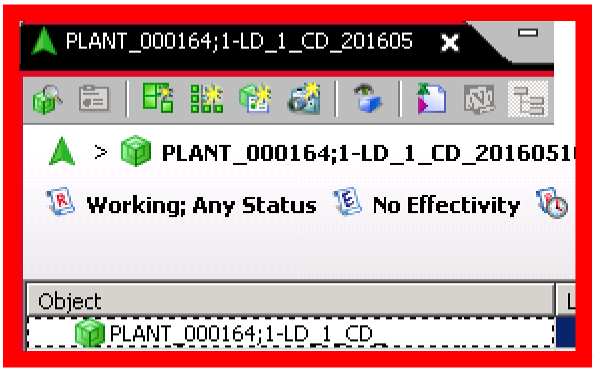
Model ID: PLANT_000164

Name:* LD_1_CD

03b_03

5. Click **Finish**.

6. Click **Close**.

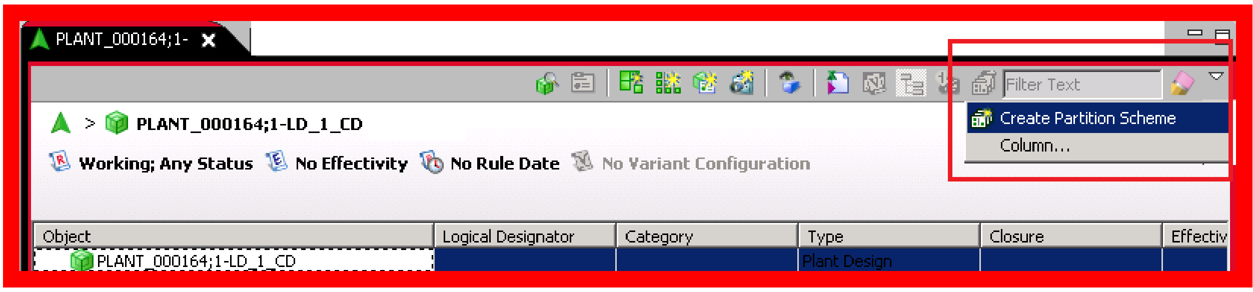


03b_04

3.2. Create partition scheme

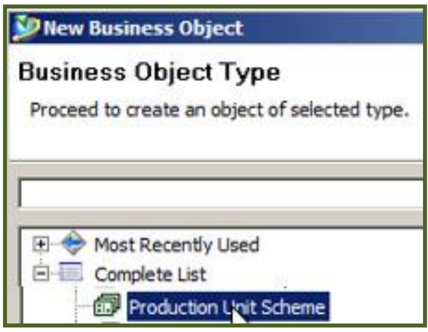
Partition schemes can be functional, spatial, or physical. Partitions are created within partition schemes . For this Getting Started you create a single partition scheme.

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



03b_05

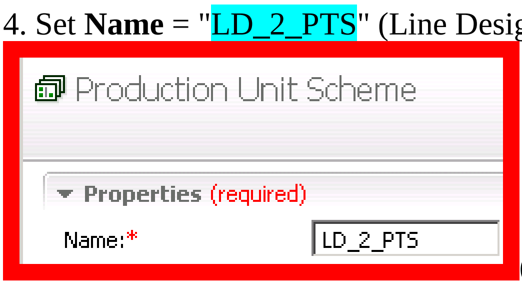
2. Select **Production Unit Scheme**.



03b_06

3. Click **Next**.

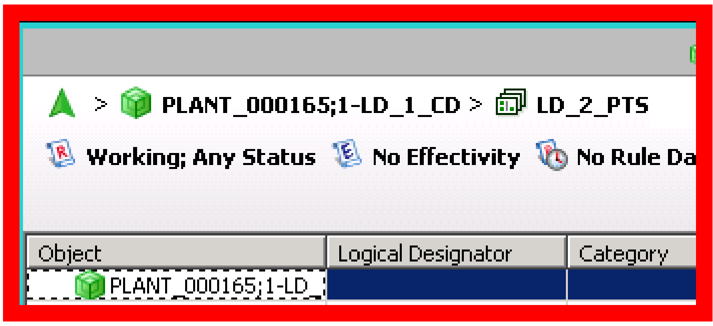
4. Set **Name** = "LD_2_PTS" (Line Designer Partition Scheme).



03b_07

5. Click **Finish**.

6. Click **Close**.



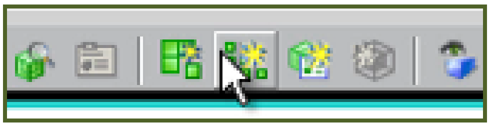
03b_08

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 place 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.

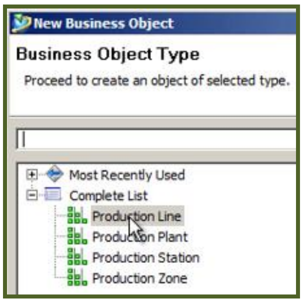
Create the partition objects (business objects) line, station and zone.

1. Click on **Create partition**.



03b_09

2. Select **Production Line**.
3. Click **Next**.



New Business Object

Business Object Type

Proceed to create an object of selected type.

Most Recently Used

Complete List

Production Line

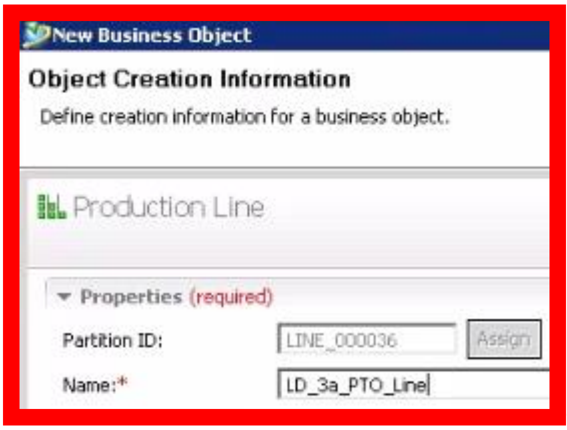
Production Plant

Production Station

Production Zone

03b_10

4. Click **Assign**.
5. Set **Name** = "LD_3a_PTO_Line".



New Business Object

Object Creation Information

Define creation information for a business object.

Production Line

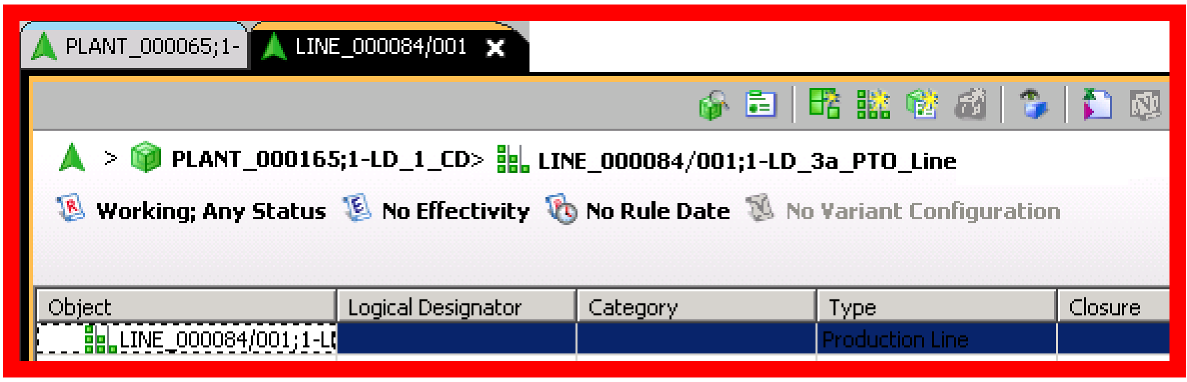
Properties (required)

Partition ID: LINE_000036 Assign

Name:* LD_3a_PTO_Line

03b_11

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



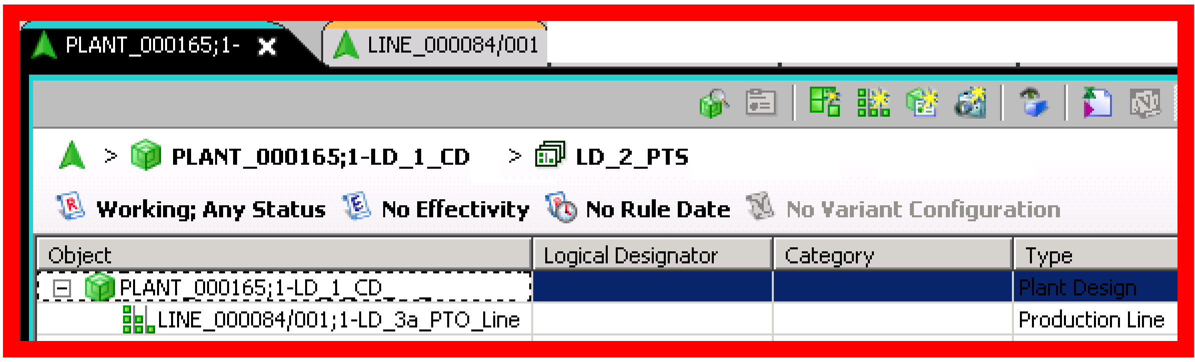
PLANT_000065;1- LINE_000084/001

> PLANT_000165;1-LD_1_CD> LINE_000084/001;1-LD_3a_PTO_Line

Working; Any Status No Effectivity No Rule Date No Variant Configuration

Object	Logical Designator	Category	Type	Closure
LINE_000084/001;1-LI			Production Line	

03b_12



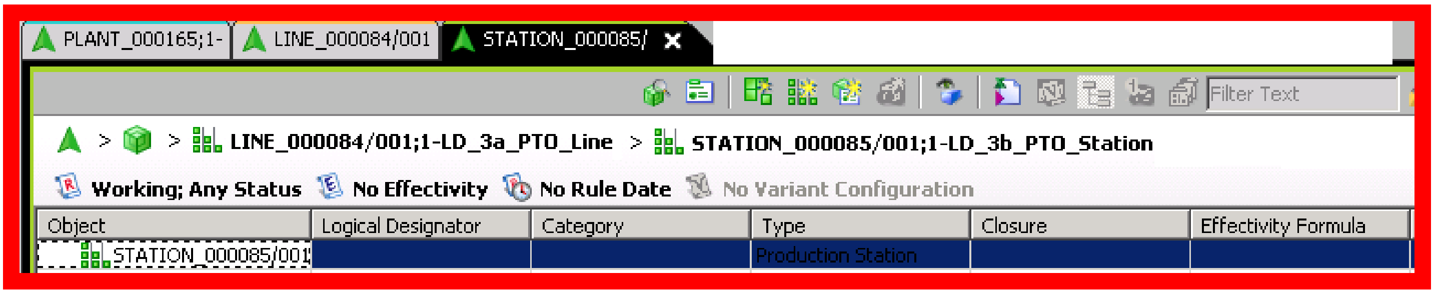
> PLANT_000165;1-LD_1_CD > LD_2_PTS

Working; Any Status No Effectivity No Rule Date No Variant Configuration

Object	Logical Designator	Category	Type
PLANT_000165;1-LD_1_CD			Plant Design
LINE_000084/001;1-LD_3a_PTO_Line			Production Line

03b_13

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.

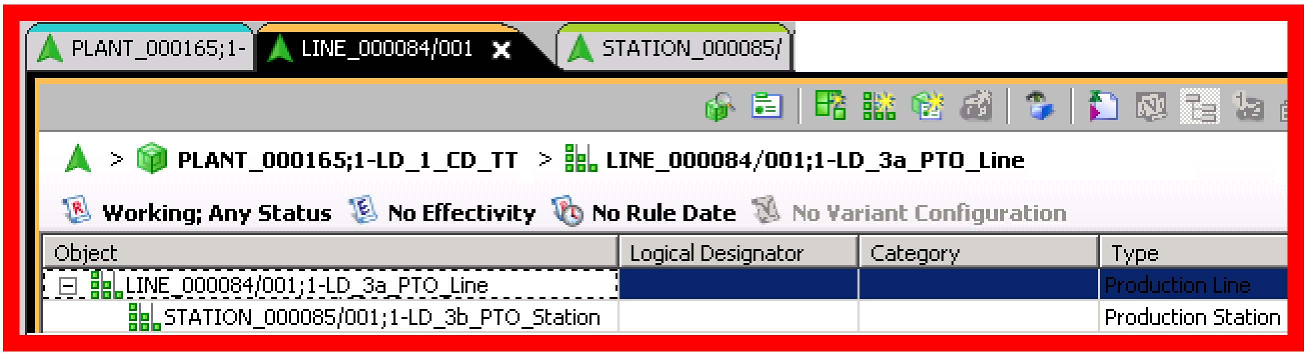


> > LINE_000084/001;1-LD_3a_PTO_Line > STATION_000085/001;1-LD_3b_PTO_Station

Working; Any Status No Effectivity No Rule Date No Variant Configuration

Object	Logical Designator	Category	Type	Closure	Effectivity Formula
STATION_000085/001			Production Station		

03b_14

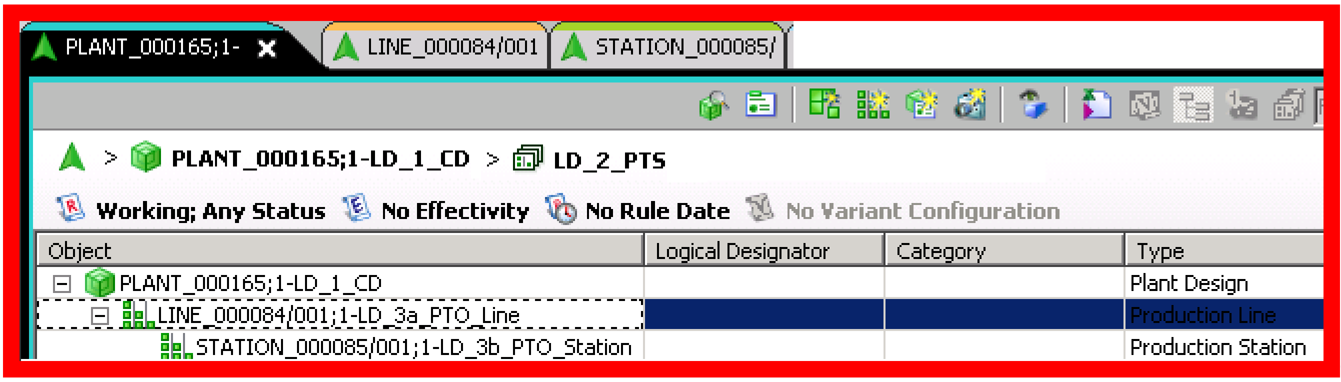


> PLANT_000165;1-LD_1_CD_TT > LINE_000084/001;1-LD_3a_PTO_Line

Working; Any Status No Effectivity No Rule Date No Variant Configuration

Object	Logical Designator	Category	Type
LINE_000084/001;1-LD_3a_PTO_Line			Production Line
STATION_000085/001;1-LD_3b_PTO_Station			Production Station

03b_15



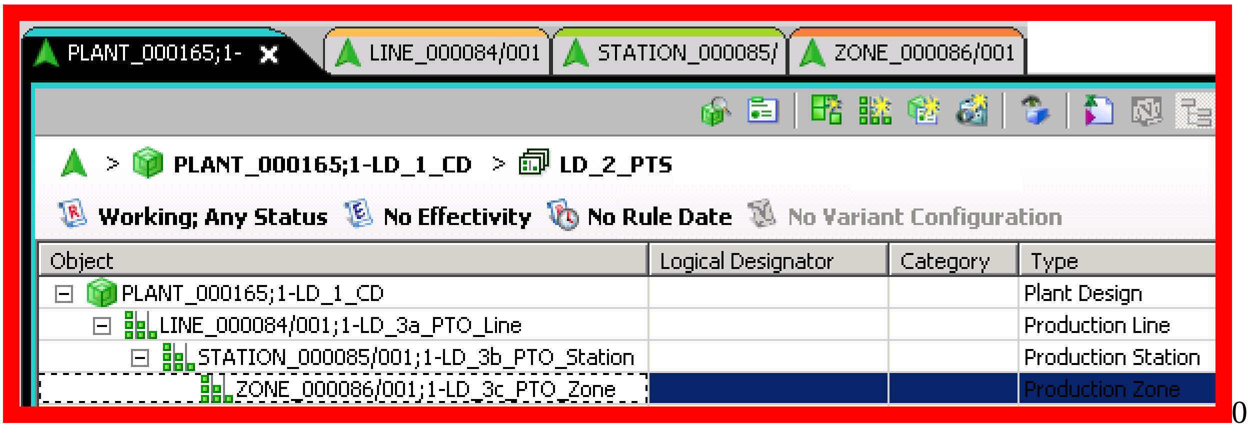
> PLANT_000165;1-LD_1_CD > LD_2_PTS

Working; Any Status No Effectivity No Rule Date No Variant Configuration

Object	Logical Designator	Category	Type
PLANT_000165;1-LD_1_CD			Plant Design
LINE_000084/001;1-LD_3a_PTO_Line			Production Line
STATION_000085/001;1-LD_3b_PTO_Station			Production Station

03b_16

16. Create a "Production Zone" partition under the station partition with **Name** = "LD_3c_PTO_Zone".



PLANT_000165;1- | LINE_000084/001 | STATION_000085/ | ZONE_000086/001

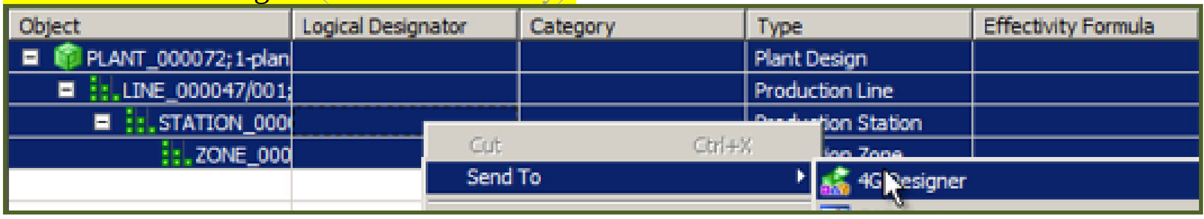
> PLANT_000165;1-LD_1_CD > LD_2_PTS

Working; Any Status No Effectivity No Rule Date No Variant Configuration

Object	Logical Designator	Category	Type
PLANT_000165;1-LD_1_CD			Plant Design
LINE_000084/001;1-LD_3a_PTO_Line			Production Line
STATION_000085/001;1-LD_3b_PTO_Station			Production Station
ZONE_000086/001;1-LD_3c_PTO_Zone			Production Zone

03b_17

17. Send to 4GDesigner (I don't know why).



Object	Logical Designator	Category	Type	Effectivity Formula
PLANT_000072;1-plan			Plant Design	
LINE_000047/001;			Production Line	
STATION_000			Production Station	
ZONE_000			ion Zone	

03b_18

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

For 4GD requirements you must do the following:

4.1. Create a Line Designer workset

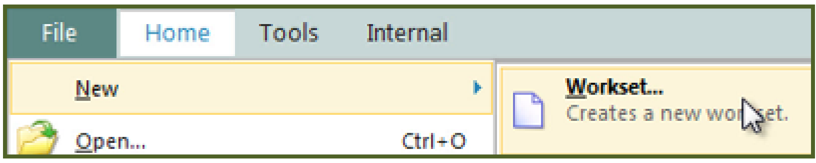
4.2. Create a Line Designer subset and add partitions to recipe

4.3. Add two Line Designer conveyors

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.

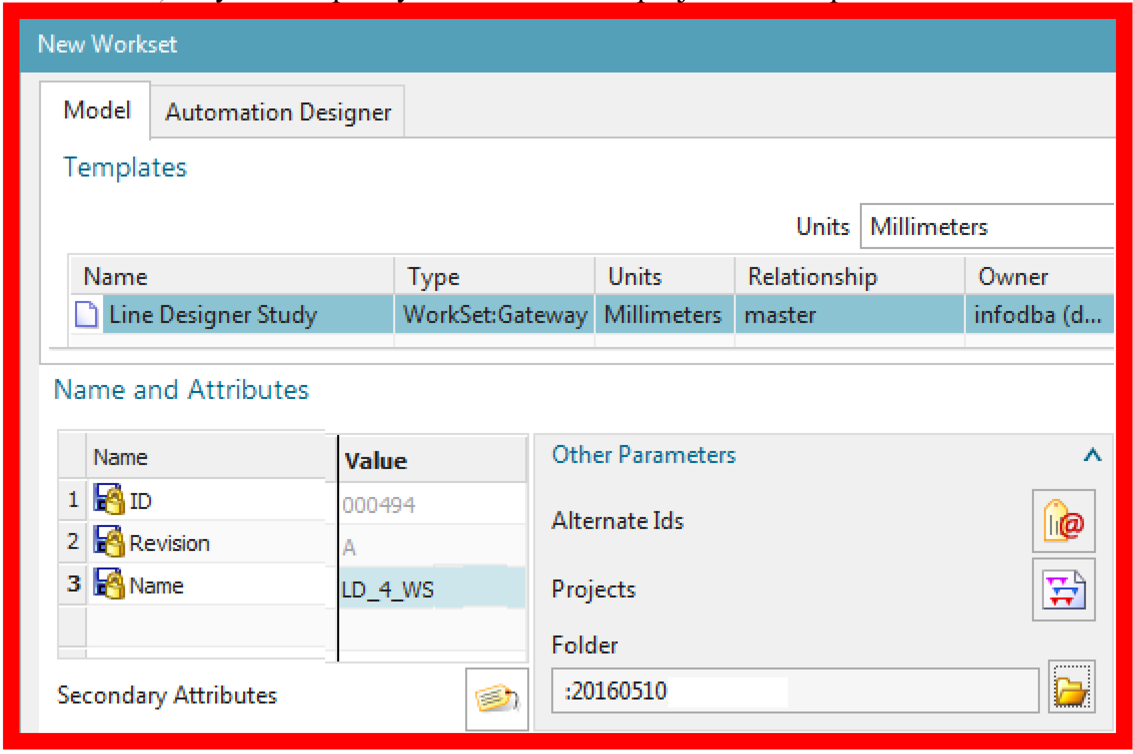
1. Create a new workset.



04b_01

2. Select **Model→Line Designer Study**.

3. Set **Name** = "LD_4_WS" (note that I selected folder 20160510_TERRY. This is where it will be place in TeamCenter). If you not specify a folder then the project will be put in "Newstuff".



New Workset

Model | Automation Designer

Templates

Units: Millimeters

Name	Type	Units	Relationship	Owner
Line Designer Study	WorkSet:Gateway	Millimeters	master	infodba (d...

Name and Attributes

	Name	Value
1	ID	000494
2	Revision	A
3	Name	LD_4_WS

Secondary Attributes

Other Parameters

Alternate Ids

Projects

Folder: :20160510

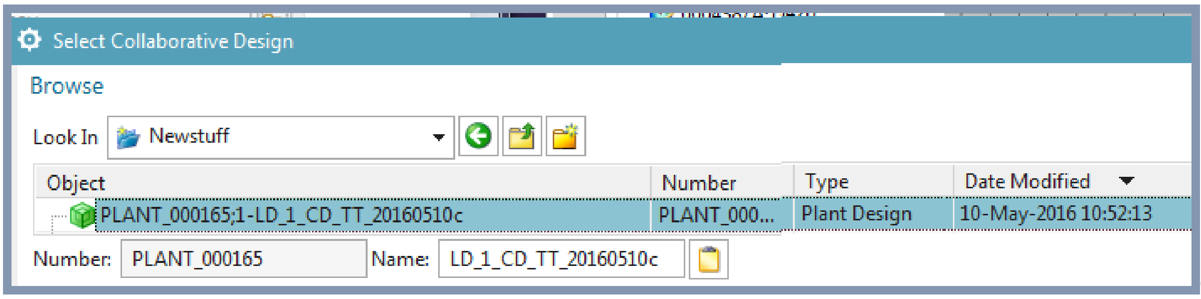
04b_02

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

4.2. Create 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".



Select Collaborative Design

Browse

Look In: Newstuff

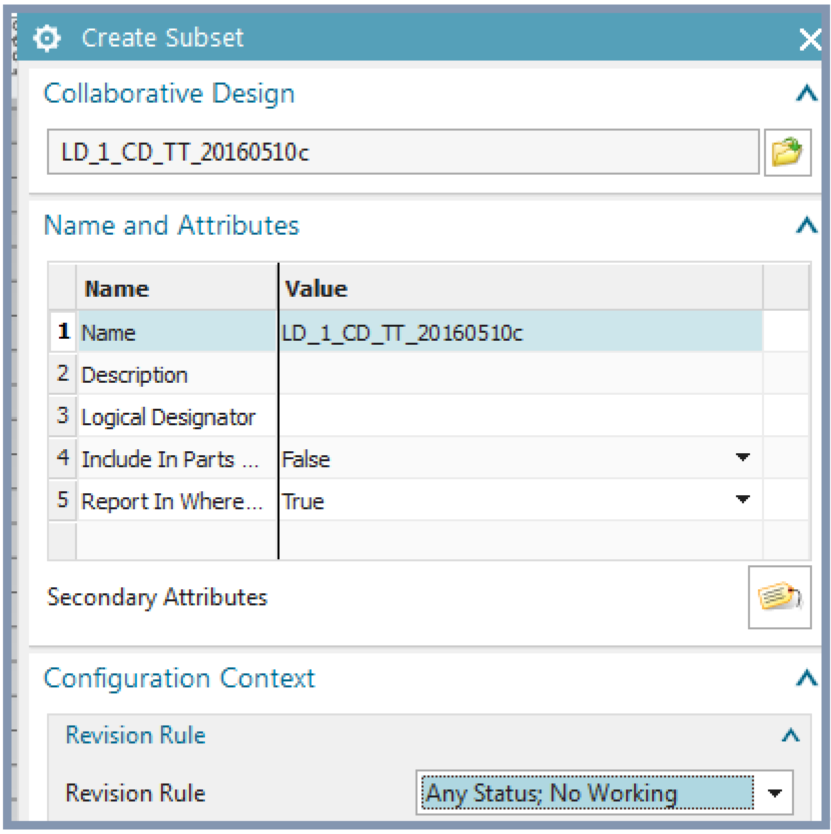
Object	Number	Type	Date Modified
PLANT_000165;1-LD_1_CD_TT_20160510c	PLANT_000...	Plant Design	10-May-2016 10:52:13

Number: PLANT_000165 Name: LD_1_CD_TT_20160510c

04b_03

2. Click **OK**.

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



Create Subset

Collaborative Design

LD_1_CD_TT_20160510c

Name and Attributes

	Name	Value
1	Name	LD_1_CD_TT_20160510c
2	Description	
3	Logical Designator	
4	Include In Parts ...	False
5	Report In Where...	True

Secondary Attributes

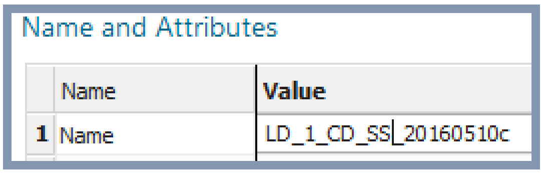
Configuration Context

Revision Rule

Revision Rule: Any Status; No Working

04b_04

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

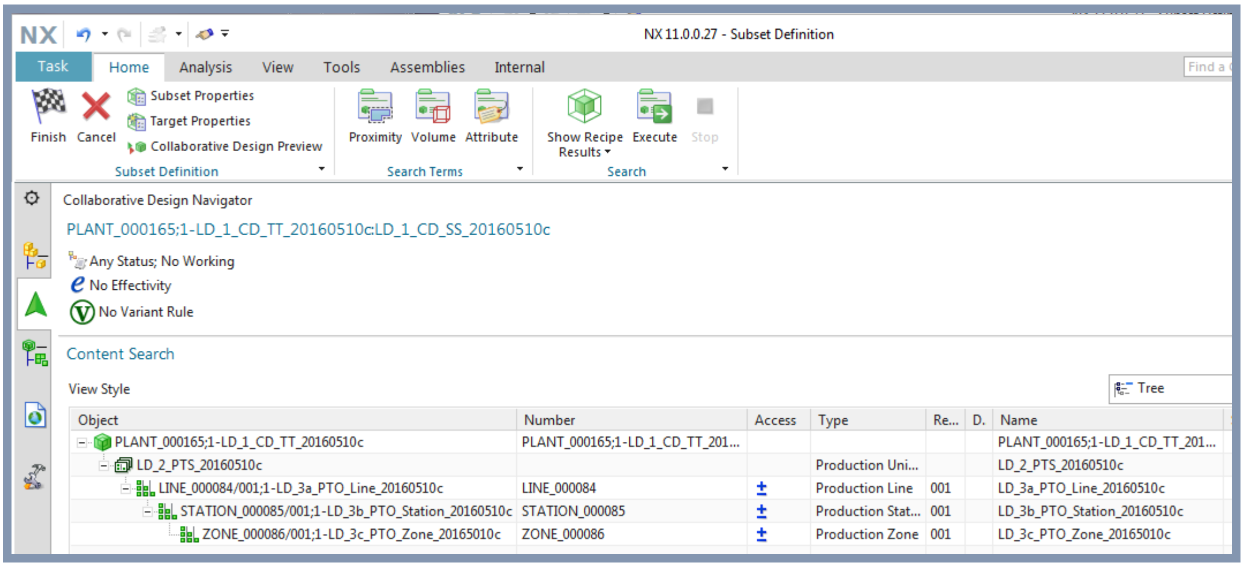


Name and Attributes

	Name	Value
1	Name	LD_1_CD_SS_20160510c

04b_05

5. Click **OK**. The Subset Definition appears.



Collaborative Design Navigator

PLANT_000165;1-LD_1_CD_TT_20160510c:LD_1_CD_SS_20160510c

Any Status; No Working

No Effectivity

No Variant Rule

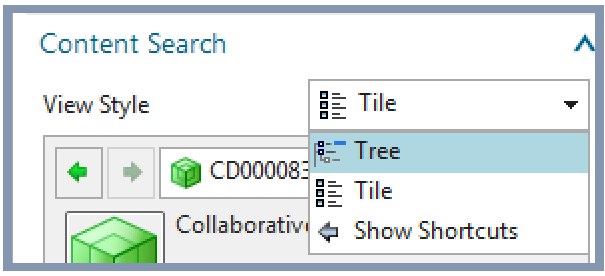
Content Search

View Style: Tree

Object	Number	Access	Type	Re...	D.	Name
PLANT_000165;1-LD_1_CD_TT_20160510c	PLANT_000165;1-LD_1_CD_TT_201...					PLANT_000165;1-LD_1_CD_TT_201...
LD_2_PTS_20160510c			Production Uni...			LD_2_PTS_20160510c
LINE_000084/001;1-LD_3a_PTO_Line_20160510c	LINE_000084	±	Production Line	001		LD_3a_PTO_Line_20160510c
STATION_000085/001;1-LD_3b_PTO_Station_20160510c	STATION_000085	±	Production Stat...	001		LD_3b_PTO_Station_20160510c
ZONE_000086/001;1-LD_3c_PTO_Zone_20165010c	ZONE_000086	±	Production Zone	001		LD_3c_PTO_Zone_20165010c

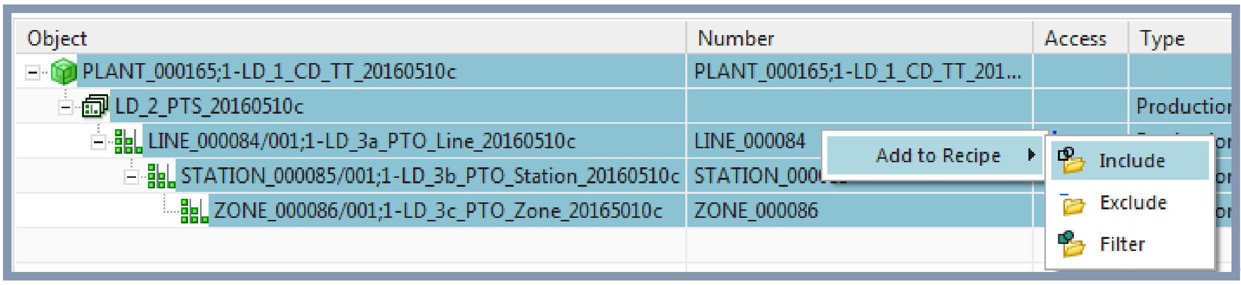
04b_06

6. Select the tree if not shown as above.



04b_07

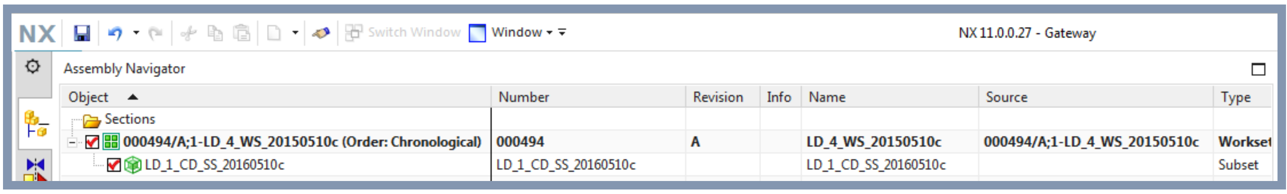
7. Select all, right click and select **Add to Recipe→Include**. This adds the subset to the recipe.



Object	Number	Access	Type
PLANT_000165;1-LD_1_CD_TT_20160510c	PLANT_000165;1-LD_1_CD_TT_201...		
LD_2_PTS_20160510c			Production
LINE_000084/001;1-LD_3a_PTO_Line_20160510c	LINE_000084		
STATION_000085/001;1-LD_3b_PTO_Station_20160510c	STATION_000		
ZONE_000086/001;1-LD_3c_PTO_Zone_20165010c	ZONE_000086		

04b_08

8. Click **Finish**. You see the workset and subset listed. Note that you are in the Gateway.



Assembly Navigator

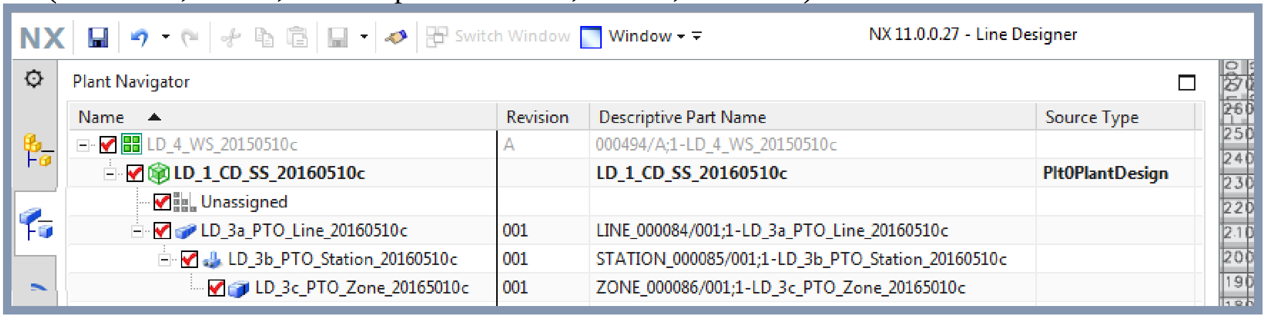
Object	Number	Revision	Info	Name	Source	Type
Sections						
000494/A;1-LD_4_WS_20150510c (Order: Chronological)	000494	A		LD_4_WS_20150510c	000494/A;1-LD_4_WS_20150510c	Workset
LD_1_CD_SS_20160510c	LD_1_CD_SS_20160510c			LD_1_CD_SS_20160510c		Subset

04b_09

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).

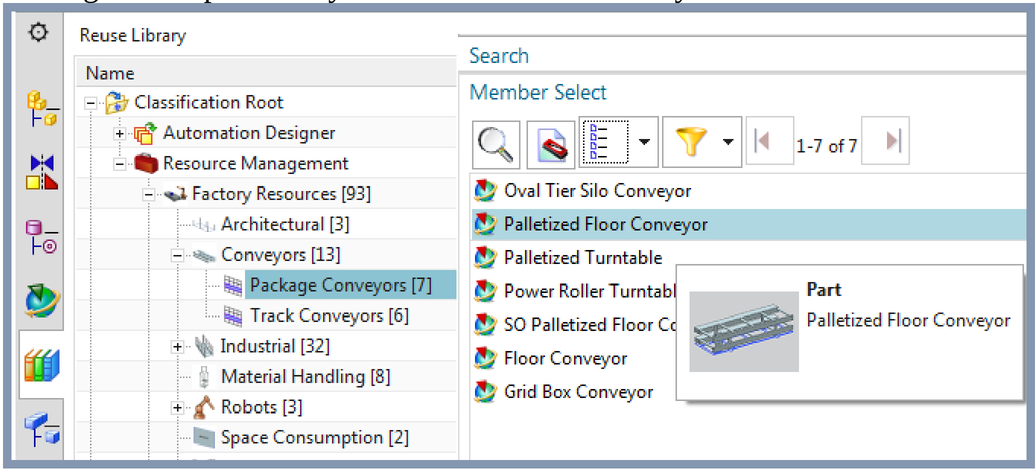


Plant Navigator

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			
☑ 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_20165010c	001	ZONE_000086/001;1-LD_3c_PTO_Zone_20165010c	

04b_10

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



Reuse Library

Name
- Classification Root
  - Automation Designer
  - Resource Management
    - Factory Resources [93]
      - Architectural [3]
      - Conveyors [13]
        - Package Conveyors [7]
        - Track Conveyors [6]
      - Industrial [32]
      - Material Handling [8]
      - Robots [3]
      - Space Consumption [2]

Search

Member Select

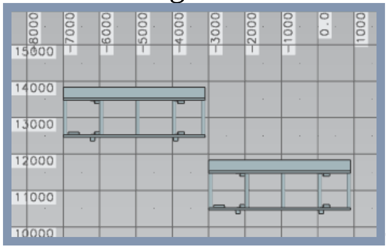
1-7 of 7

- Oval Tier Silo Conveyor
- Palletized Floor Conveyor
- Palletized Turntable
- Power Roller Turntabl
- SO Palletized Floor Co
- Floor Conveyor
- Grid Box Conveyor

Part
Palletized Floor Conveyor

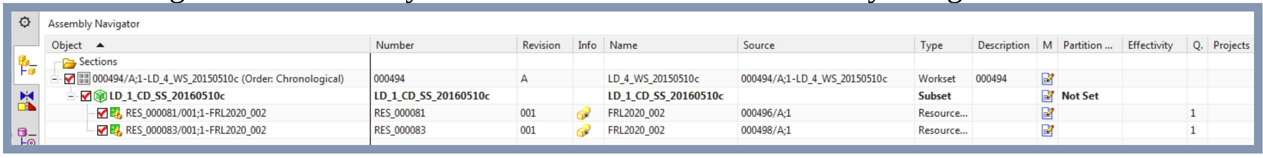
04b_11

The following shows the resulting conveyors.



04b_12

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

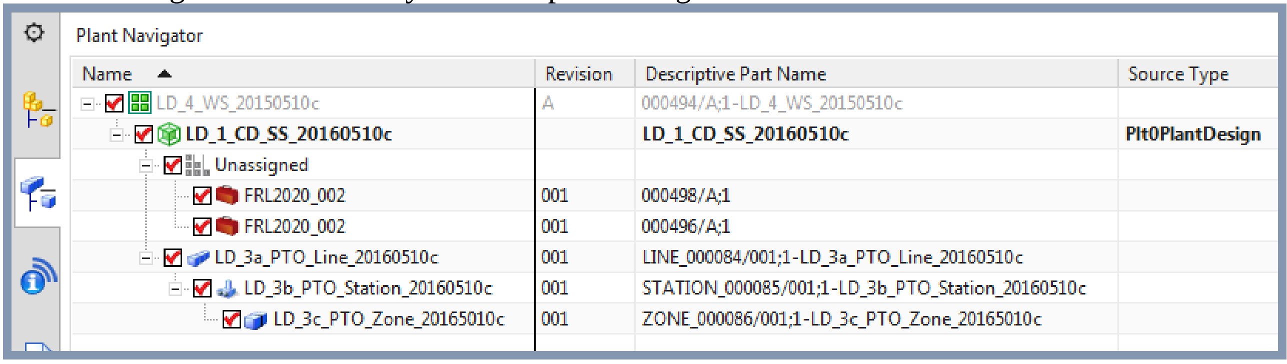


Assembly Navigator

Object	Number	Revision	Info	Name	Source	Type	Description	M	Partition ...	Effectivity	Q.	Projects
Sections												
☑ 000494/A;1-LD_4_WS_20150510c (Order: Chronological)	000494	A		LD_4_WS_20150510c	000494/A;1-LD_4_WS_20150510c	Workset	000494					
☑ **LD_1_CD_SS_20160510c**	**LD_1_CD_SS_20160510c**			**LD_1_CD_SS_20160510c**		**Subset**			**Not Set**			
☑ RES_000081/001;1-FRL2020_002	RES_000081	001		FRL2020_002	000496/A;1	Resource...					1	
☑ RES_000083/001;1-FRL2020_002	RES_000083	001		FRL2020_002	000498/A;1	Resource...					1	

04b_13

The following shows the conveyors in the plant navigator.



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_20165010c	001	ZONE_000086/001;1-LD_3c_PTO_Zone_20165010c	

04b_14

5. Create Automation Designer workset (and Collaborative Design, subset) and Engineering Objects

Now you need to create similar components for Automation Designer as you did for Line Designer (but only using Automation Designer).

5.1. Create project workset (and Collaborative Design + subset)
5.2. Create Engineering Object Definitions
5.3. Create Engineering Object names and aspect naming rules
5.4. Add Engineering Objects

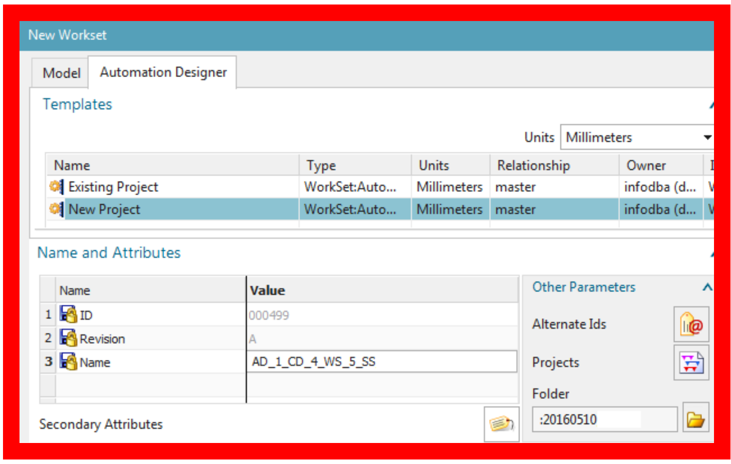
5.1. Create project workset (and Collaborative Design + subset)

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. 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".



Units: Millimeters

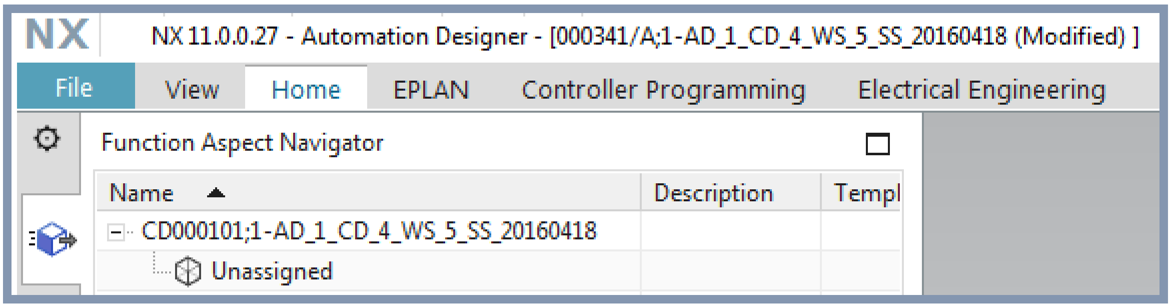
Name	Type	Units	Relationship	Owner
Existing Project	WorkSet:Auto...	Millimeters	master	infodba (d...
New Project	WorkSet:Auto...	Millimeters	master	infodba (d...

	Name	Value
1	ID	000499
2	Revision	A
3	Name	AD_1_CD_4_WS_5_SS

Folder: :20160510

05b_01

4. Click **OK**.

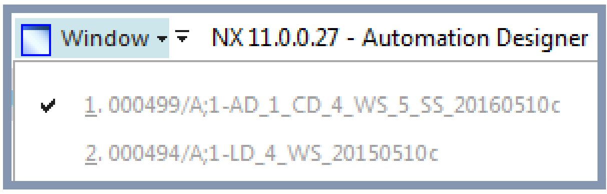


NX 11.0.0.27 - Automation Designer - [000341/A;1-AD_1_CD_4_WS_5_SS_20160418 (Modified)]

CD000101;1-AD_1_CD_4_WS_5_SS_20160418
Unassigned

05b_02

Note that Line Designer workset and the Automation Designer Collaborative Design, workset and subset are open. For the past month for some reason they are greyed out can cannot switch between them.



NX 11.0.0.27 - Automation Designer

✓ 1. 000499/A;1-AD_1_CD_4_WS_5_SS_20160510c
2. 000494/A;1-LD_4_WS_20150510c

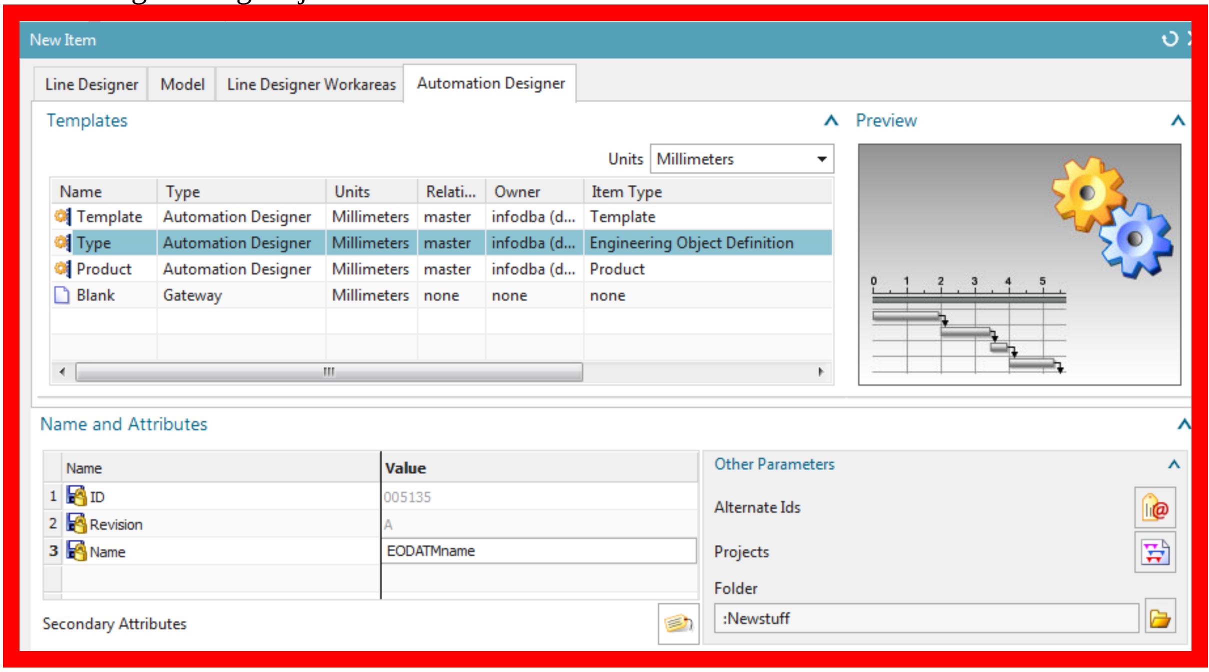
05_03

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.



Units: Millimeters

Name	Type	Units	Relati...	Owner	Item Type
Template	Automation Designer	Millimeters	master	infodba (d...	Template
Type	Automation Designer	Millimeters	master	infodba (d...	Engineering Object Definition
Product	Automation Designer	Millimeters	master	infodba (d...	Product
Blank	Gateway	Millimeters	none	none	none

	Name	Value
1	ID	005135
2	Revision	A
3	Name	EODATMname

Secondary Attributes

Other Parameters

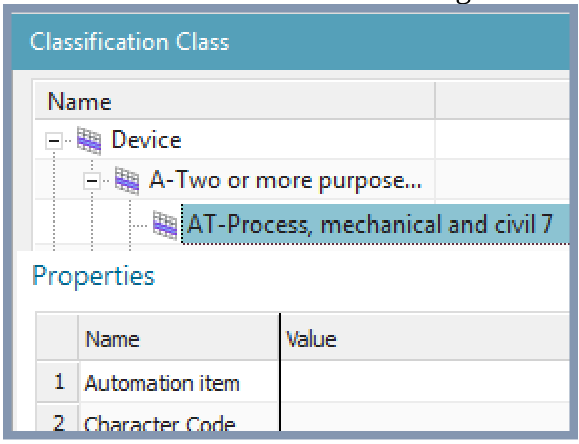
Alternate Ids

Projects

Folder: :Newstuff

05b_04

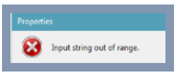
4. Click **OK**.
5. In the **Classification Class** dialog select **Device / A / AT**.



	Name	Value
1	Automation item	
2	Character Code	

05b_05

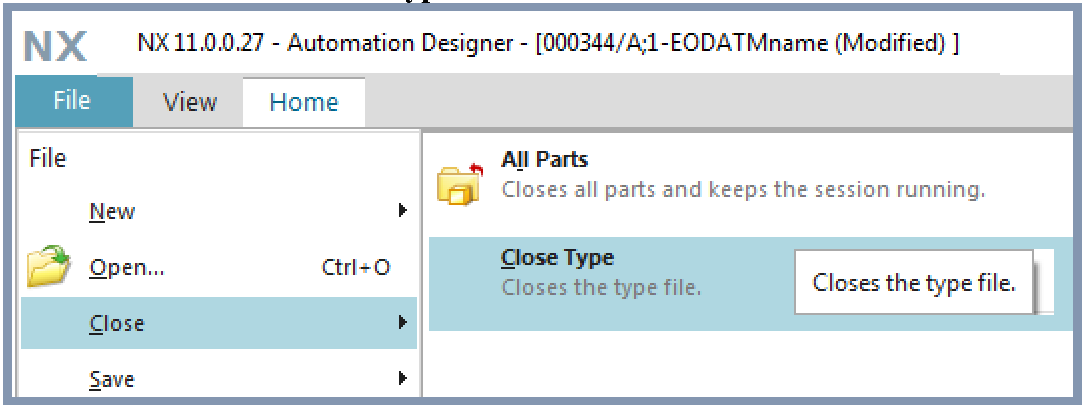
Click OK.



Input string out of range.

6. Click **OK**.

7. Select **File→Close→Close type**.


NX 11.0.0.27 - Automation Designer - [000344/A;1-EODATMname (Modified)]

All Parts

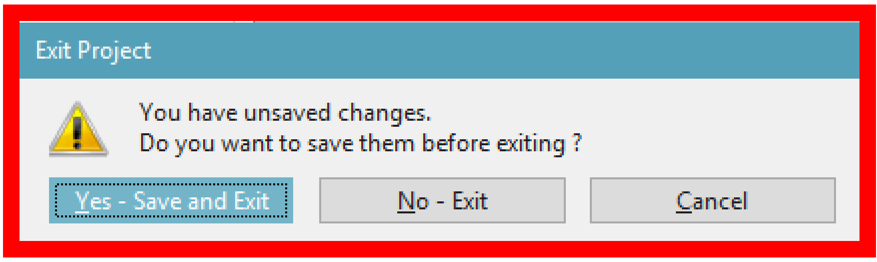
Closes all parts and keeps the session running.

Close Type

Closes the type file.

Closes the type file.

05b_06

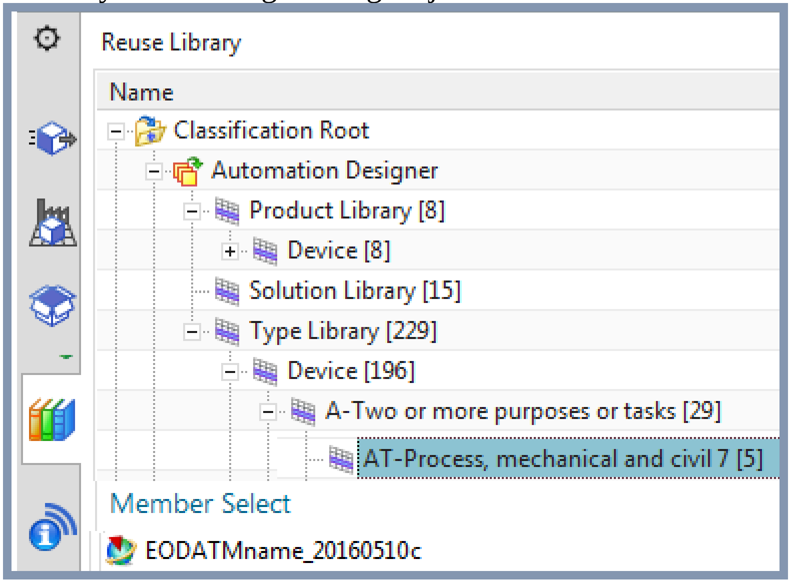

Exit Project

You have unsaved changes.
Do you want to save them before exiting ?

05c_01

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

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



05b_07

10. Create the remaining Engineering Object Definitions:

Type	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

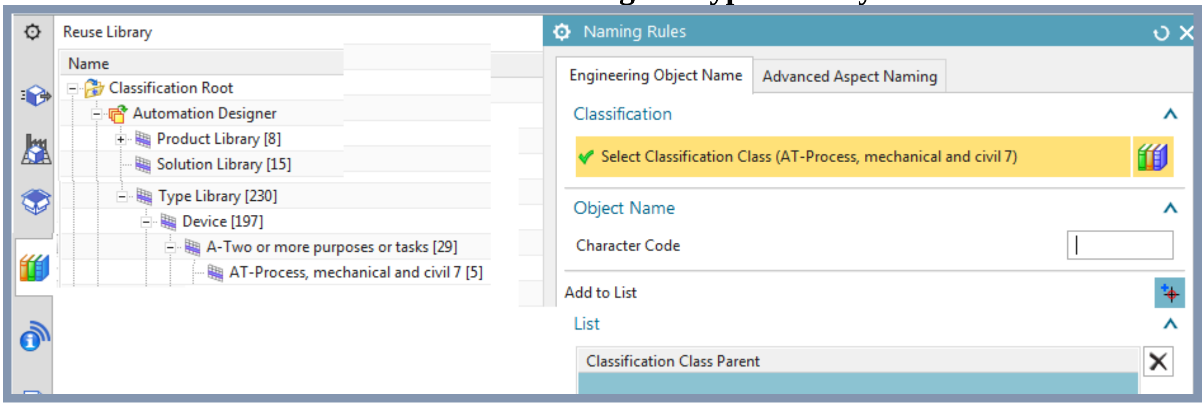
Naming Rules allow you to define how the designations of Engineering Objects are generated. Customer Default Naming Rules apply to all projects and to all Engineering Obejcts, regardless of their Classification Class.

Project-specific Naming Rules apply to the project in which you define them. Every rule applies to all Engineering Objects with a specified Classification Class.
There are two types of project-specific Naming Rules.
• Aspect Naming Rules
These rules define how to generate the Aspect Names of Engineering Objects with the same Classification Class. You define separate rules for each Aspect Navigator. Use them to define Aspect Names that deviate from the Engineering Object's Name.
• Rules for creating the Object Name
These rules define which string is used in the Object Name of Engineering Objects with the same Classification Class.
Use them to define Object Names that deviate from the character code defined by the Engineering Objects' Types.

In the customer defaults, on the **Naming Rules** tab, you can specify Customer Default Naming Rules. They apply to all projects and all Engineering Objects, regardless of their classification, unless they are overwritten by project-specific Naming Rules.

Engineering Object names

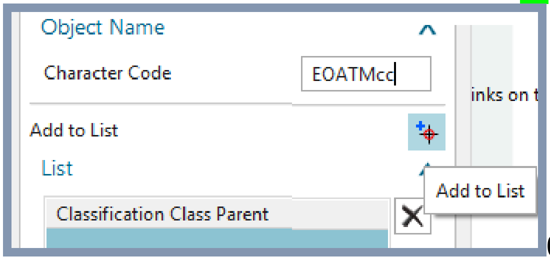
You now create the Engineering Object names that will display in the aspect tree.
1. Click on **Home→Naming Rules**.
2. Select **Classification Root/Automation Designer/Type Library/Device/A/AT**.



Select Classification Class (AT-Process, mechanical and civil 7)

05b_08

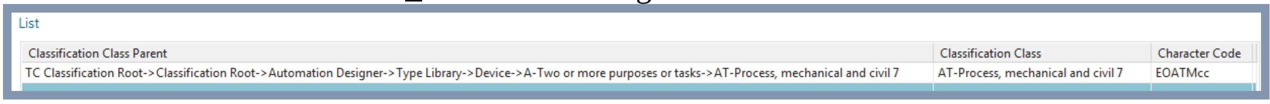
3. For “Character Code” enter “EOATMcc”.



Character Code EOATMcc

05b_09

4. Click **Add to List** 05c_02. The following is the result.



List

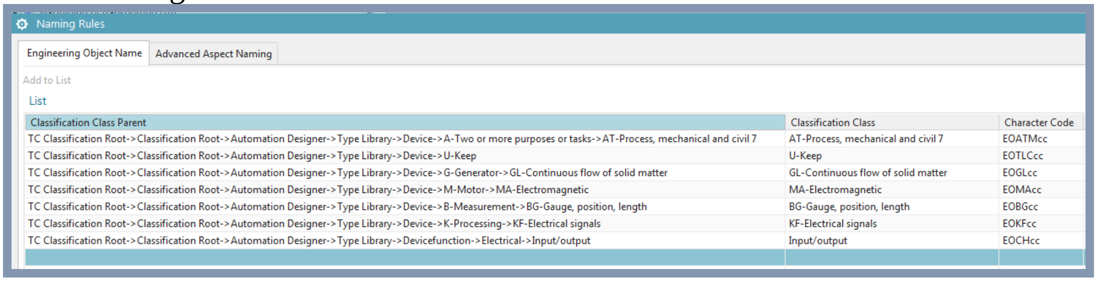
Classification Class Parent	Classification Class	Character Code
TC Classification Root->Classification Root->Automation Designer->Type Library->Device->A-Two or more purposes or tasks->AT-Process, mechanical and civil 7	AT-Process, mechanical and civil 7	EOATMcc

05b_10

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.



Classification Class Parent	Classification Class	Character Code
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	EOTLCcc
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

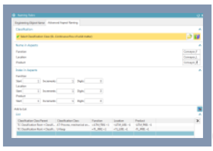
05b_11

xxxxxAdvanced aspect naming

Advancing aspect naming. This naming can override the Engineering Object names.

If the customer defaults specify that the Aspect Naming Rules shall apply automatically and if an Aspect Naming Rule was defined for the object's Classification Class, Automation Designer uses the rule to generate the Aspect Names.

1. Under “Name in Aspects” for “Function” enter “Conveyor_F”.
2. Add “Conveyor_L” and “Conveyor_P”.
3. Click Add to List.

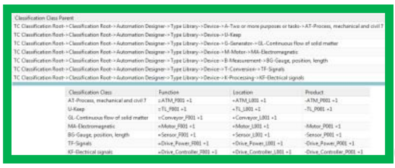


05b_12

4. Set the “Name in aspects” for the remaining Engineering Objects.

Engineering Object type	Classification	Name in aspects
ATM		
TL		
Conveyor	Device -> G Generator -> GL Continuous flow of solid Materials	Conveyor
Sensors	Device -> B Measurement -> BG Gauge, position, length	Sensor
Motor	Device -> M Motor -> MA Electromagnetic	Motor
G120D Power Module	Device -> T Conversion -> TF Signals	Drive_Power
G120D Control Module	Device -> K Processing -> KF Electrical Signals	Drive_Controller

Result:



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 T
TC Classification Root->Classification Root->Automation Designer->Type Library->Device->U-Keep
TC Classification Root->Classification Root->Automation Designer->Type Library->Device->G-Generator->GL-Continuous flow of solid matter
TC Classification Root->Classification Root->Automation Designer->Type Library->Device->M-Motor->MA-Electromagnetic
TC Classification Root->Classification Root->Automation Designer->Type Library->Device->B-Measurement->BG-Gauge, position, length
TC Classification Root->Classification Root->Automation Designer->Type Library->Device->T-Conversion->TF-Signals
TC Classification Root->Classification Root->Automation Designer->Type Library->Device->K-Processing->KF-Electrical signals

Classification Class	Function	Location	Product
AT-Process, mechanical and civil T	=ATM_F001 +1	+ATM_L001 +1	-ATM_P001 +1
U-Keep	=TL_F001 +1	+TL_L001 +1	-TL_P001 +1
GL-Continuous flow of solid matter	=Conveyor_F001 +1	+Conveyor_L001 +1	
MA-Electromagnetic	=Motor_F001 +1	+Motor_L001 +1	-Motor_P001 +1
BG-Gauge, position, length	=Sensor_F001 +1	+Sensor_L001 +1	-Sensor_P001 +1
TF-Signals	=Drive_Power_F001 +1	+Drive_Power_L001 +1	-Drive_Power_P001 +1
KF-Electrical signals	=Drive_Controller_F001 +1	+Drive_Controller_L001 +1	-Drive_Controller_P001 +1

05b_13

5.4. Add Engineering Objects

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:

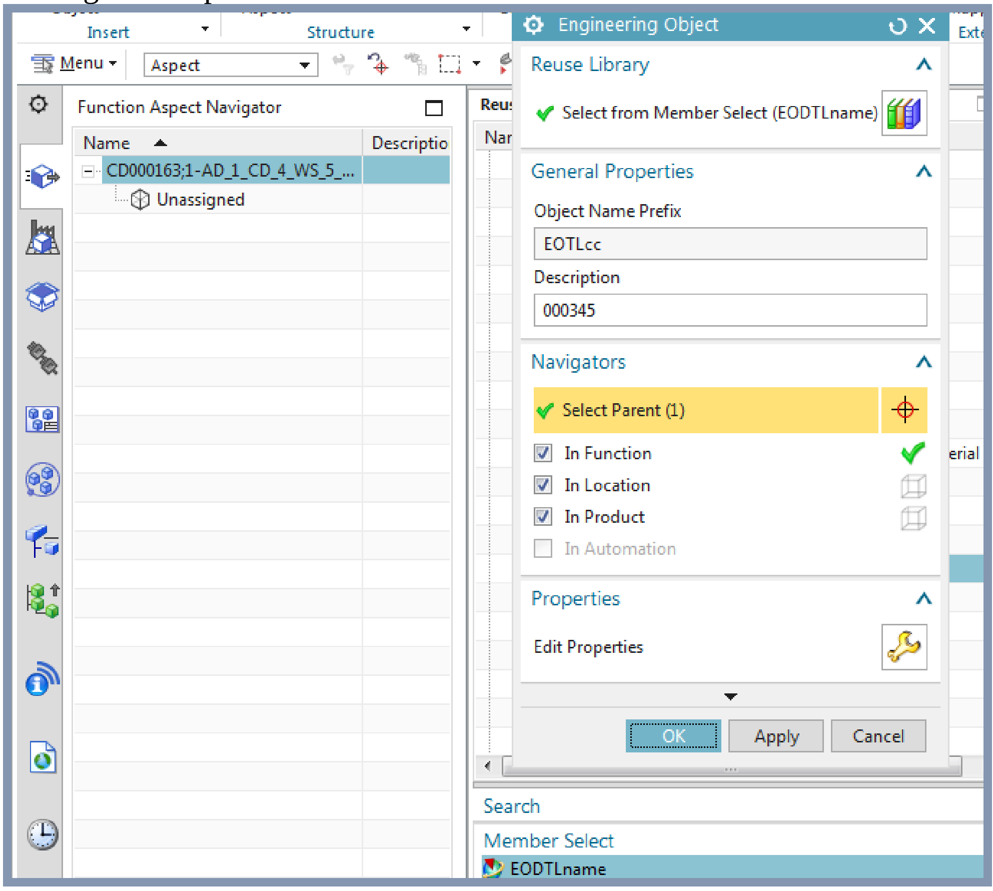
- Functional aspect: What is the functional purpose of an object within the production system or machine or what does the object actually do?
- Location aspect: Where in the production system or machine is the object installed, builtin, or placed, and is it in itself an installation place for other objects?
- 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).

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

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

1. Drag and drop EOTLcc.



Engineering Object

Reuse Library

✓ Select from Member Select (EODTLname)

General Properties

Object Name Prefix: EOTLcc

Description: 000345

Navigators

✓ Select Parent (1)

☑ In Function

☑ In Location

☑ In Product

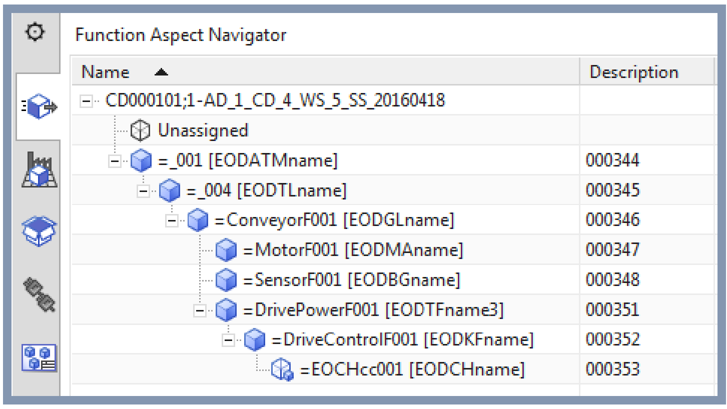
□ In Automation

Properties

Edit Properties

05b_14

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

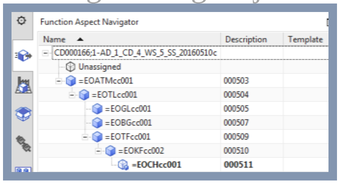


Function Aspect Navigator

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 [EODBGname]	000348
=DrivePowerF001 [EODTFname3]	000351
=DriveControlF001 [EODKFname]	000352
=EOCHcc001 [EODCHname]	000353

05b_15

TERRY: sometimes this works, usually not. Later in this doc you will therefore often see the following, with the Engineering Object names, not the advanced aspect names.



Function Aspect Navigator

Name	Description	Template
CD000166;1-AD_1_CD_4_WS_5_SS_20160510c		
Unassigned		
=EOATMcc001	000503	
=EOTLcc001	000504	
=EOGLcc001	000505	
=EOBGcc001	000507	
=EOTFcc001	000509	
=EOKFcc002	000510	
=EOCHcc001	**000511**	

5.5. Location-Product aspects

Introduction to Engineering Objects and aspects

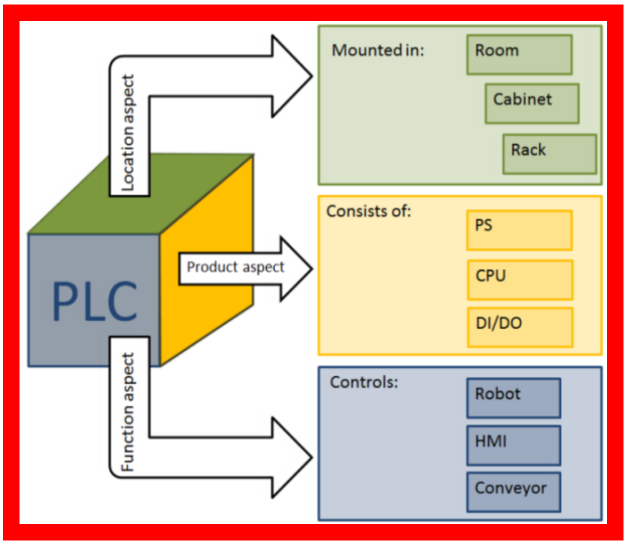
The IEC 81346 standard describes principles for structuring and naming objects and their associated information in industrial systems, installations, equipment, and industrial products. The goal of these principles is to handle the large sets of information that are available in these systems efficiently.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



05c_03

PIC

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.

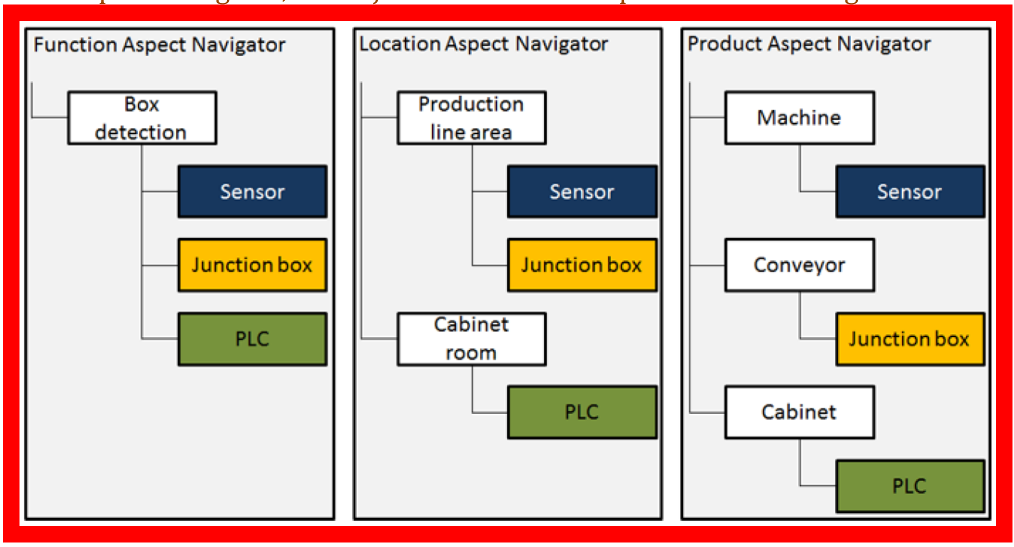
Because each aspect represents a view that may structure objects differently, Automation Designer provides several Aspect Navigators. Each *Aspect Navigator* represents one view of planning. In each Aspect Navigator, you define the hierarchical structure of the Engineering Objects for the current aspect.

One Engineering Object for independent aspect structures

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.



05c_04

Note

For program blocks and EPLAN macros it is enough if you place them in one Aspect Navigator. Engineering Objects for control-related hardware and program blocks are also visible in the **Automation** Navigator.

When you select an Engineering Object in one navigator, it is also selected in the other navigators in which it is available.

Aspect Navigators

- 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.

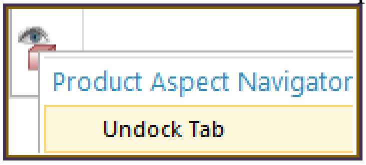
Aspects are predefined engineering categories according to IED 81346 that specifies how to organize the different engineering objects and elements based on

1. Function (symbol =): Functional description (main function, sub-function, etc.).
2. Location (symbol +): Physical location (hall, sector, cabinet, etc.).
3. Product (symbol -): How the parts are purchased.

05_25

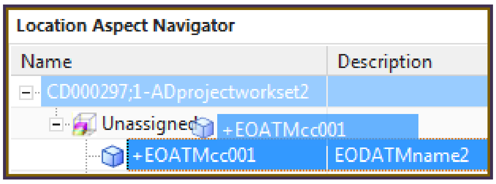
Now you will configure the location and product aspects.

1. Undock the location and product aspects.



05_28

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

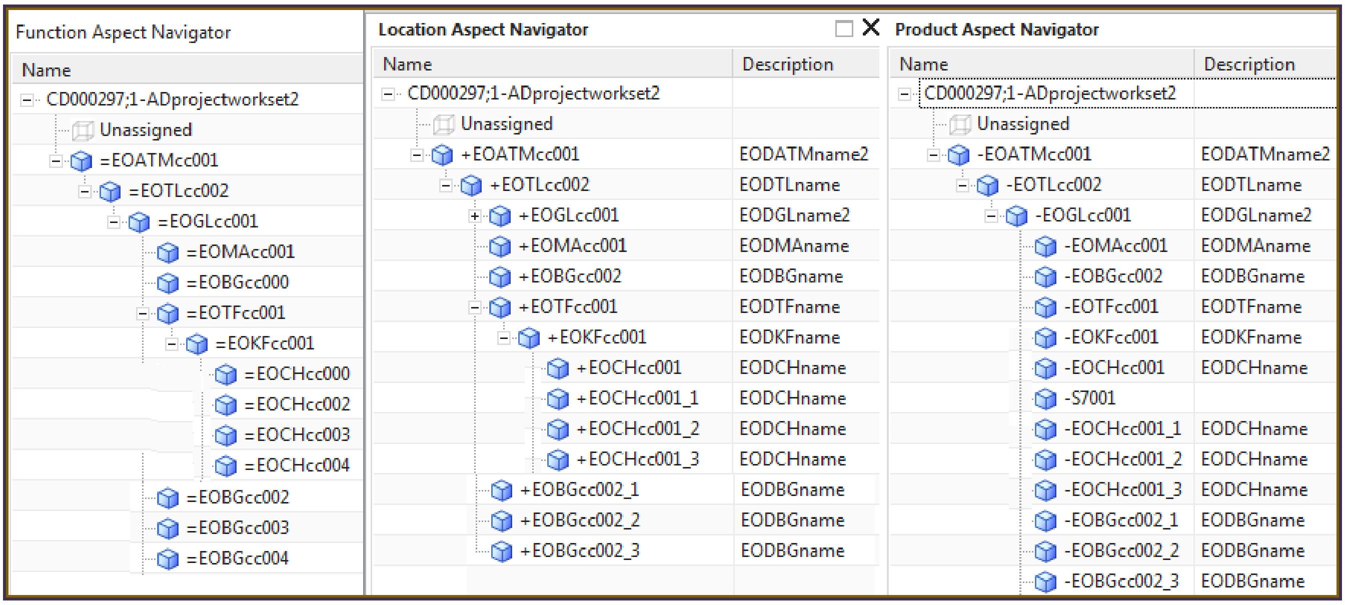


Location Aspect Navigator

Name	Description
CD000297;1-ADprojectworkset2	
Unassigned	
+EOATMcc001	EODATMname2

05_29

3. The result should be like this.



Function Aspect Navigator

Name
CD000297;1-ADprojectworkset2
Unassigned
=EOATMcc001
=EOTLcc002
=EOGLcc001
=EOMAcc001
=EOBGcc000
=EOTFcc001
=EOKFcc001
=EOCHcc000
=EOCHcc002
=EOCHcc003
=EOCHcc004
=EOBGcc002
=EOBGcc003
=EOBGcc004

Location Aspect Navigator

Name	Description
CD000297;1-ADprojectworkset2	
Unassigned	
+EOATMcc001	EODATMname2
+EOTLcc002	EODTLname
+EOGLcc001	EODGLname2
+EOMAcc001	EODMAname
+EOBGcc002	EODBGname
+EOTFcc001	EODTFname
+EOKFcc001	EODKFname
+EOCHcc001	EODCHname
+EOCHcc001_1	EODCHname
+EOCHcc001_2	EODCHname
+EOCHcc001_3	EODCHname
+EOBGcc002_1	EODBGname
+EOBGcc002_2	EODBGname
+EOBGcc002_3	EODBGname

Product Aspect Navigator

Name	Description
CD000297;1-ADprojectworkset2	
Unassigned	
-EOATMcc001	EODATMname2
-EOTLcc002	EODTLname
-EOGLcc001	EODGLname2
-EOMAcc001	EODMAname
-EOBGcc002	EODBGname
-EOTFcc001	EODTFname
-EOKFcc001	EODKFname
-EOCHcc001	EODCHname
-S7001	
-EOCHcc001_1	EODCHname
-EOCHcc001_2	EODCHname
-EOCHcc001_3	EODCHname
-EOBGcc002_1	EODBGname
-EOBGcc002_2	EODBGname
-EOBGcc002_3	EODBGname

05_30

Part 2. Config (non-template) Line Designer, EPLAN, TIA Portal

This part shows how to create the mapping and TIA Portal software for a single conveyor. You will not create a template and will not use expressions or ports until part 3 (to keep things simple).

6. Map Line Designer-Automation Designer.
7. Configure (non-template) EPLAN.
8. Configure (non-template) TIA Portal.

6. Map Line Designer-Automation Designer

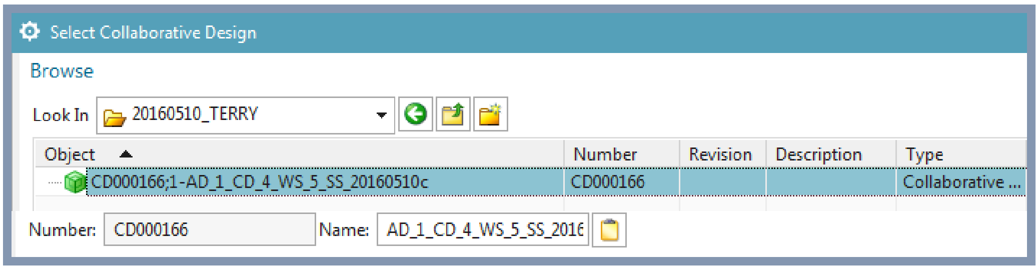
Mappings from mechanical and line design data to automation data eliminate the need for reentering data and make mechanical changes transparent. You can directly use the mechanical and line design data in Automation Designer.

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.

6.1. Link Automation Designer and Line Designer Collaborative Designs

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.



Select Collaborative Design

Browse

Look In: 20160510_TERRY

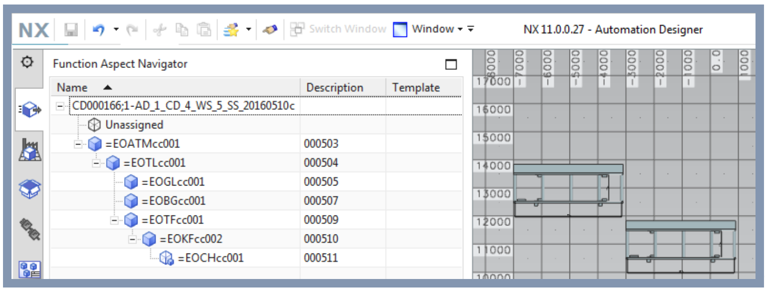
Object	Number	Revision	Description	Type
CD000166;1-AD_1_CD_4_WS_5_SS_20160510c	CD000166			Collaborative ...

Number: CD000166 Name: AD_1_CD_4_WS_5_SS_2016

06b_02

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

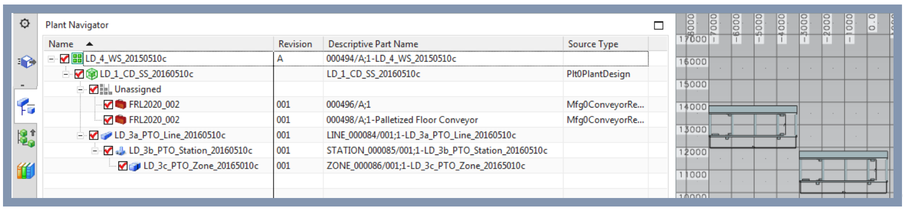
20160510: today the conveyors are there. Sometimes not.



Function Aspect Navigator

Name	Description	Template
CD000166;1-AD_1_CD_4_WS_5_SS_20160510c		
Unassigned		
=EOATMcc001	000503	
=EOTLcc001	000504	
=EOGLcc001	000505	
=EOBGcc001	000507	
=EOTFcc001	000509	
=EOKFcc002	000510	
=EOCHcc001	000511	

06b_03



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	000496/A;1	Mfg0ConveyorRe...
☑ FRL2020_002	001	000498/A;1-Palletized Floor Conveyor	Mfg0ConveyorRe...
☑ 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_20165010c	001	ZONE_000086/001;1-LD_3c_PTO_Zone_20165010c	

06b_04

20160509 they were not there.

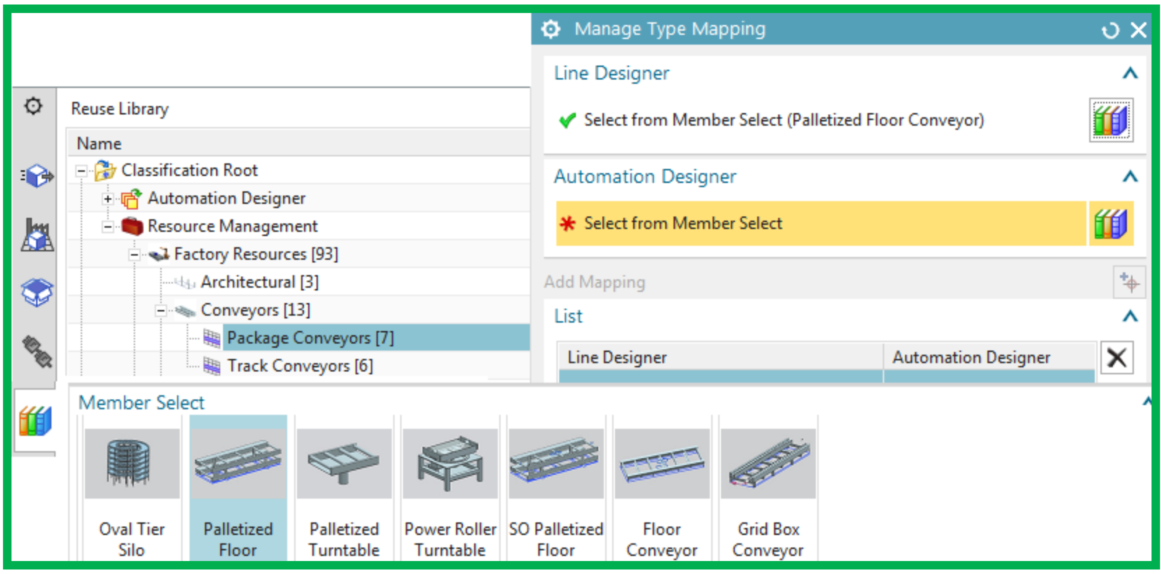
6.2. Manage type mapping

NOTHING ON THIS IN USER GUIDE. Caution: this dialog is buggy. Sometimes you don’t see mappings, but they are there.

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**.

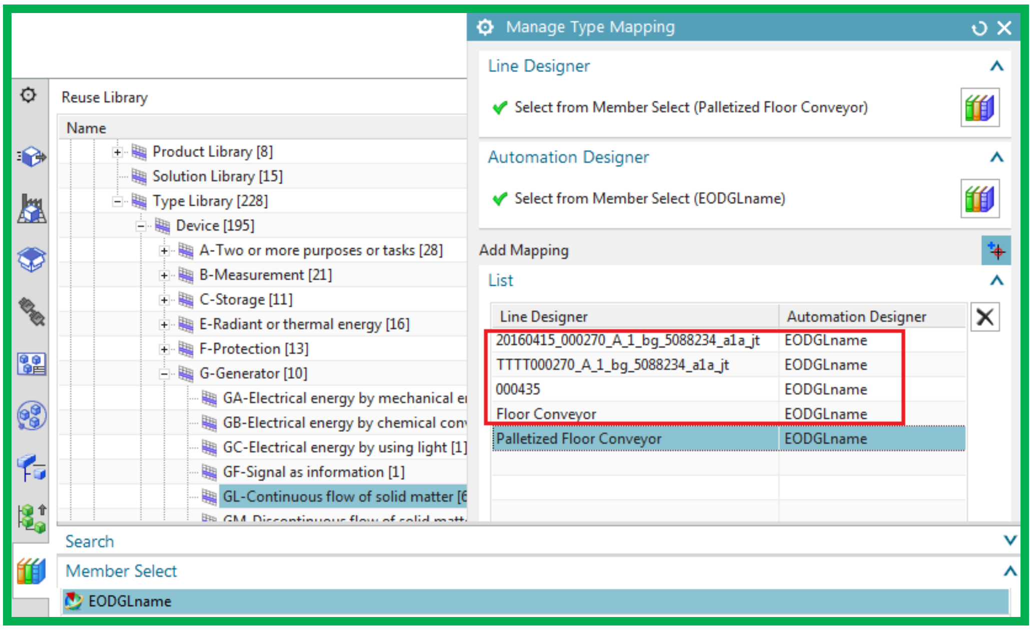
1. Open the **Manage Type Mapping** dialog.

2. Under **Line Designer** select the conveyor.



06b_05

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


Reuse Library
Name
Product Library [8]
Solution Library [15]
Type Library [228]
Device [195]
A-Two or more purposes or tasks [28]
B-Measurement [21]
C-Storage [11]
E-Radiant or thermal energy [16]
F-Protection [13]
G-Generator [10]
GA-Electrical energy by mechanical e
GB-Electrical energy by chemical con
GC-Electrical energy by using light [1]
GF-Signal as information [1]
GL-Continuous flow of solid matter [6
Manage Type Mapping
Line Designer
Select from Member Select (Palletized Floor Conveyor)
Automation Designer
Select from Member Select (EODGLname)
Add Mapping
List
Search
Member Select
EODGLname

Line Designer	Automation Designer
20160415_000270_A_1_bg_5088234_a1a_jt	EODGLname
TTTT000270_A_1_bg_5088234_a1a_jt	EODGLname
000435	EODGLname
Floor Conveyor	EODGLname
Palletized Floor Conveyor	EODGLname

06b_06

3. I tried to delete all of the unneeded. But the logic behind the dialog is strange.

06b_07

6.3. Manage object mapping

Sometimes this works, sometimes not. 20160510 works. 20160509 not work. 20160426 works. 20160420 not work. Good luck. Note: Previously.. I forgot to add MA and put in right places. I added here.

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:

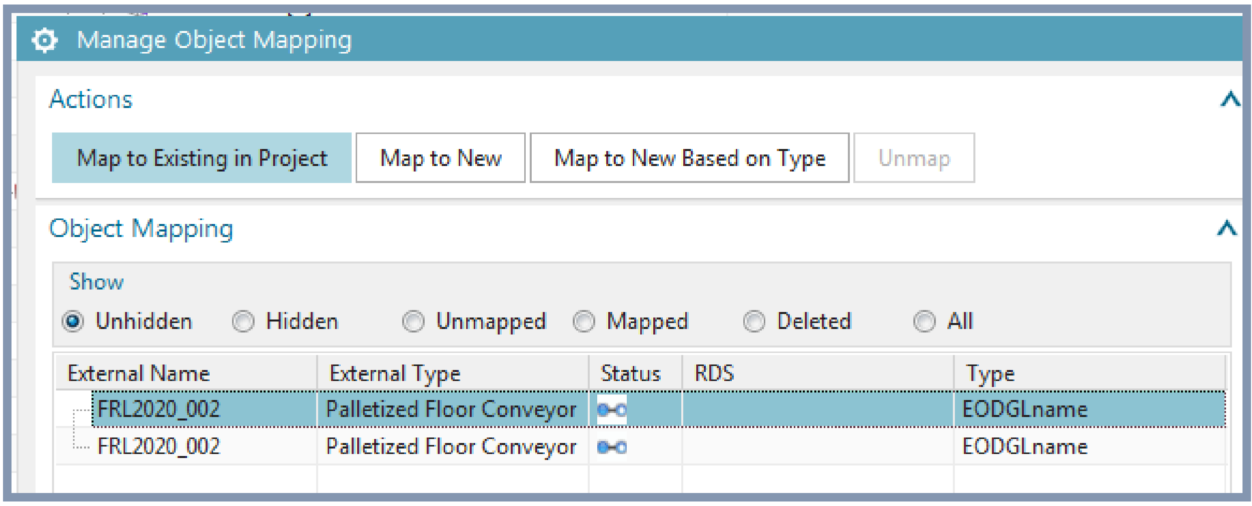
1. Map to existing
2. Map to new
3. Map to new based on type

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.



Manage Object Mapping

Actions

Object Mapping

Show

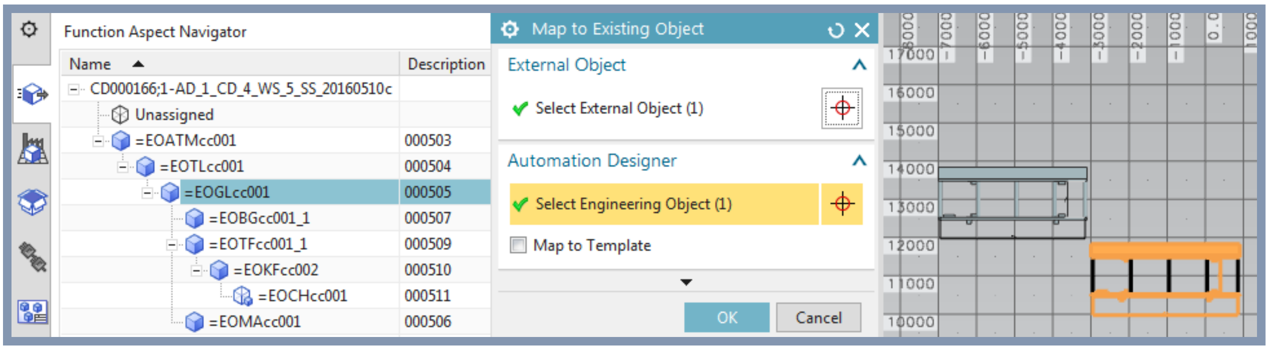
● Unhidden ○ Hidden ○ Unmapped ○ Mapped ○ Deleted ○ All

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

06b_08

3. Click **Map to existing in project**.

4. Select GL.



Function Aspect Navigator

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

Map to Existing Object

External Object

✓ Select External Object (1)

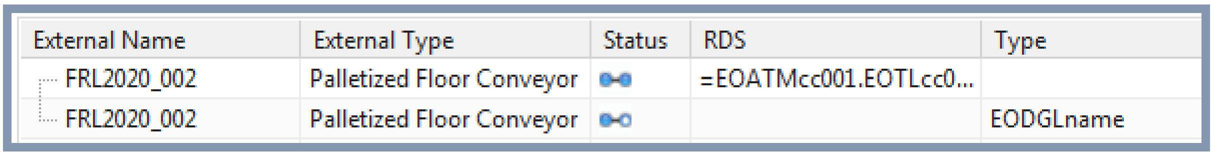
Automation Designer

✓ Select Engineering Object (1)

□ Map to Template

06b_09

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

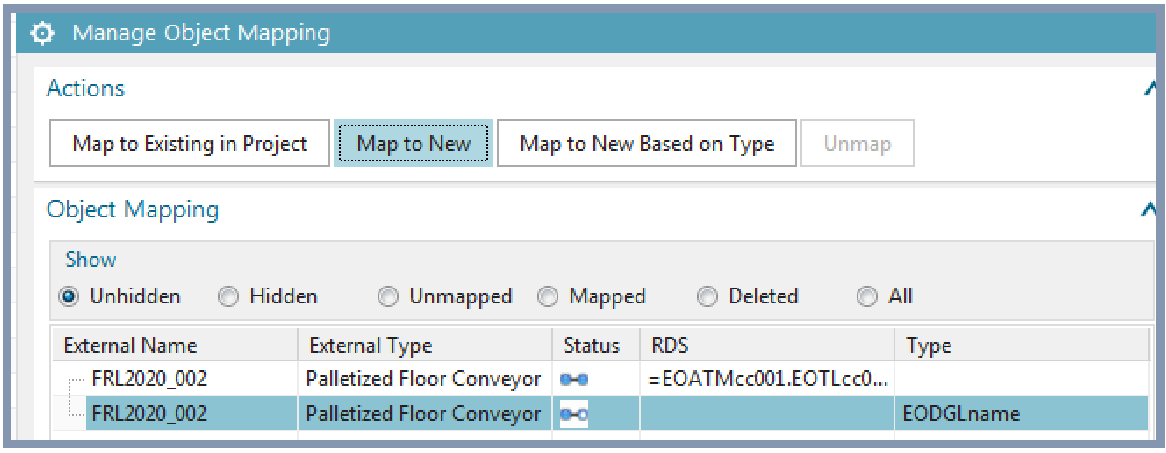
06b_10

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**.



Manage Object Mapping

Actions

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

Object Mapping

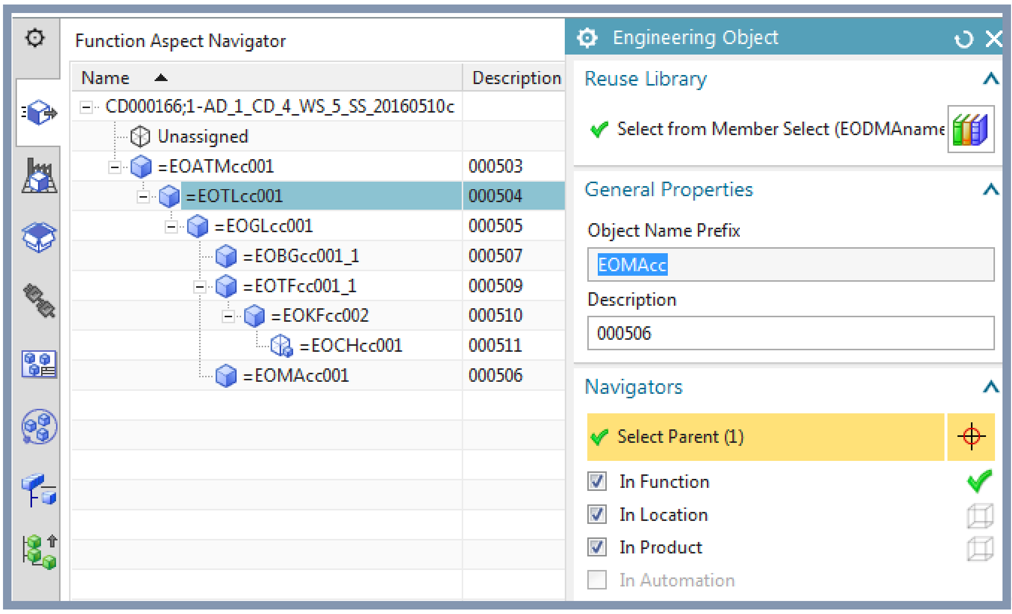
Show

● 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	⚯		EODGLname

06b_11

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



Function Aspect Navigator

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

Engineering Object

Reuse Library

✓ Select from Member Select (EODMAname

General Properties

Object Name Prefix: EOMAcc

Description: 000506

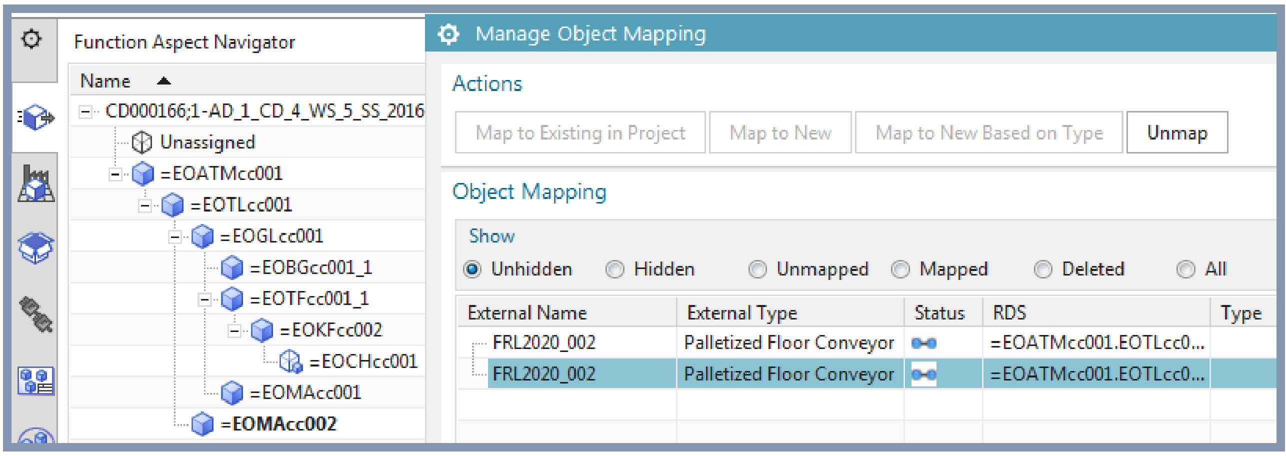
Navigators

✓ Select Parent (1)

☑ In Function ✓
☑ In Location
☑ In Product
□ In Automation

06b_12

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



Function Aspect Navigator

Name

CD000166;1-AD_1_CD_4_WS_5_SS_2016
Unassigned
=EOATMcc001
=EOTLcc001
=EOGLcc001
=EOBGcc001_1
=EOTFcc001_1
=EOKFcc002
=EOCHcc001
=EOMAcc001
=EOMAcc002

Manage Object Mapping

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

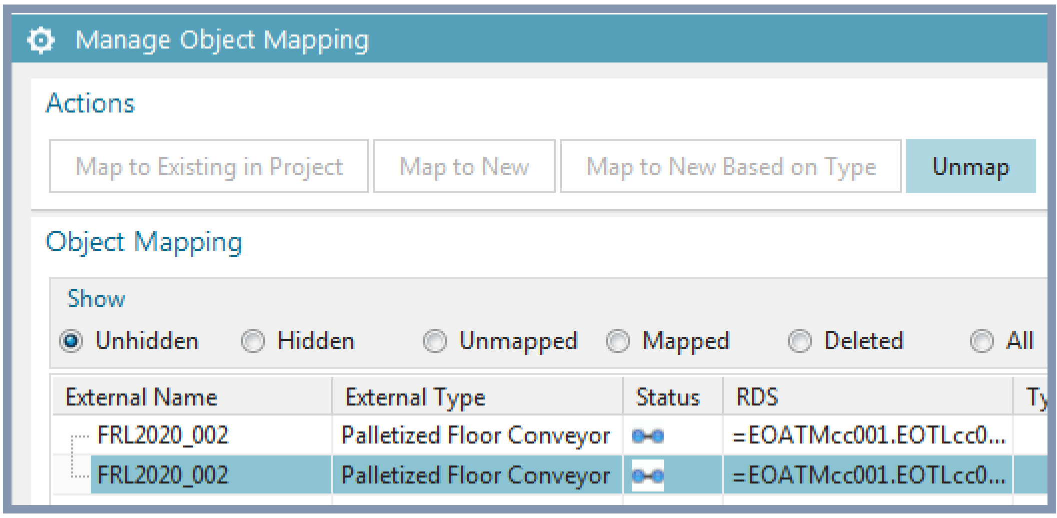
External Name	External Type	Status	RDS	Type
FRL2020_002	Palletized Floor Conveyor	⚯	=EOATMcc001.EOTLcc0...	
FRL2020_002	Palletized Floor Conveyor	⚯	=EOATMcc001.EOTLcc0...	

06b_13

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).

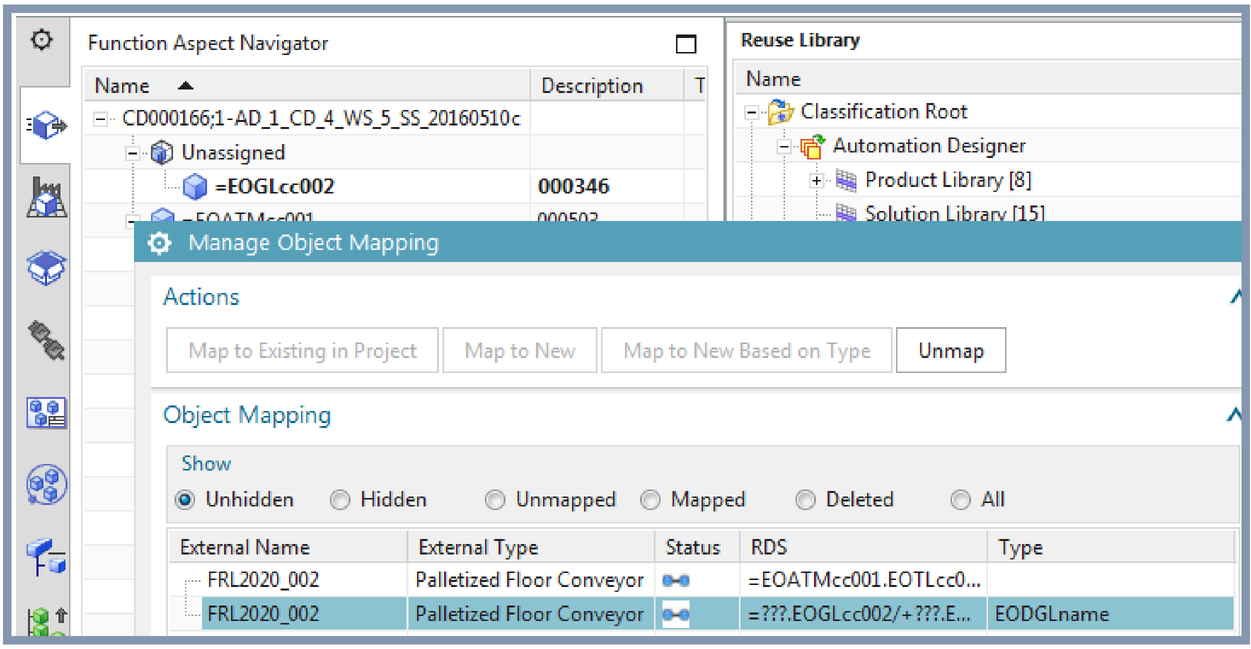


● Unhidden ○ Hidden ○ Unmapped ○ Mapped ○ Deleted ○ All

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

06b_14

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



Name	Description	T
CD000166;1-AD_1_CD_4_WS_5_SS_20160510c		
Unassigned		
=EOGLcc002	**000346**	
=EOATMcc001	000503	

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

● 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

06b_15

7. Configure a basic AD project for EPLAN

Now you will perform the most basic configuration of EPLAN and generate a report.

7.1. Import EPLAN project template

7.2. Add PM250D macro

7.3. Generate

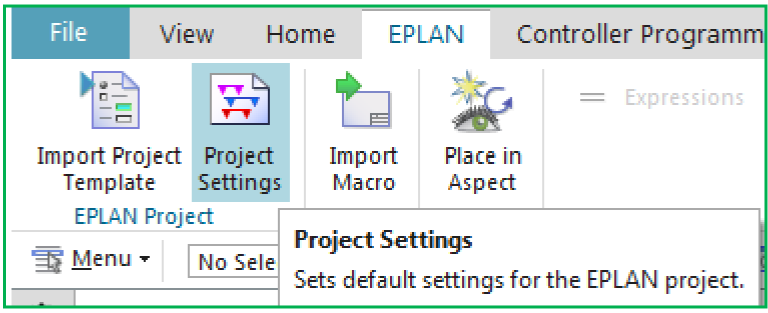
1. Sometimes need to restart the ADAgent in the SME directory (for example on my pc its in G:\20160408_SME_NX11_1612_S54_Patch1\20160403_101027_Build\automation_designer\adagent Siemens.AutomationDesigner.ADAgentUI.exe).

2. Use project template D:\EPLAN\Data\Templates\SAG\IEC_bas001.zw9

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. Click on Project Settings.

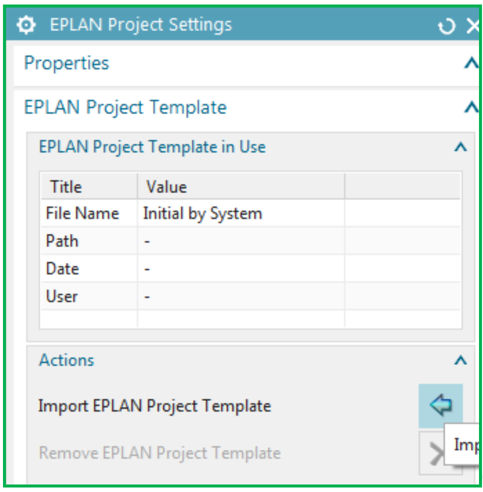


Project Settings

Sets default settings for the EPLAN project.

07b_01

2. Import the template IEC_bas001.zw9.

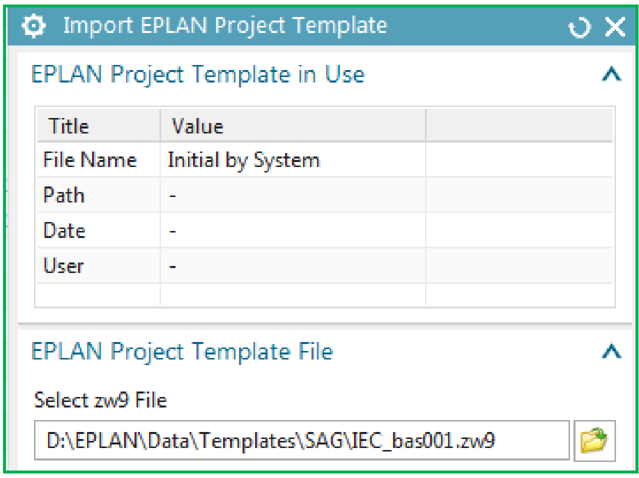


Title	Value
File Name	Initial by System
Path	-
Date	-
User	-

Import EPLAN Project Template

Remove EPLAN Project Template

07b_02


Import EPLAN Project Template
EPLAN Project Template in Use
EPLAN Project Template File

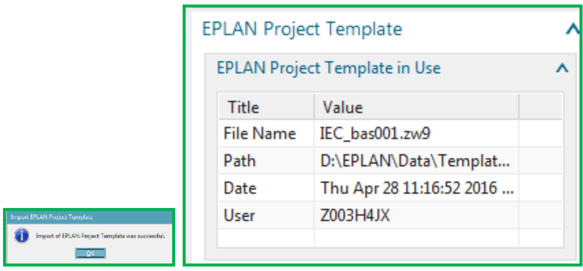
Title	Value
File Name	Initial by System
Path	-
Date	-
User	-

Select zw9 File

D:\EPLAN\Data\Templates\SAG\IEC_bas001.zw9

07b_03

3. Click OK.


Import EPLAN Project Template
Import of EPLAN Project Template was successful.
OK
EPLAN Project Template
EPLAN Project Template in Use

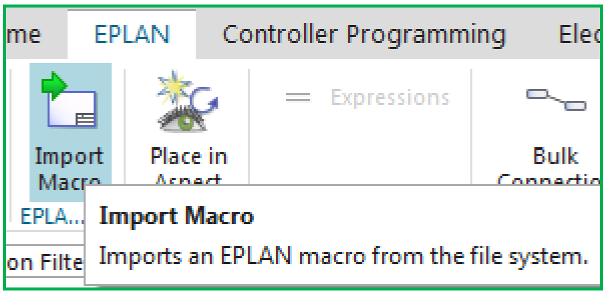
Title	Value
File Name	IEC_bas001.zw9
Path	D:\EPLAN\Data\Templat...
Date	Thu Apr 28 11:16:52 2016 ...
User	Z003H4JX

07b_04

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.
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.

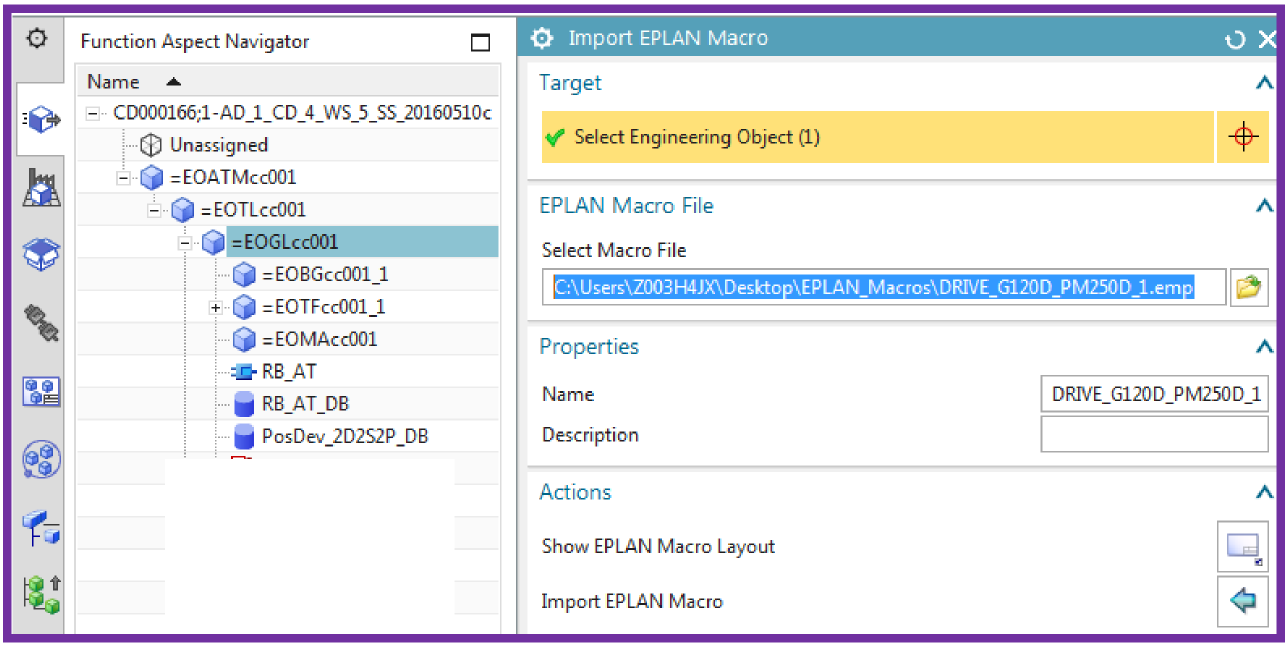
1. Click on **Import Macro**.



Import Macro
Imports an EPLAN macro from the file system.

07b_05

2. Import DRIVE_G120D_PM250D_1.emp under Engineering Object GL.



Select Engineering Object (1)

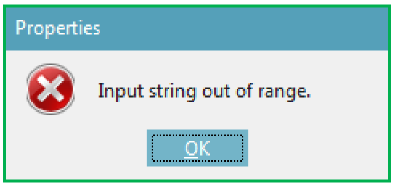
Select Macro File: C:\Users\Z003H4JX\Desktop\EPLAN_Macros\DRIVE_G120D_PM250D_1.emp

Name: DRIVE_G120D_PM250D_1

Description:

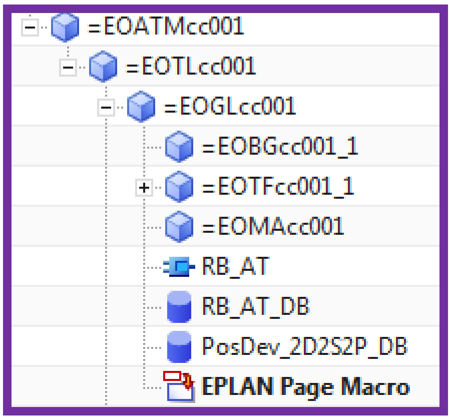
07b_06

3. Ignore this error.



Input string out of range.

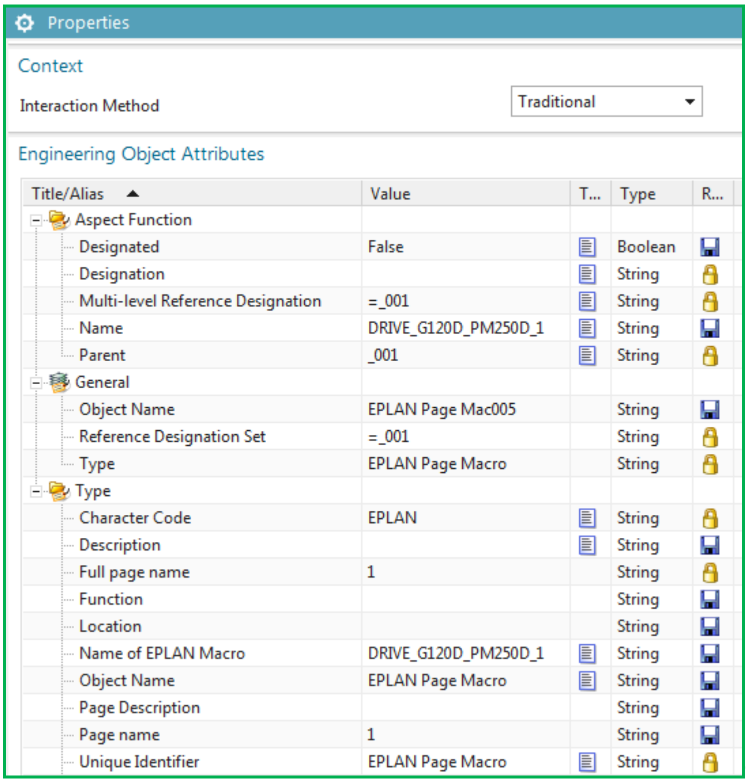
3. Click Close. The macro appears in the aspect tree.



=EOATMcc001
=EOTLcc001
=EOGLcc001
=EOBGcc001_1
=EOTFcc001_1
=EOMAcc001
RB_AT
RB_AT_DB
PosDev_2D2S2P_DB
EPLAN Page Macro

07b_07

Note the default properties.



Properties

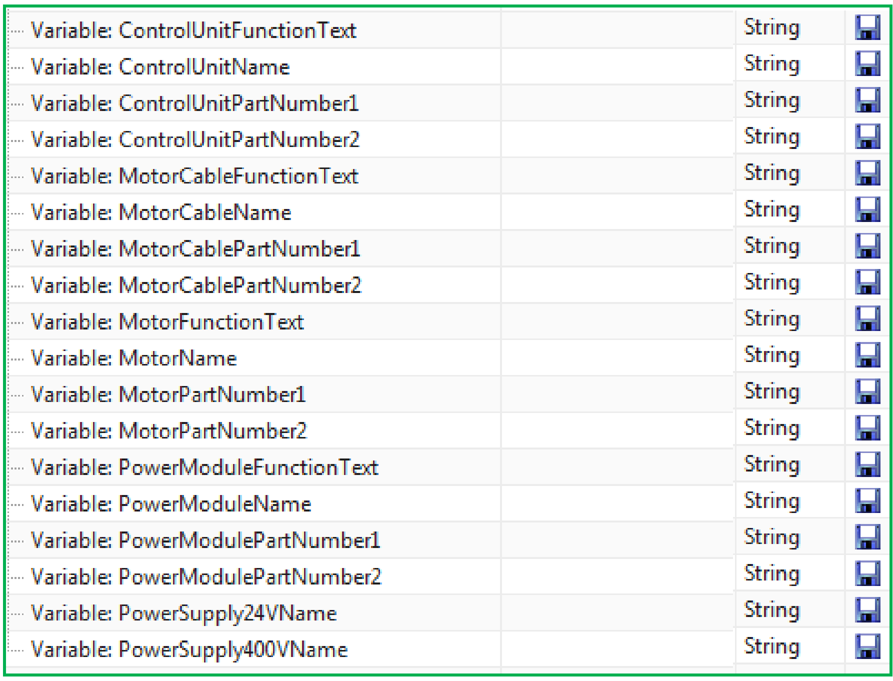
Context

Interaction Method: Traditional

Engineering Object Attributes

Title/Alias	Value	T...	Type	R...
⊟ Aspect Function				
Designated	False	📄	Boolean	💾
Designation		📄	String	🔒
Multi-level Reference Designation	=_001	📄	String	🔒
Name	DRIVE_G120D_PM250D_1	📄	String	💾
Parent	_001	📄	String	🔒
⊟ General				
Object Name	EPLAN Page Mac005		String	💾
Reference Designation Set	=_001		String	🔒
Type	EPLAN Page Macro		String	🔒
⊟ Type				
Character Code	EPLAN	📄	String	🔒
Description		📄	String	💾
Full page name	1		String	🔒
Function			String	💾
Location			String	💾
Name of EPLAN Macro	DRIVE_G120D_PM250D_1	📄	String	💾
Object Name	EPLAN Page Macro	📄	String	💾
Page Description			String	💾
Page name	1		String	💾
Unique Identifier	EPLAN Page Macro	📄	String	🔒

07b_08



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	💾

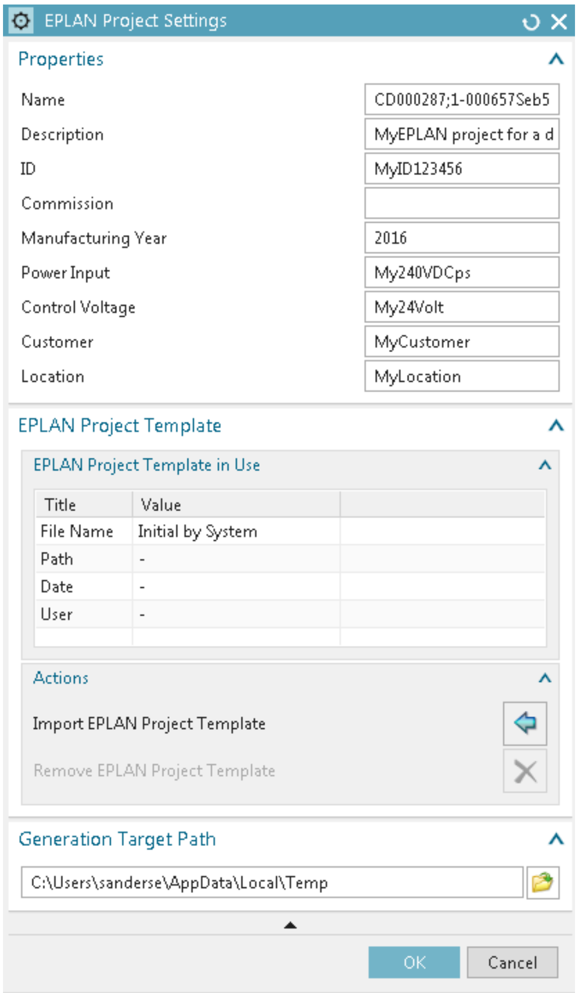
07b_09

7.3. Modify project settings

Change two values for the EPLAN template:

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

Show in EPLAN, how the values are generated on the cover page and the footer of each page.



EPLAN Project Settings

Properties

Name	CD000287;1-000657Seb5
Description	MyEPLAN project for a d
ID	MyID123456
Commission	
Manufacturing Year	2016
Power Input	My240VDCps
Control Voltage	My24Volt
Customer	MyCustomer
Location	MyLocation

EPLAN Project Template

EPLAN Project Template in Use

Title	Value
File Name	Initial by System
Path	-
Date	-
User	-

Actions

Import EPLAN Project Template

Remove EPLAN Project Template

Generation Target Path

C:\Users\sanderse\AppData\Local\Temp

7.4. Modify EPLAN macro

One change in the properties for the macro.
Let’s kick out the motor name and just focus on the data for the SINAMICS G120.

For this we need:

- AspectFunction Name: **SINAMICS G120**
- PowerModuleName: **PM240**
- ControlUnitName: **CU250S-2**

Hint: The Powermodule and the Control Unit are the most important parts of the G120 (see PDF file://debonkl0c19/adnx/Teams/Documentation/92_SebastianWork/ToTerry/SINAMICS_G120.pdf)

Assume you replaced the values as shown below (you don’t need to actually do this). Values in grey you cannot modify.

Device property new	Value new
Aspect Function	
Designated	False
Designation	
Multi-level Reference Designation	=_001
Name	**SINAMICS G120**
Parent	_001
General	
Object Name	ObjectName250
Reference Designation Set	=_001
Type	EPLAN Page Macro
Type	
Character code	EPLAN
Description	Description250
Full page name	1
Function	Function250
Location	Location250
Name of EPLAN Macro	NameOfMacro250
Object Name	TypeObjName250
Page Description	PageDescription250
Page Name	1
Unique Identifier	
ControlUnitFunctionText	ControlUnitFunctionText
ControlUnitName	**CU250S-2**
ControlUnitPartNumber1	ControlUnitPartNumber1
ControlUnitPartNumber2	ControlUnitPartNumber2
MotorCableFunctionText ??	MotorCableFunctionText
MotorCableName ??	MotorCableName
MotorCablePartNumber1	MotorCablePartNumber1
MotorCablePartNumber2	MotorCablePartNumber2
MotorFunctionText	MotorFunctionText
MotorName	MotorName
MotorPartNumber1	MotorPartNumber1
MotorPartNumber2	MotorPartNumber2

PowerModuleFunctionText	PowerModuleFunctionText
PowerModuleName	**PM240**
PowerModulePartNumber1	PowerModulePartNumber1
PowerModulePartNumber2	PowerModulePartNumber2
PowerSupply24VName	PowerSupply24VName
PowerSupply400VName	PowerSupply400VName

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.

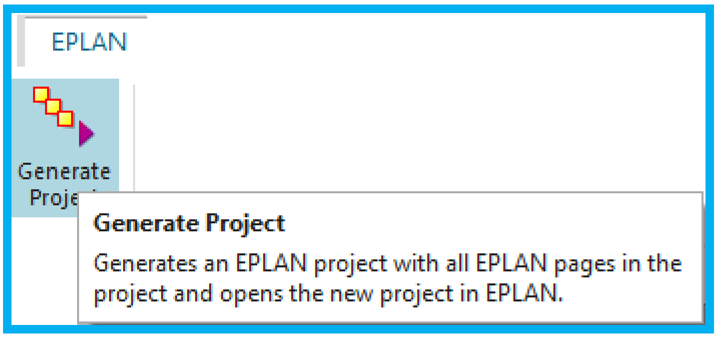
Generate an EPLAN report with

7.3.1. Default values

7.3.2. Simple text values

7.3.1. Default values

1. Click Generate Project.

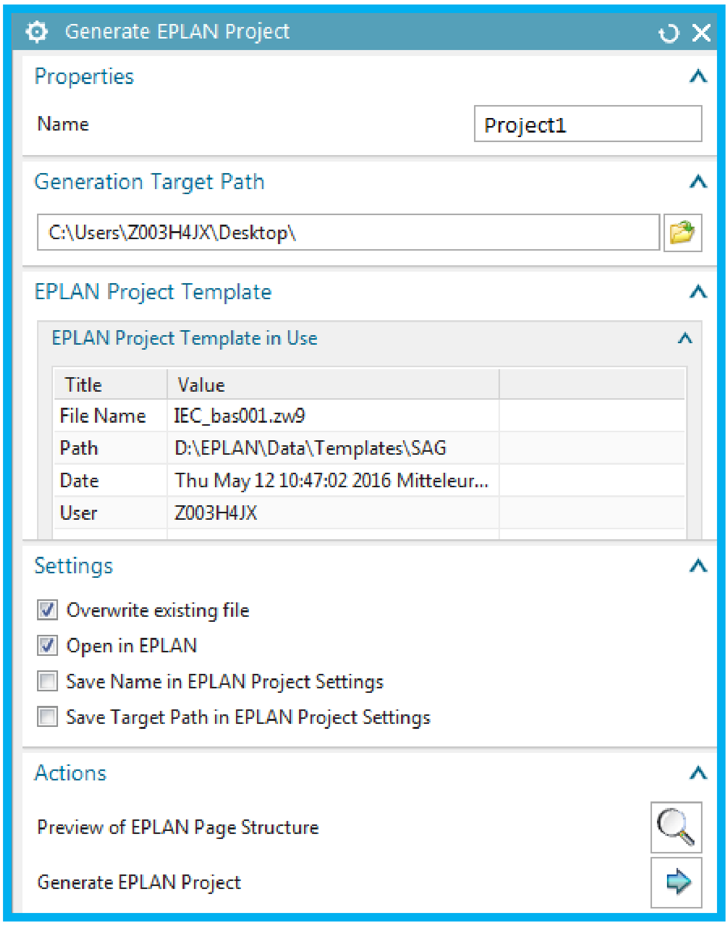


Generate Project

Generates an EPLAN project with all EPLAN pages in the project and opens the new project in EPLAN.

07b_10

2. Enter the **Name**, **Generation Target Path**, and check **Open in EPLAN**.



Generate EPLAN Project

Properties

Name: Project1

Generation Target Path

C:\Users\Z003H4JX\Desktop\

EPLAN Project Template

EPLAN Project Template in Use

Title	Value
File Name	IEC_bas001.zw9
Path	D:\EPLAN\Data\Templates\SAG
Date	Thu May 12 10:47:02 2016 Mitteleur...
User	Z003H4JX

Settings

☑ Overwrite existing file

☑ Open in EPLAN

□ Save Name in EPLAN Project Settings

□ Save Target Path in EPLAN Project Settings

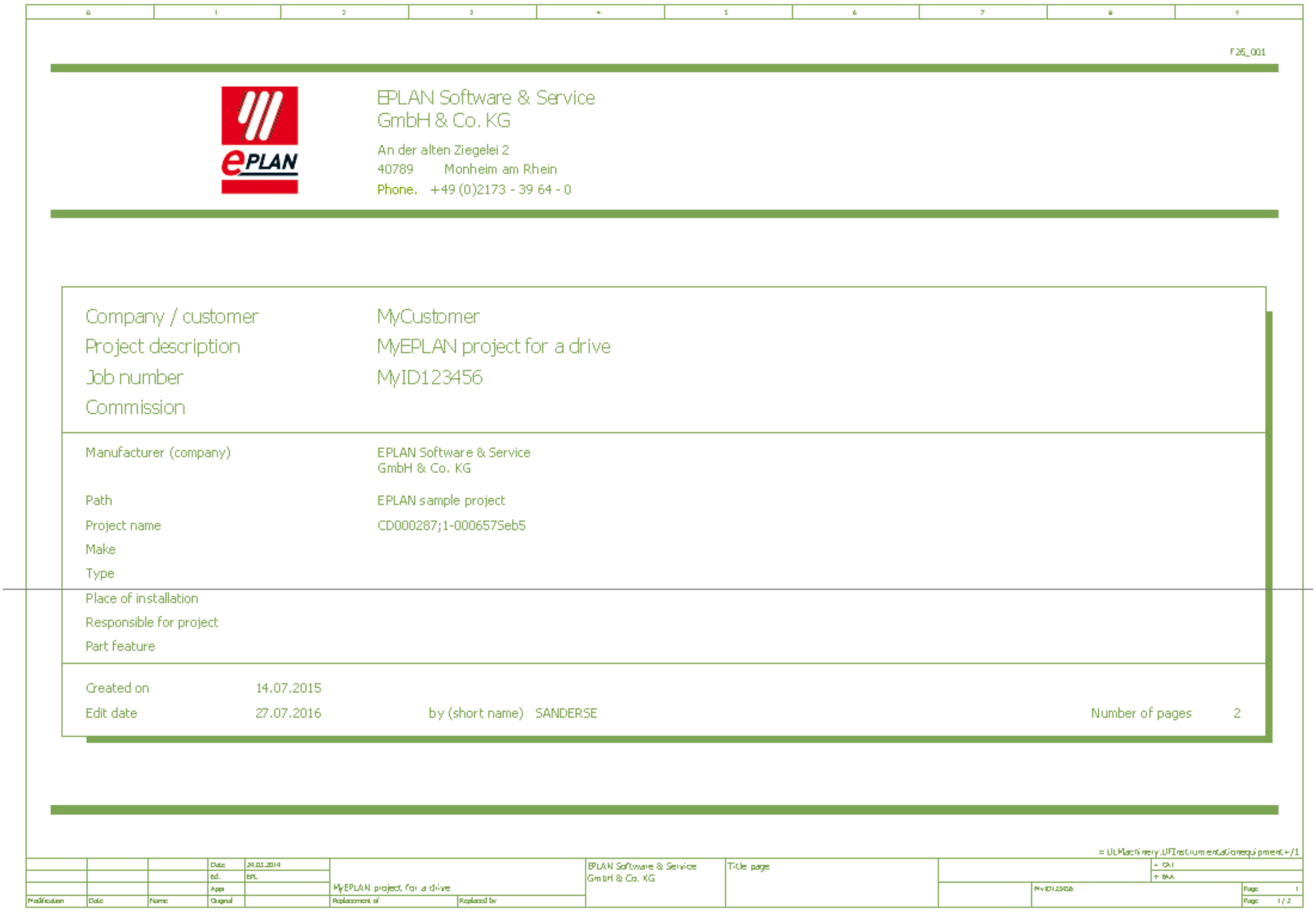
Actions

Preview of EPLAN Page Structure

Generate EPLAN Project

07b_11

3. Click **Generate**. The project is opened in EPLAN.



F26_001

EPLAN Software & Service
GmbH & Co. KG

An der alten Ziegelei 2
40789 Monheim am Rhein
Phone. +49 (0)2173 - 39 64 - 0

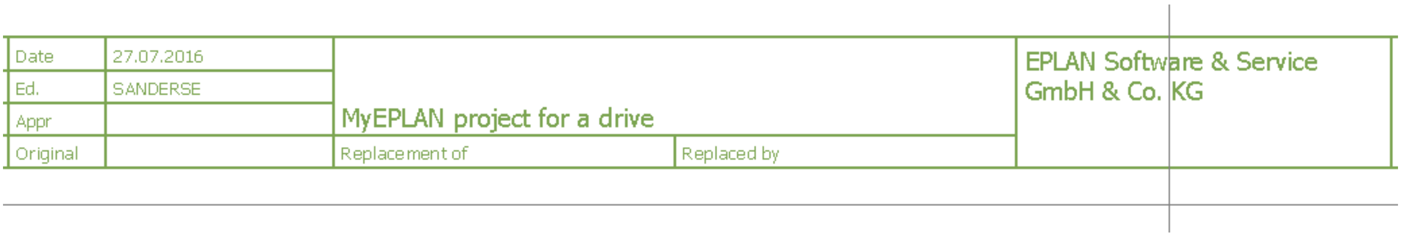
Company / customer	MyCustomer
Project description	MyEPLAN project for a drive
Job number	MyID123456
Commission	
Manufacturer (company)	EPLAN Software & Service GmbH & Co. KG
Path	EPLAN sample project
Project name	CD000287;1-0006575eb5
Make	
Type	
Place of installation	
Responsible for project	
Part feature	

Created on 14.07.2015

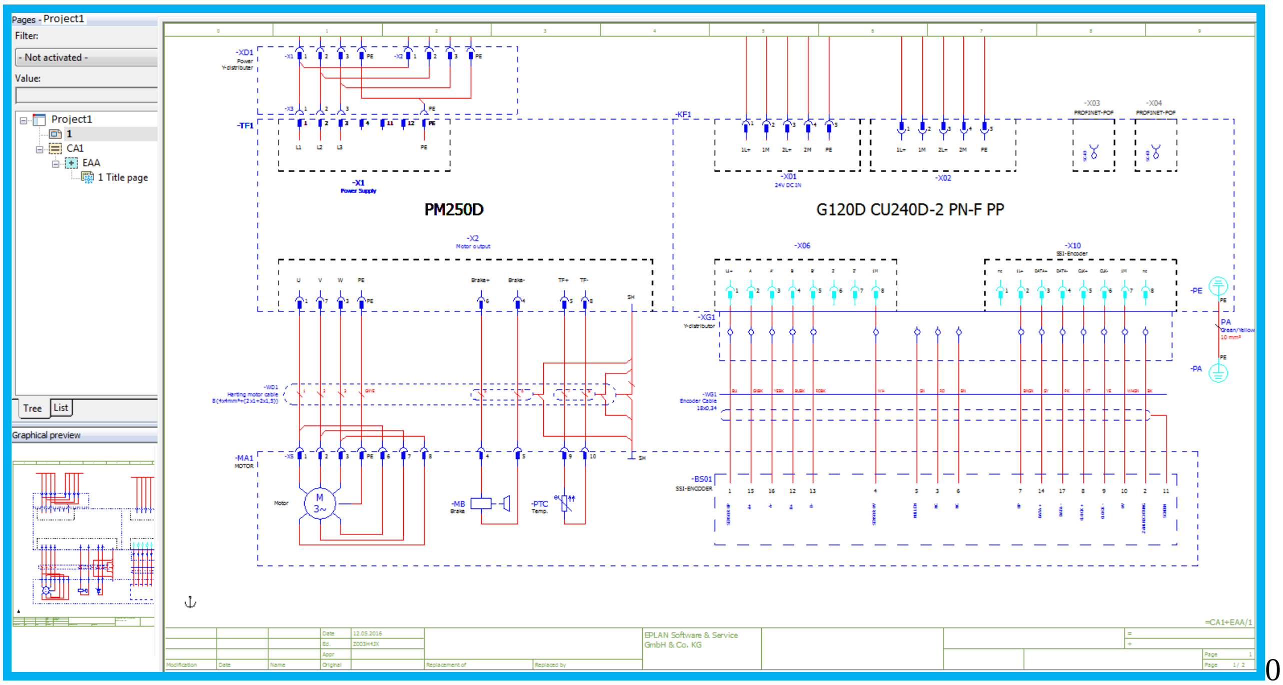
Edit date 27.07.2016 by (short name) SANDERSE Number of pages 2

= ULMachinery.UFInstrumentationequipment+/1

			Date	24.05.2014		EPLAN Software & Service GmbH & Co. KG	Title page		= CA1
			Ed.	EPL					+ EAA
			Appr		MyEPLAN project for a drive			MyID123456	Page 1
Modification	Date	Name	Original		Replacement of	Replaced by			Page 1 / 2



Date	27.07.2016			EPLAN Software & Service GmbH & Co. KG
Ed.	SANDERSE			
Appr		MyEPLAN project for a drive		
Original		Replacement of	Replaced by	



Modification	Date	Name	Date	12.05.2016
			Ed.	Z003H4JX
			Appr	
			Original	

Replacement of
Replaced by

EPLAN Software & Service
GmbH & Co. KG

Page 1
Page 1/ 2

07b_13

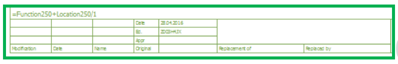
xxxxx7.3.2. Simple text values

Assume you replaced the values as shown below (you don’t need to actually do this). Values in grey you cannot modify.

Device property new	Value new
Aspect Function	
Designated	False
Name	Name250
General	
Object Name	ObjectName250
Type	
Description	Description250
Function	Function250
Location	Location250
Name of EPLAN Macro	NameOfMacro250
Object Name	TypeObjName250
Page Description	PageDescription250
Page Name	1
ControlUnitFunctionText	ControlUnitFunctionText
ControlUnitName	ControlUnitName
ControlUnitPartNumber1	ControlUnitPartNumber1
ControlUnitPartNumber2	ControlUnitPartNumber2
MotorCableFunctionText ??	MotorCableFunctionText
MotorCableName ??	MotorCableName
MotorCablePartNumber1	MotorCablePartNumber1
MotorCablePartNumber2	MotorCablePartNumber2
MotorFunctionText	MotorFunctionText
MotorName	MotorName
MotorPartNumber1	MotorPartNumber1
MotorPartNumber2	MotorPartNumber2
PowerModuleFunctionText	PowerModuleFunctionText
PowerModuleName	PowerModuleName
PowerModulePartNumber1	PowerModulePartNumber1
PowerModulePartNumber2	PowerModulePartNumber2
PowerSupply24VName	PowerSupply24VName
PowerSupply400VName	PowerSupply400VName

The diagrams below show the resulting output macro and the relationship to the variables.

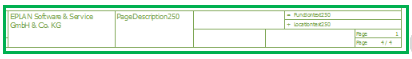
07b_14



=Function250+Location250/1

			Date	28.04.2016		
			Ed.	2003H4JX		
			Appr			
Modification	Date	Name	Original		Replacement of	Replaced by

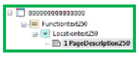
07b_15



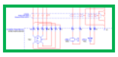
EPLAN Software & Service GmbH & Co. KG	PageDescription250		= Functiontext250 + Locationtext250	
				Page 1 Page 4 / 4

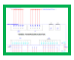
07b_16

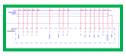
old pics:



0000000000000000
Functiontext250
Locationtext250
1 PageDescription250







8. 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.

You will now configure the Automation Designer project for non-template TIA Portal generation.

8.1. Receive hardware/software
8.2. Place FB's in aspect and create IDB's
8.3. Add/delete tags
8.4. Create TL constant value
8.5. Dynamize software
8.6. Assign software to hardware
8.7. Generate 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.

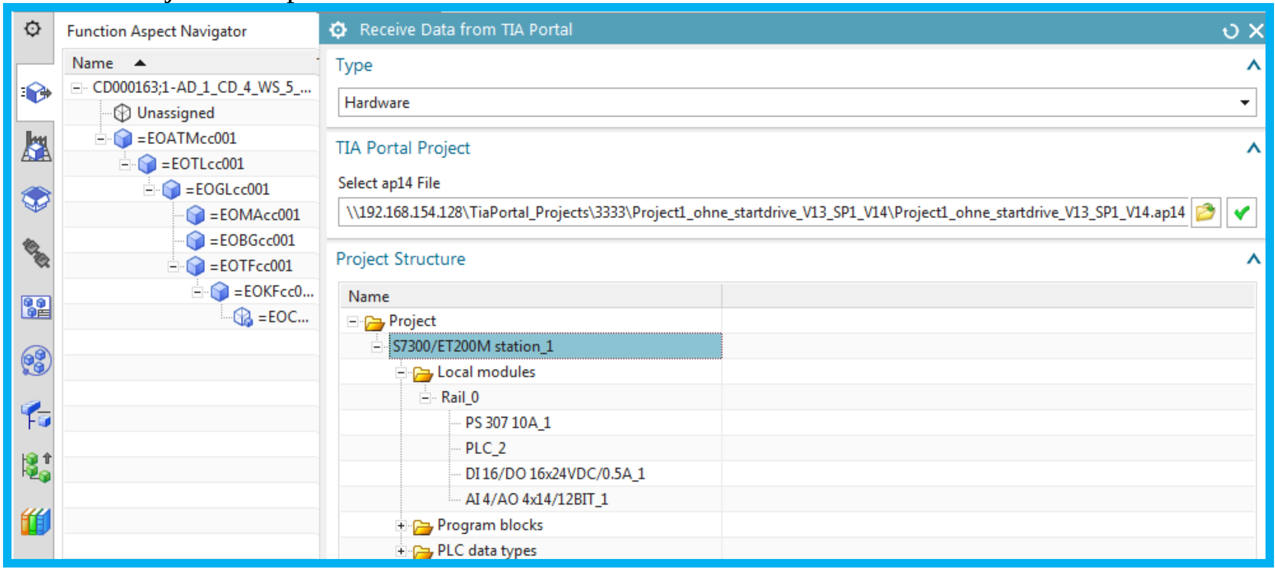
~~Note: The steps required to connect Automation Designer to TIA Portal are not described here.~~

~~8.1.1. Receive hardware~~
~~8.1.2. Import software tags (OB1, G120x, PosDev, RB_AT)~~

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 Project to import the PLC station with its modules.



Type: Hardware

TIA Portal Project

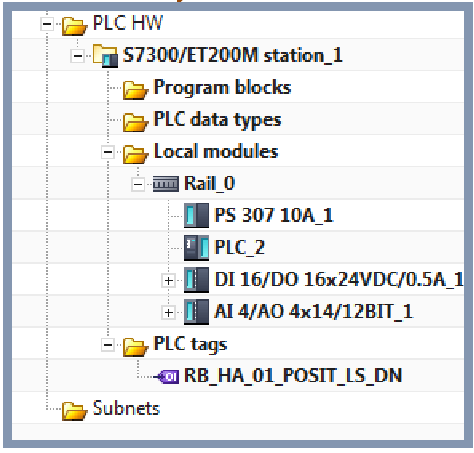
Select ap14 File: \\192.168.154.128\TiaPortal_Projects\3333\Project1_ohne_startdrive_V13_SP1_V14\Project1_ohne_startdrive_V13_SP1_V14.ap14

08b_01

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

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.

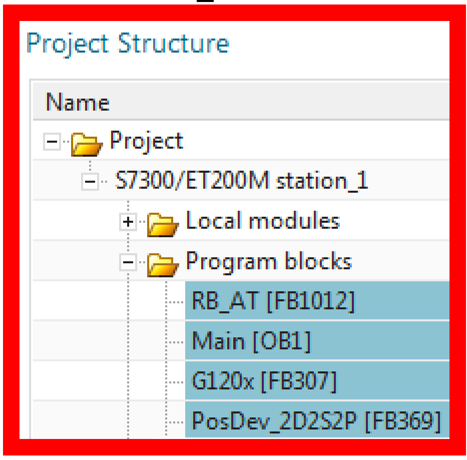


- PLC HW
  - S7300/ET200M station_1
    - Program blocks
    - PLC data types
    - Local modules
      - Rail_0
        - PS 307 10A_1
        - PLC_2
        - DI 16/DO 16x24VDC/0.5A_1
        - AI 4/AO 4x14/12BIT_1
    - PLC tags
      - RB_HA_01_POSIT_LS_DN
  - Subnets

08b_02

8.1.2. Import software

1. For **Type** select software.
2. Select the following blocks
   - Main [OB1]
   - RB_AT
   - G120x
   - PosDev_2D2S2P



Project Structure

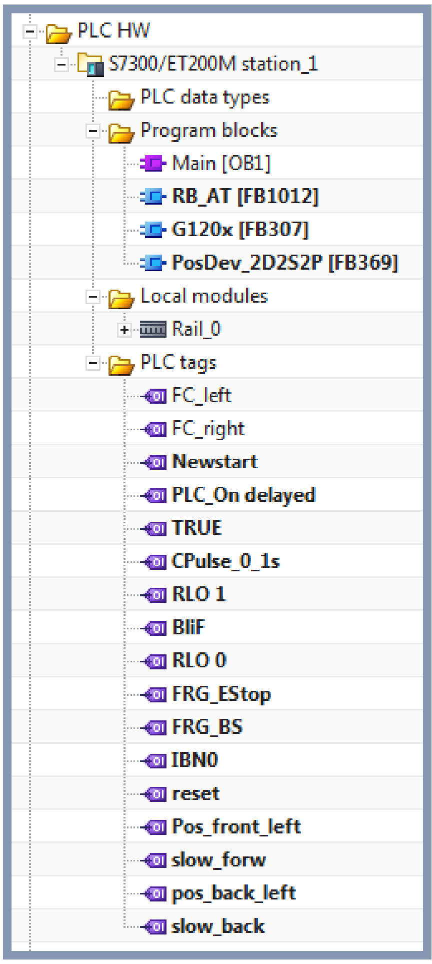
Name

- Project
  - S7300/ET200M station_1
    - Local modules
    - Program blocks
      - RB_AT [FB1012]
      - Main [OB1]
      - G120x [FB307]
      - PosDev_2D2S2P [FB369]

08c_01

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.



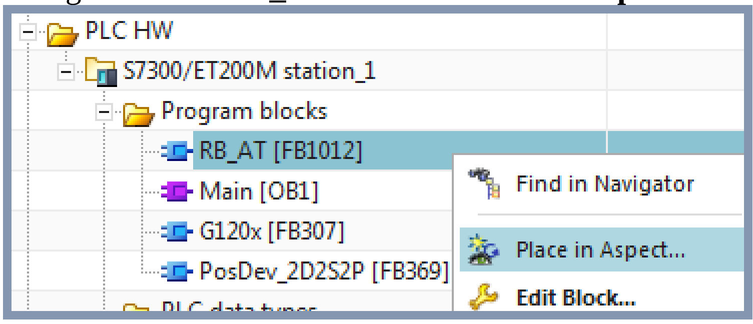
- PLC HW
  - S7300/ET200M station_1
    - PLC data types
    - Program blocks
      - Main [OB1]
      - **RB_AT [FB1012]**
      - **G120x [FB307]**
      - **PosDev_2D2S2P [FB369]**
    - Local modules
      - Rail_0
    - PLC tags
      - FC_left
      - FC_right
      - **Newstart**
      - **PLC_On delayed**
      - **TRUE**
      - **CPulse_0_1s**
      - **RLO 1**
      - **BliF**
      - **RLO 0**
      - **FRG_EStop**
      - **FRG_BS**
      - **IBN0**
      - **reset**
      - **Pos_front_left**
      - **slow_forw**
      - **pos_back_left**
      - **slow_back**

08b_03

8.2. Place the function blocks in aspects and create IDBs

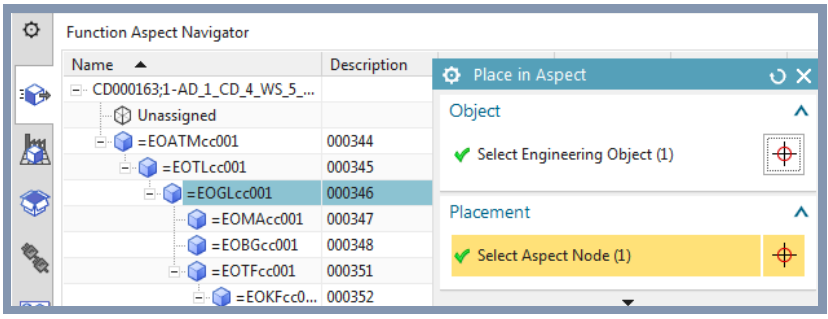
You now need to place the function blocks in the aspects and create IDBs in the aspects. Where you place the function blocks determines the aspect chain that will be used to create unique indentifiers (symbolic names). In this Getting Started you focus only on the Function aspect.

1. Right-click on RB_AT and select **Place in Aspect**.



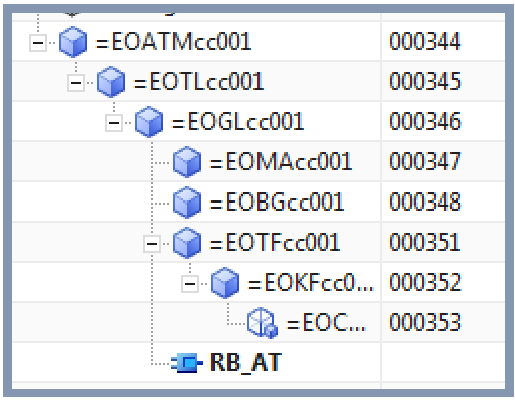
08b_04

2. For Placement select GL.



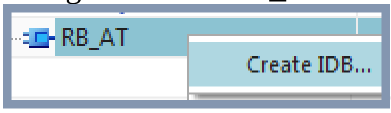
08b_05

3. Click **OK**.



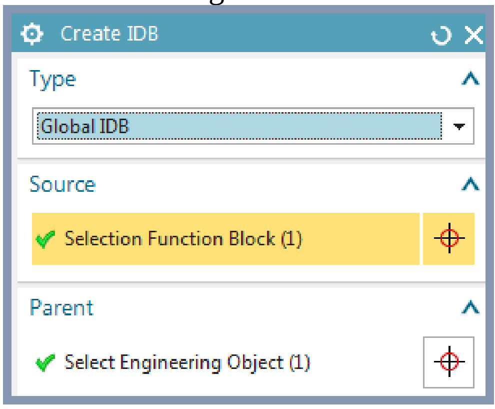
08b_06

4. Right-click on RB_AT and select **Create IDB**.



08b_07

5. Default settings are correct. Click **OK**.



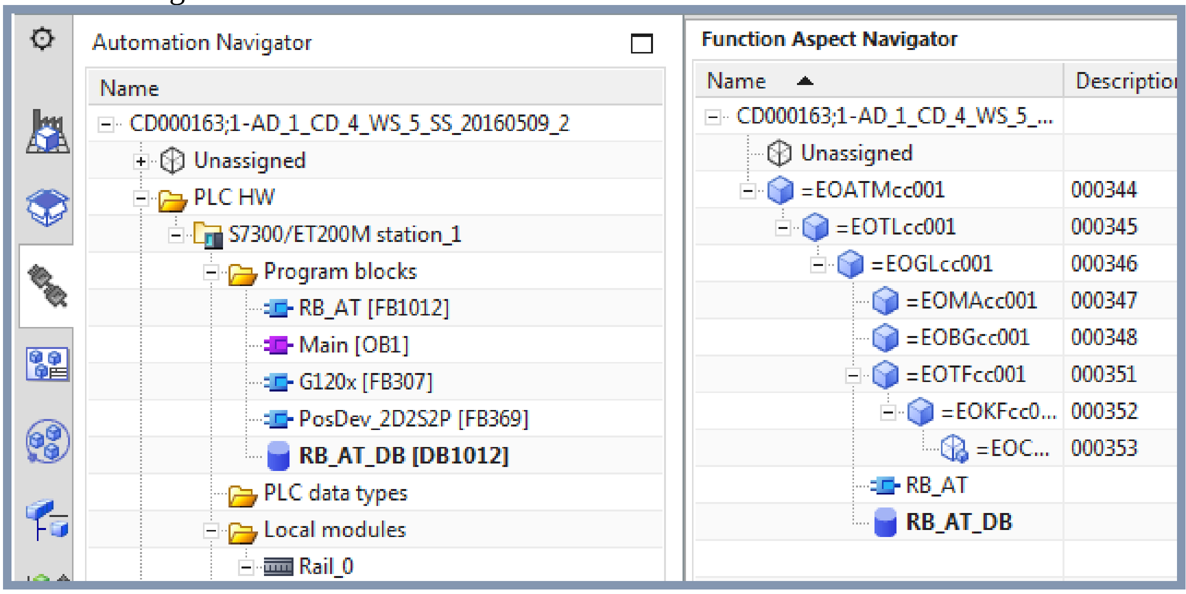
Type
Global IDB

Source
Selection Function Block (1)

Parent
Select Engineering Object (1)

08b_08

The following is the result.



Automation Navigator

```
Name
- 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
```

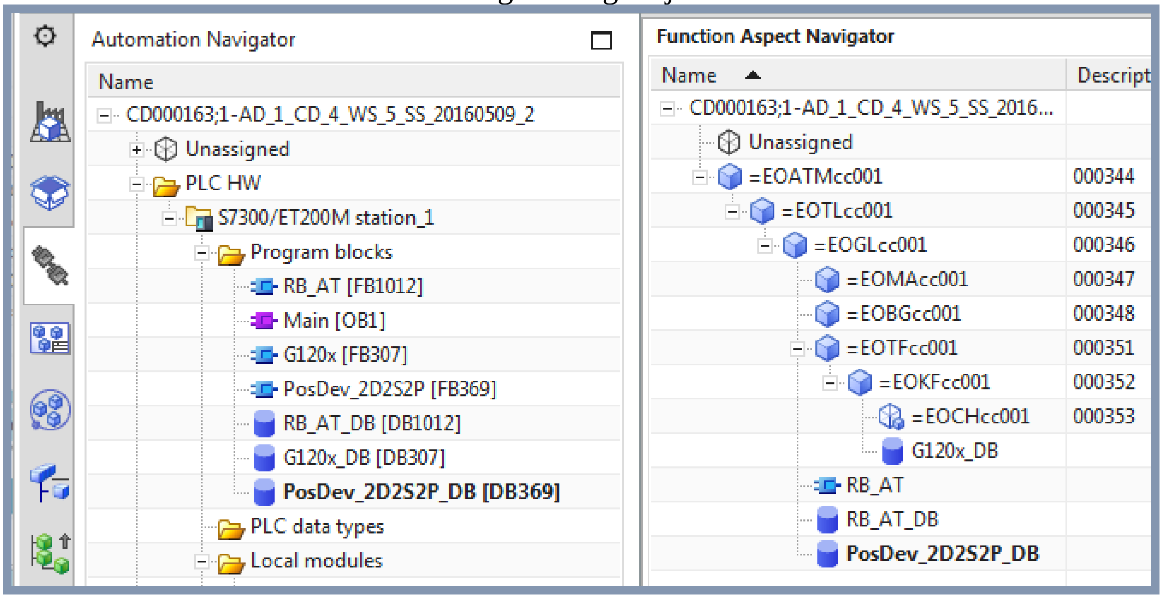
Function Aspect Navigator

Name	Description
CD000163;1-AD_1_CD_4_WS_5_...	
Unassigned	
=EOATMcc001	000344
=EOTLcc001	000345
=EOGLcc001	000346
=EOMAcc001	000347
=EOBGcc001	000348
=EOTFcc001	000351
=EOKFcc0...	000352
=EOC...	000353
RB_AT	
RB_AT_DB	

08b_09

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

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



Automation Navigator

```
Name
- 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]
                G120x_DB [DB307]
                PosDev_2D2S2P_DB [DB369]
              PLC data types
            - Local modules
```

Function Aspect Navigator

Name	Descript
CD000163;1-AD_1_CD_4_WS_5_SS_2016...	
Unassigned	
=EOATMcc001	000344
=EOTLcc001	000345
=EOGLcc001	000346
=EOMAcc001	000347
=EOBGcc001	000348
=EOTFcc001	000351
=EOKFcc001	000352
=EOCHcc001	000353
G120x_DB	
RB_AT	
RB_AT_DB	
PosDev_2D2S2P_DB	

08b_10

8.3. Add tags

For this Getting Started we only add 2 tags.

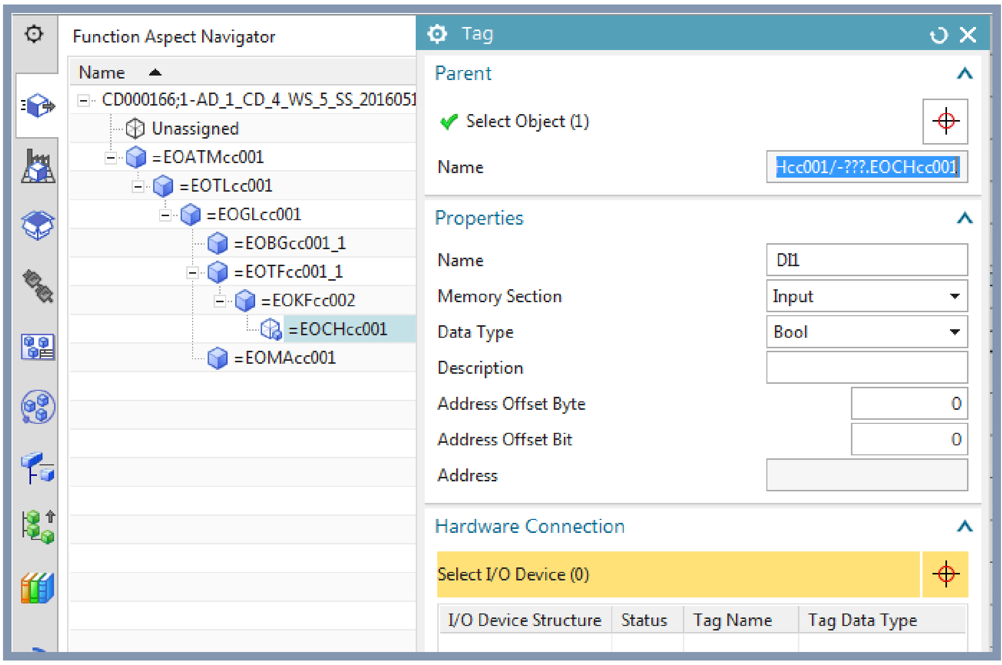
8.2.1. DI1
8.2.2. PID0

8.3.1. DI1

When you copy **CH001**, you also copy its **DI** tag. The copied tag has the same expressions as the original tag. Automation Designer evaluates the expressions at the copies and the copied tags automatically use the description and Multi Reference Designation of their connected sensor. TERRY: NO it doesn't.

Add the sensor tag DI1 (boolean input).

1. Click **Tag**.
2. Specify the tag properties:
   - Parent Engineering Object = CH
   - Name = DI1
   - Memory section = Input
   - Data Type = Boolean



Function Aspect Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_SS_2016051
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
          - =EOKFcc002
            - =EOCHcc001
        - =EOMAcc001

Tag

Parent

Select Object (1)

Name	Hcc001/-???.EOCHcc001

Properties

Name	DI1
Memory Section	Input
Data Type	Bool
Description	
Address Offset Byte	0
Address Offset Bit	0
Address	

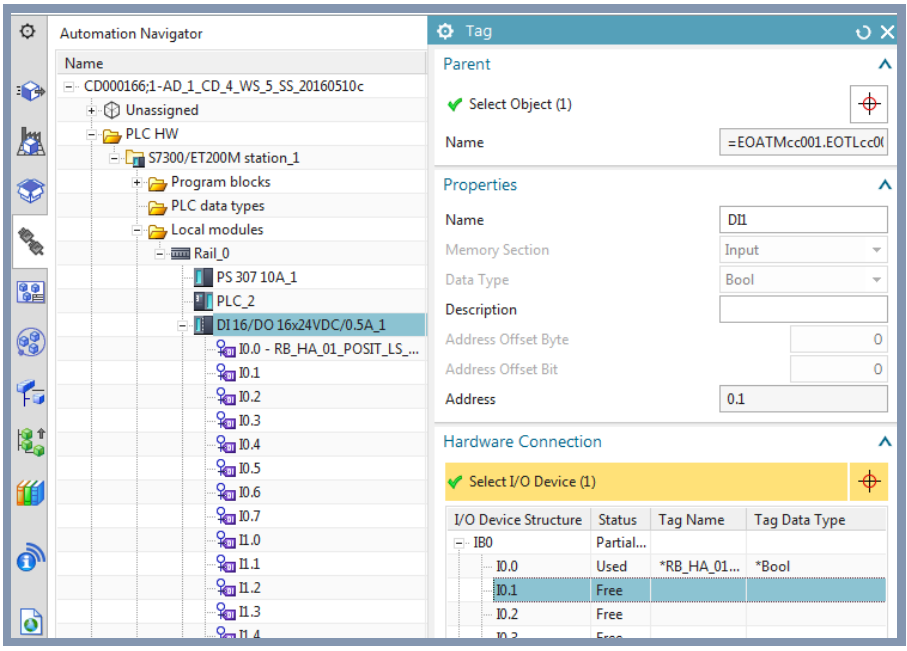
Hardware Connection

Select I/O Device (0)

I/O Device Structure	Status	Tag Name	Tag Data Type

08b_11

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.



Automation Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_SS_20160510c
  - Unassigned
  - PLC HW
    - S7300/ET200M station_1
      - Program blocks
      - PLC data types
      - Local modules
        - Rail_0
          - PS 307 10A_1
          - PLC_2
          - DI 16/DO 16x24VDC/0.5A_1
            - I0.0 - RB_HA_01_POSIT_LS_...
            - I0.1
            - I0.2
            - I0.3
            - I0.4
            - I0.5
            - I0.6
            - I0.7
            - I1.0
            - I1.1
            - I1.2
            - I1.3
            - I1.4

Tag

Parent

✓ Select Object (1)

Name =EOATMcc001.EOTLcc0(

Properties

Name DI1

Memory Section Input

Data Type Bool

Description

Address Offset Byte 0

Address Offset Bit 0

Address 0.1

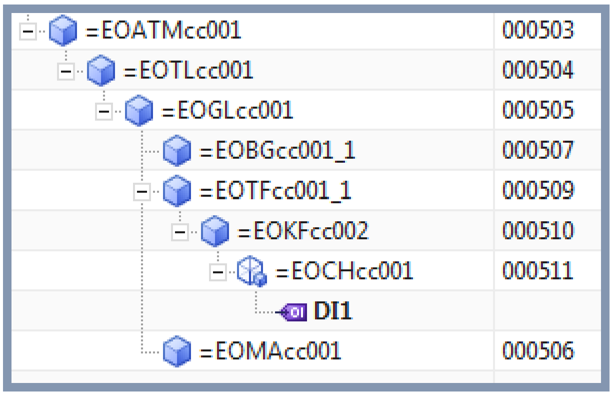
Hardware Connection

✓ Select I/O Device (1)

I/O Device Structure	Status	Tag Name	Tag Data Type
IB0	Partial...		
I0.0	Used	*RB_HA_01...	*Bool
I0.1	Free		
I0.2	Free		
I0.3	Free		

08b_12

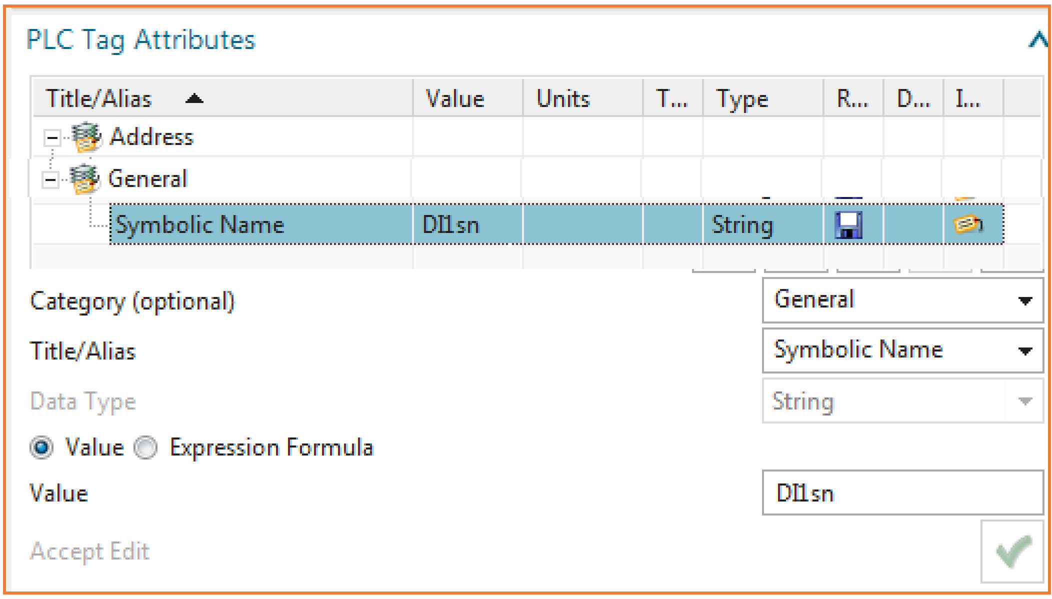
6. Click **OK**.



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

08b_13

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)	General
Title/Alias	Symbolic Name
Data Type	String
● Value ○ Expression Formula	
Value	DI1sn
Accept Edit	✓

08b_14

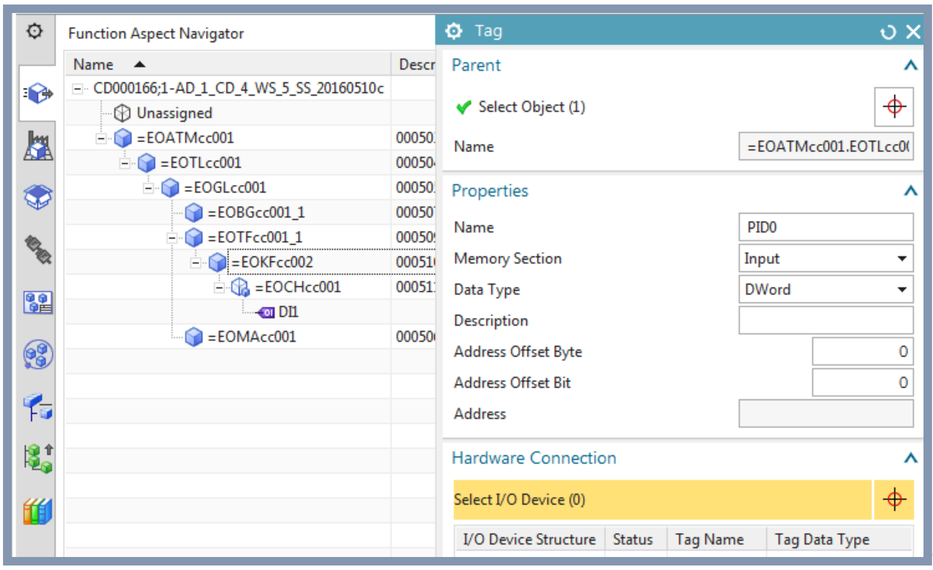
8.3.2. PID0

Add the tag PID0 (digital input).

1. Click **Tag**.

2. Specify the tag properties:

- Parent Engineering Object = KF
- Name = PID0
- Memory section = Input
- Data Type = DWord



Name: =EOATMcc001.EOTLcc0(

Name: PID0

Memory Section: Input

Data Type: DWord

Description:

Address Offset Byte: 0

Address Offset Bit: 0

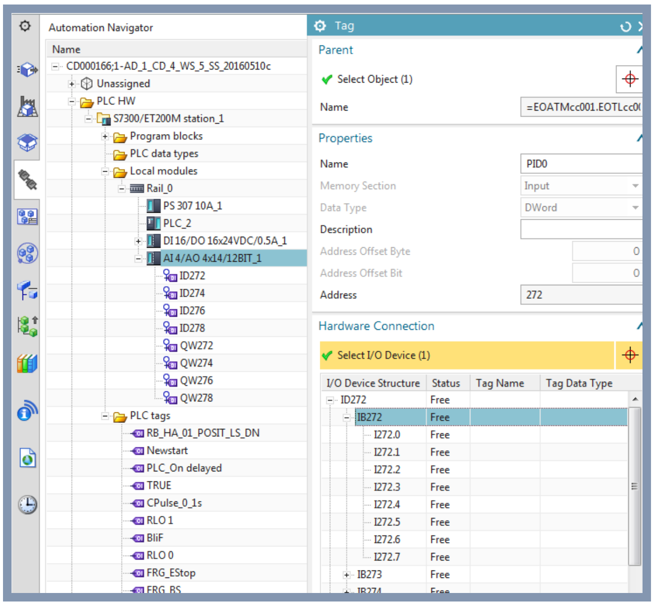
Address:

08b_15

3. Click on **Select I/O device**.

4. Click on the local module **AI 4 / AO 4x14**.

5. Select a free IO.



Name: =EOATMcc001.EOTLcc0(

Name: PID0

Memory Section: Input

Data Type: DWord

Description:

Address Offset Byte: 0

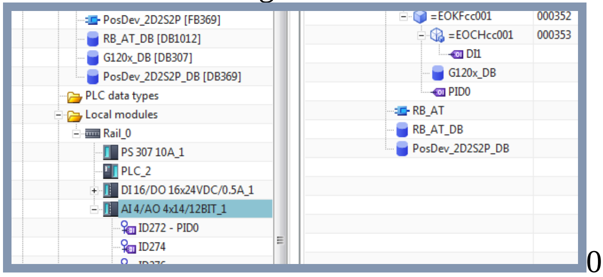
Address Offset Bit: 0

Address: 272

I/O Device Structure	Status	Tag Name	Tag Data Type
ID272	Free		
IB272	Free		
I272.0	Free		
I272.1	Free		
I272.2	Free		
I272.3	Free		
I272.4	Free		
I272.5	Free		
I272.6	Free		
I272.7	Free		
IB273	Free		
IB274	Free		

08b_16

6. Click **OK**. The tag is added.

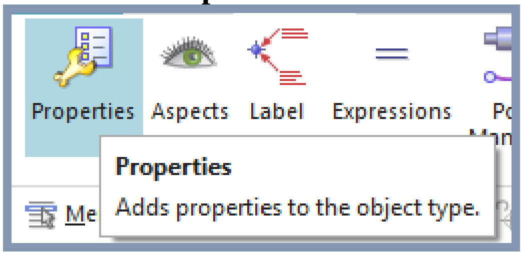


08b_17

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**.



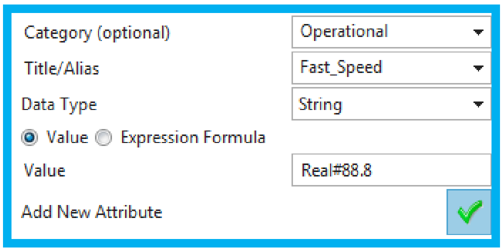
Properties

Adds properties to the object type.

08b_18

3. Create a property with following :

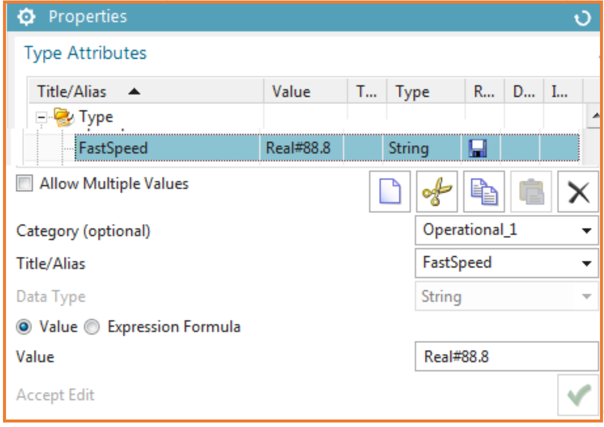
- Category = Operational_1
- Title/Alias = Fast_Speed
- Data Type = String
- Value = Real#88.8



Category (optional)	Operational
Title/Alias	Fast_Speed
Data Type	String
● Value ○ Expression Formula	
Value	Real#88.8
Add New Attribute	✓

08b_19

4. Click the green arrow. Note that the category is not shown.



Properties

Type Attributes

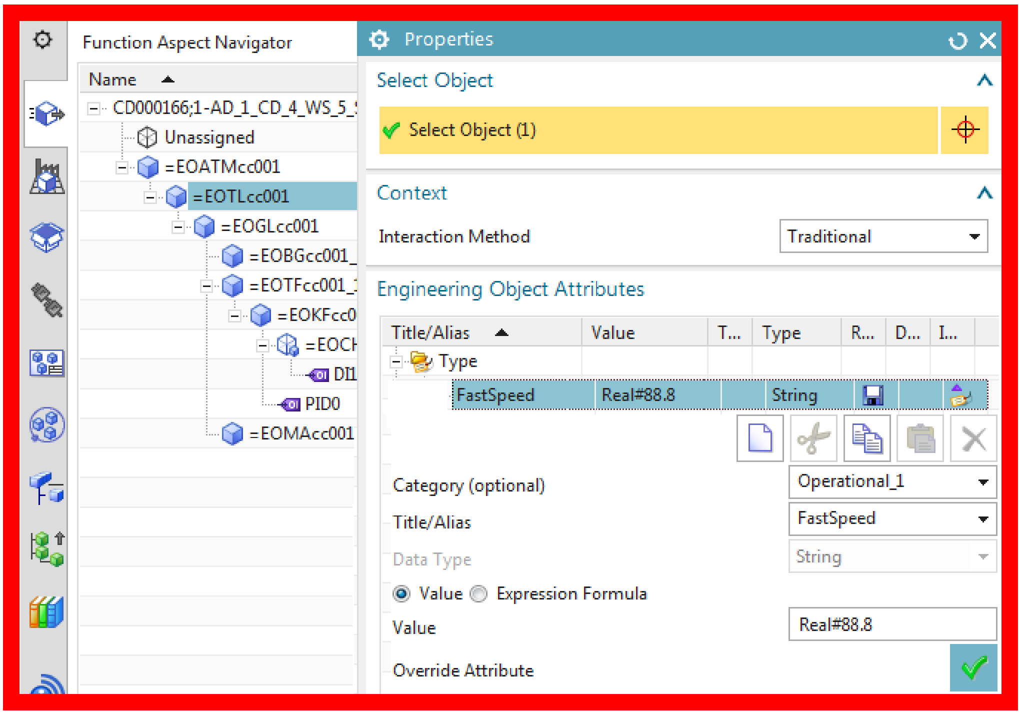
Title/Alias	Value	T...	Type	R...	D...	I...
Type						
FastSpeed	Real#88.8		String			

□ Allow Multiple Values

Category (optional)	Operational_1
Title/Alias	FastSpeed
Data Type	String
● Value ○ Expression Formula	
Value	Real#88.8
Accept Edit	✓

08b_20

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



Function Aspect Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_
        - =EOTFcc001_
          - =EOKFcc0
            - =EOCH
              - DI1
          - PID0
        - =EOMAcc001

Properties

Select Object

✓ Select Object (1)

Context

Interaction Method: Traditional

Engineering Object Attributes

Title/Alias	Value	T...	Type	R...	D...	I...
Type						
FastSpeed	Real#88.8		String			

Category (optional): Operational_1

Title/Alias: FastSpeed

Data Type: String

● Value ○ Expression Formula

Value: Real#88.8

Override Attribute ✓

08c_02

NOTE: You can edit the Engineering Object in the aspects. And shows the CATEGORY (FFFF)

8.5. Dynamize 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.

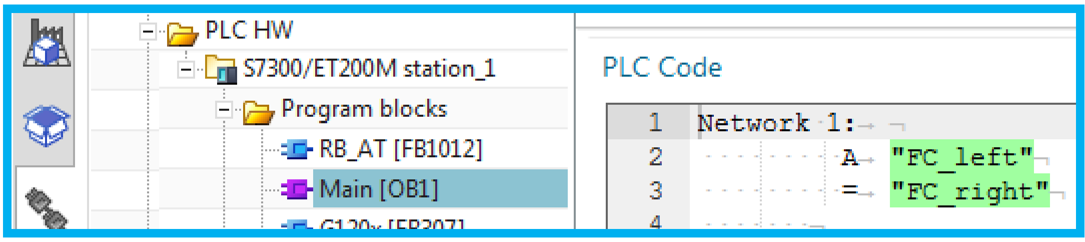
In this section dynamize the calls (OB1, RB_AT) and tag references (RB_AT, PosDev) in the imported software blocks.

8.5.1. RB_AT manual OB1 replace by call
8.5.2. RB_AT manual connect to RB_AT automation tags
8.5.3. RB_AT manual connect to CH DI tags
8.5.4. RB_AT->PosDev replace by call
8.5.5. RB_AT->G120x replace by call
8.5.6. PosDev manual connecto to AUTOMATION tags

8.5.1. OB1->RB_AT_DB replace by call

OB1 calls the RB_AT IDB.

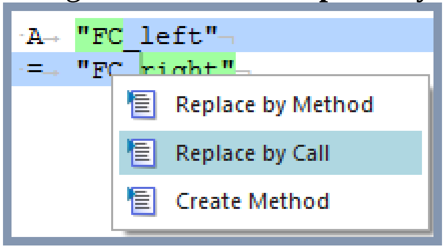
1. Double-click on **OB1**.



```
Network 1:
    A  "FC_left"
    =  "FC_right"
```

08b_22

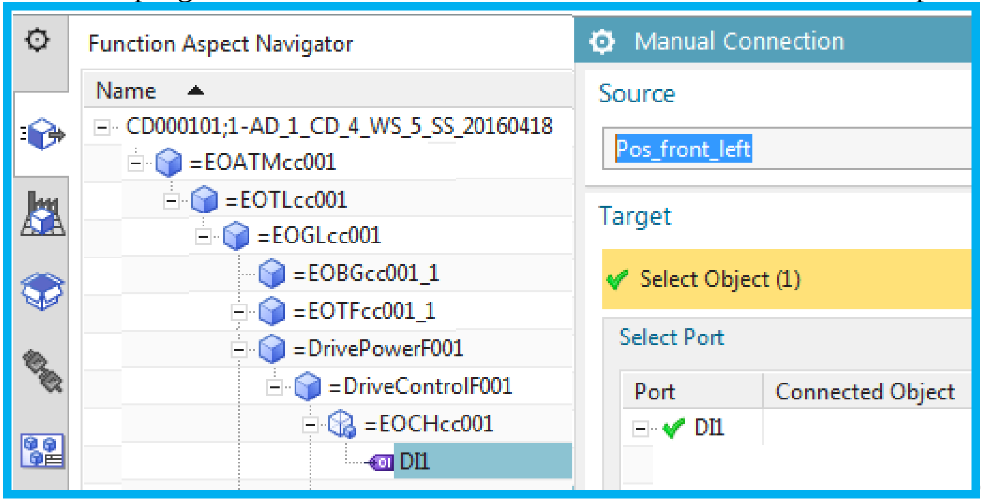
2. Select the lines of OB1 code.
3. Right-click. Select **Replace by Call**.



```
A  "FC_left"
=  "FC_right"
```

08b_23

4. For **Selection** select **Object selection**.
5. For the program block select the RB_AT_DB IDB in the Function aspect.



Source: Pos_front_left

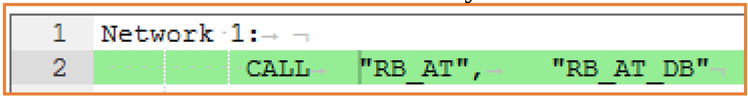
Target: ✓ Select Object (1)

Select Port

Port	Connected Object
✓ DI1	

08b_24

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



```
Network 1:
    CALL  "RB_AT",  "RB_AT_DB"
```

08b_25

You can now view the ports that were created. *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 conected 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.

You can use ports to do the following:

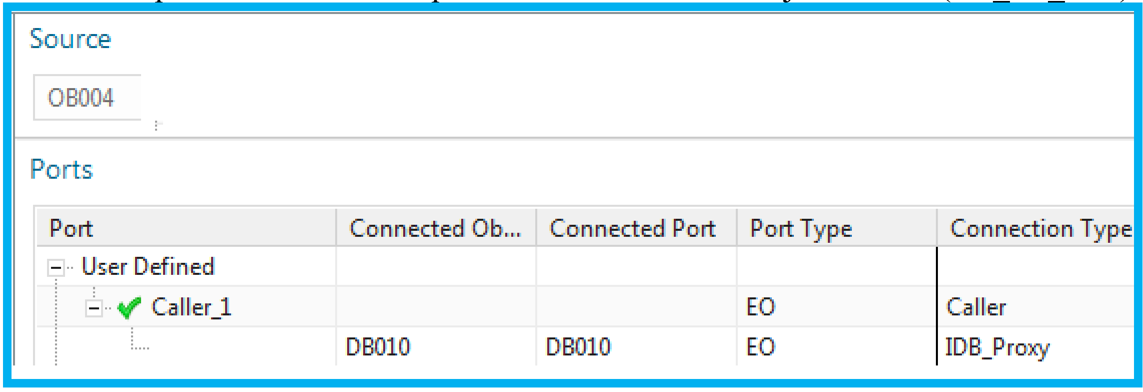
- Define data flow, material flow, or flow of electrical current between objects. Flow ports specify which items flow between the Engineering Object and its environment, and in which direction.
- Plan subnets.
- Define plugs and sockets on EPLAN macros.
- Define a logical connection between two objects 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

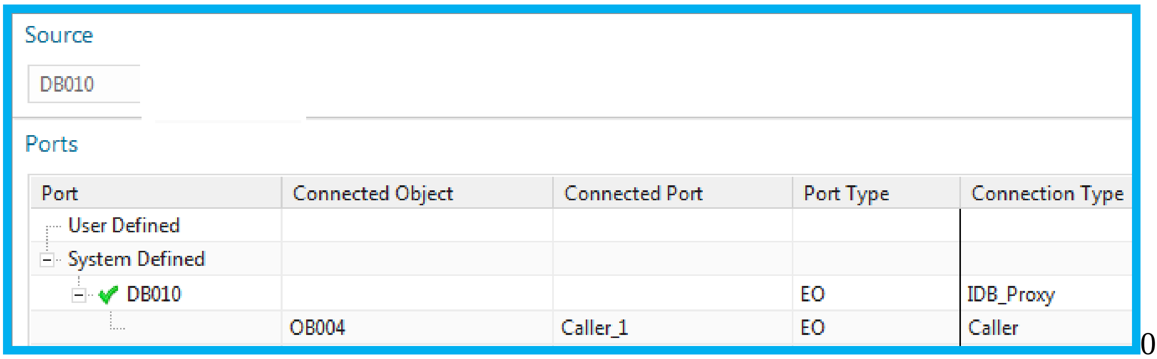
Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type
User Defined				
✓ Caller_1			EO	Caller
	DB010	DB010	EO	IDB_Proxy

08b_26

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

DB010

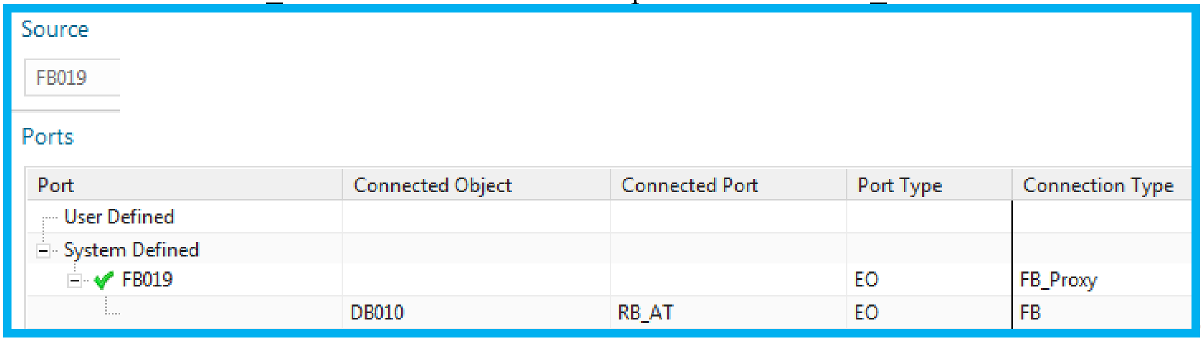
Ports

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

08b_27

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

FB019

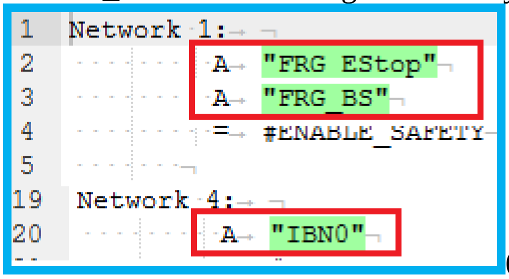
Ports

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

08b_28

8.5.2. RB_AT manual connect to RB_AT AUTOMATION tags (FRG_EStop)

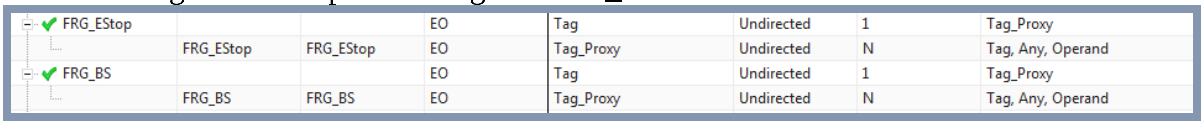
The RB_AT automation tags are already connected.



```
Network 1:
        A  "FRG_EStop"
        A  "FRG_BS"
        =  #ENABLE_SAFETY

Network 4:
        A  "IBN0"
```

08b_29

The following shows the ports manager for RB_AT.

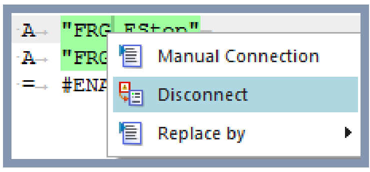


✓ FRG_EStop			EO	Tag	Undirected	1	Tag_Proxy
	FRG_EStop	FRG_EStop	EO	Tag_Proxy	Undirected	N	Tag, Any, Operand
✓ FRG_BS			EO	Tag	Undirected	1	Tag_Proxy
	FRG_BS	FRG_BS	EO	Tag_Proxy	Undirected	N	Tag, Any, Operand

08b_30

The following shows you how to make this connection.

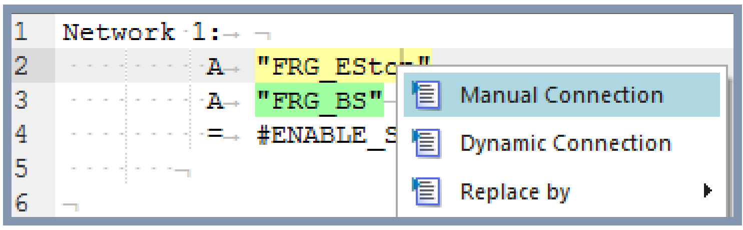
1. Disconnect.



```
A  "FRG_EStop"
A  "FRG
=  #ENA
```

08b_31

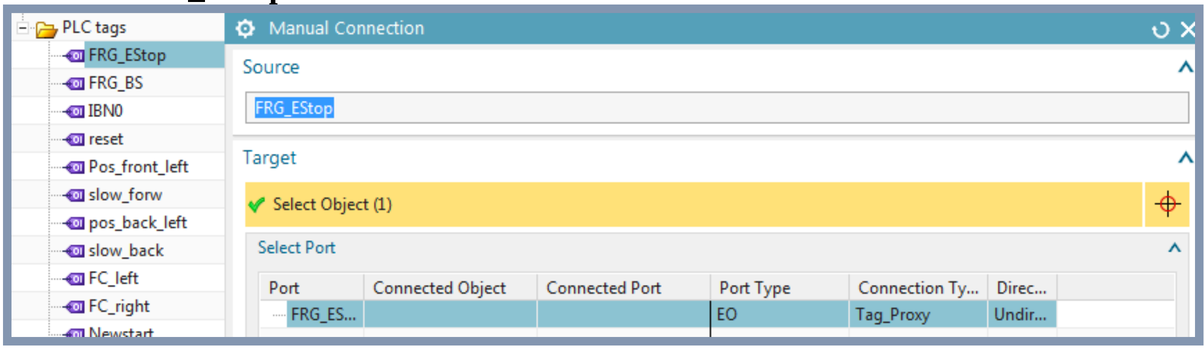
2. Select **Manual Connection**.



```
Network 1:
        A  "FRG_EStop"
        A  "FRG_BS"
        =  #ENABLE_S
```

08b_32

3. Select **FRG_EStop**.



Source

FRG_EStop

Target

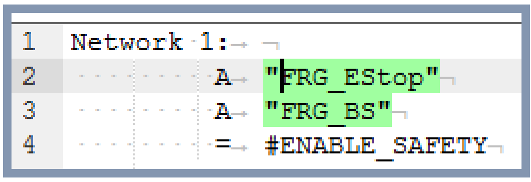
✓ Select Object (1)

Select Port

Port	Connected Object	Connected Port	Port Type	Connection Ty...	Direc...
FRG_ES...			EO	Tag_Proxy	Undir...

08b_33

4. Click **OK**.



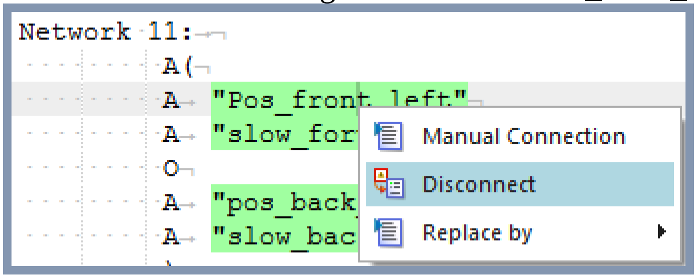
```
Network 1:
        A  "FRG_EStop"
        A  "FRG_BS"
        =  #ENABLE_SAFETY
```

08b_34

8.5.3. RB_AT manual connect to CH DI tags

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

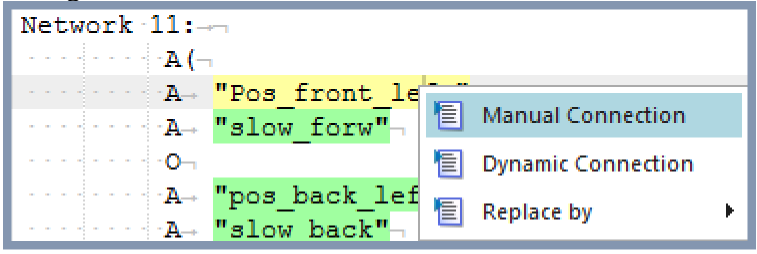
1. Disconnect the existing connection for **Pos_front_left**.



```
Network 11:
    A(
    A "Pos_front_left"
    A "slow_for
    O
    A "pos_back
    A "slow_bac
```

08b_35

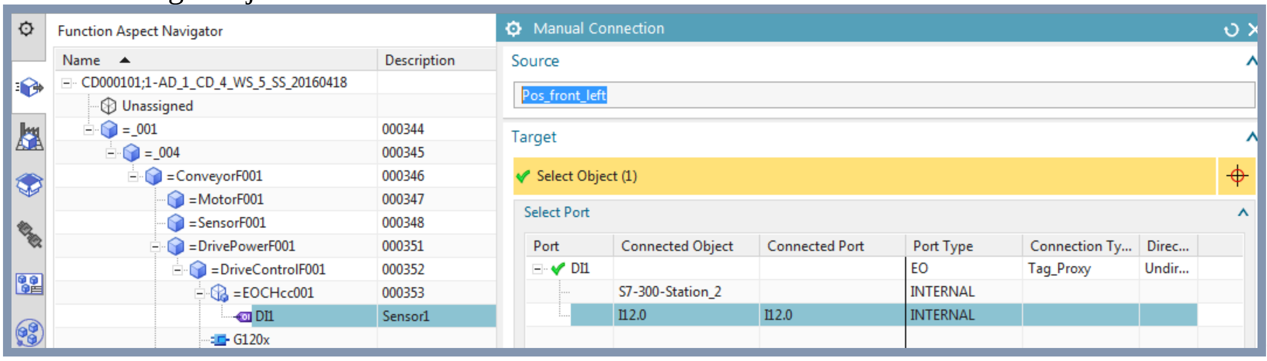
2. Right-click on **Pos_front_left** and select **Manual Connection**.



```
Network 11:
    A(
    A "Pos_front_le
    A "slow_forw"
    O
    A "pos_back_lef
    A "slow_back"
```

08b_36

3. For the target object select **DI1**.



Function Aspect Navigator

Name	Description
CD000101;1-AD_1_CD_4_WS_5_SS_20160418	
Unassigned	
=_001	000344
=_004	000345
=ConveyorF001	000346
=MotorF001	000347
=SensorF001	000348
=DrivePowerF001	000351
=DriveControlF001	000352
=EOCHcc001	000353
DI1	Sensor1
G120x	

Manual Connection

Source

Pos_front_left

Target

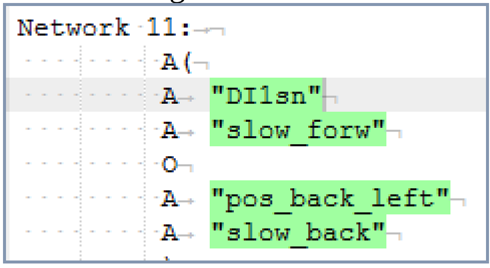
Select Object (1)

Select Port

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

08b_37

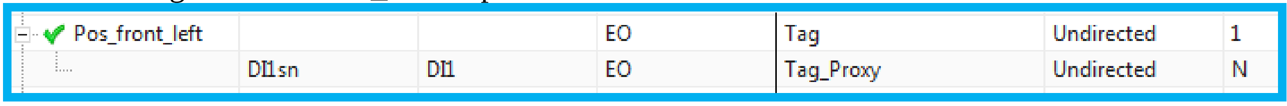
The following shows the result.



```
Network 11:
    A(
    A "DI1sn"
    A "slow_forw"
    O
    A "pos_back_left"
    A "slow_back"
```

08b_38

The following shows the RB_AT FB port.



✓ Pos_front_left			EO	Tag	Undirected	1
	DI1sn	DI1	EO	Tag_Proxy	Undirected	N

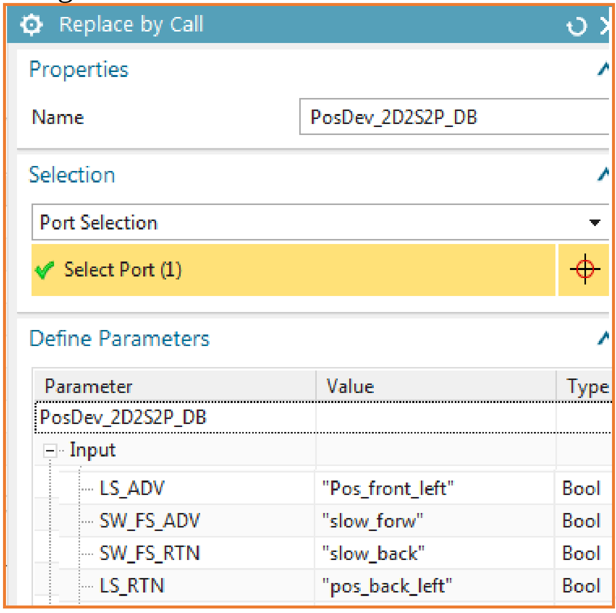
08b_39

8.5.4. RB_AT->PosDev replace by call

TERRY: nothing in user guide defining replace by call.
If the call was defined in TIA Portal and you already linked the symbolic reference to the IDB or FC, the **Define Parameters** group displays the call parameters that were defined in TIA Portal. By replacing parameter values, you can overwrite these parameters and add values for parameters that were not defined in TIA Portal. This will replace the system-defined caller rule with a user-defined caller rule.

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.



Replace by Call

Properties

Name: PosDev_2D2S2P_DB

Selection

Port Selection

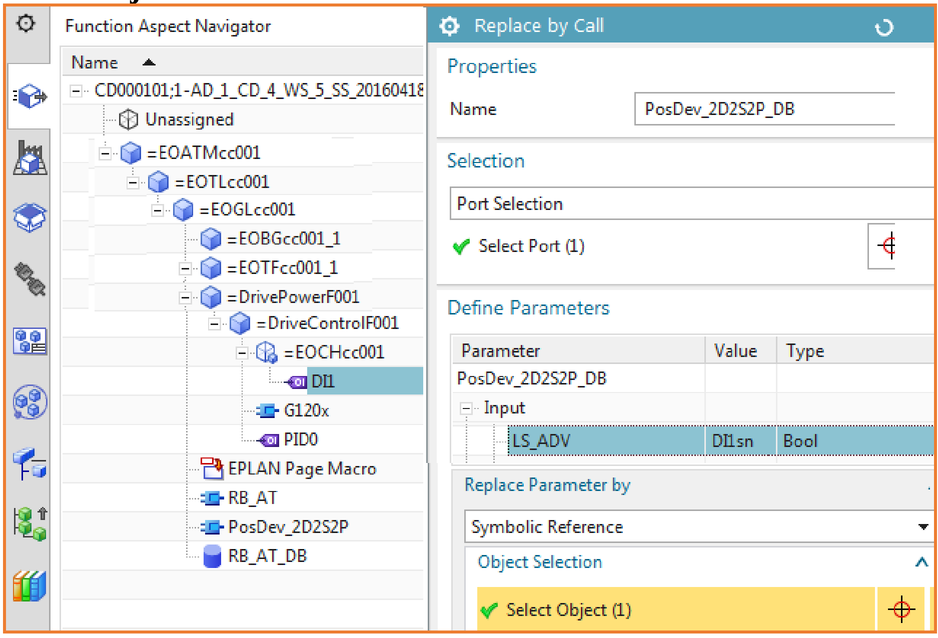
✓ Select Port (1)

Define Parameters

Parameter	Value	Type
PosDev_2D2S2P_DB		
Input		
LS_ADV	"Pos_front_left"	Bool
SW_FS_ADV	"slow_forw"	Bool
SW_FS_RTN	"slow_back"	Bool
LS_RTN	"pos_back_left"	Bool

08b_40

3. For **Replace Parameter by** select **Symbolic Reference**.
4. For **Object Selection** select **DI1**.



Replace by Call

Properties

Name: PosDev_2D2S2P_DB

Selection

Port Selection

✓ Select Port (1)

Define Parameters

Parameter	Value	Type
PosDev_2D2S2P_DB		
Input		
LS_ADV	DI1sn	Bool

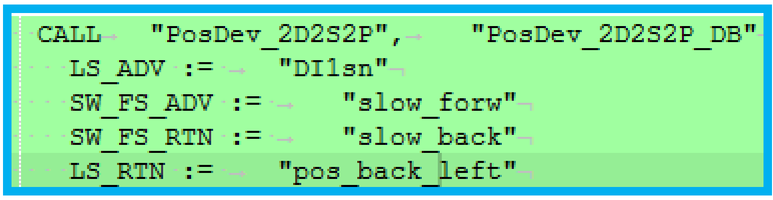
Replace Parameter by: Symbolic Reference

Object Selection

✓ Select Object (1)

08b_41

5. Click **OK**.



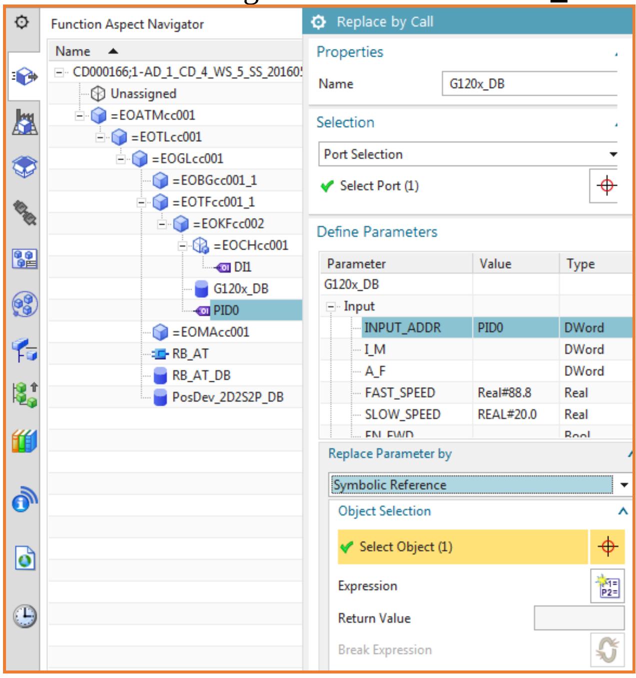
```
CALL   "PosDev_2D2S2P",    "PosDev_2D2S2P_DB"
  LS_ADV  :=     "DI1sn"
  SW_FS_ADV  :=     "slow_forw"
  SW_FS_RTN  :=     "slow_back"
  LS_RTN  :=     "pos_back_left"
```

08b_42

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 a the PID0 tag and the constant value you created earlier.

1. For G120x change the value for **INPUT_ADDR to PID0**.



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
EN_FWD		Bool

Replace Parameter by

Symbolic Reference

Object Selection

✓ Select Object (1)

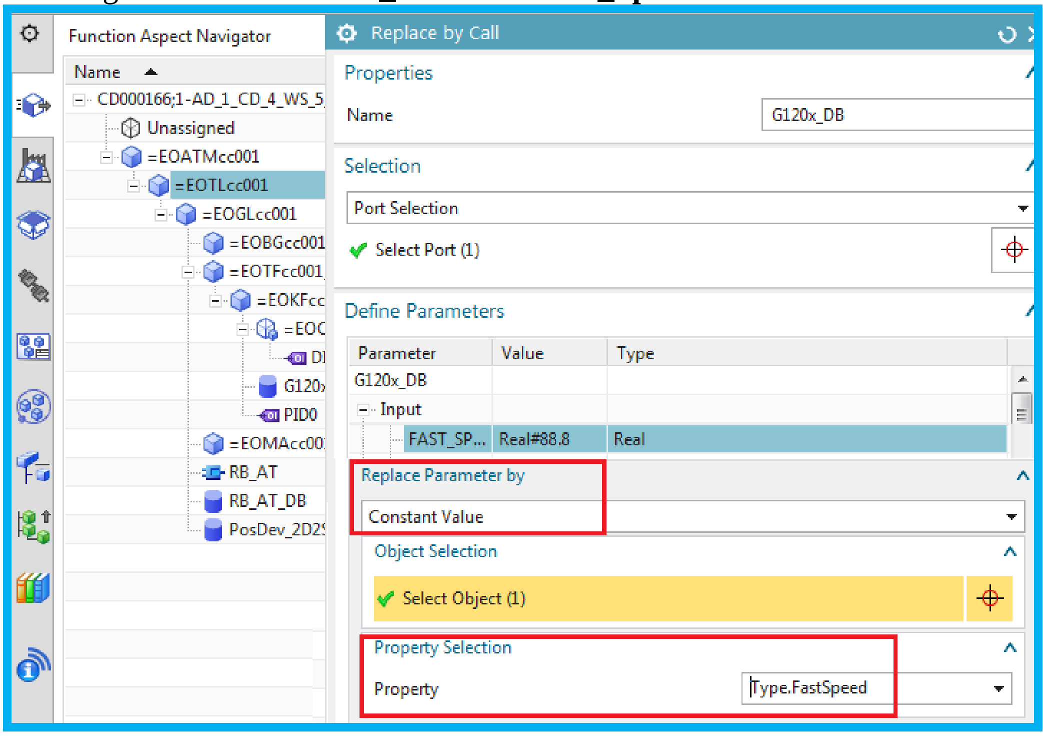
Expression

Return Value

Break Expression

08b_43

2. Change the value of **FAST_SPEED** to **Fast_Speed**.



Properties

Name: G120x_DB

Selection

Port Selection

✓ Select Port (1)

Define Parameters

Parameter	Value	Type
G120x_DB		
Input		
FAST_SP...	Real#88.8	Real

Replace Parameter by

Constant Value

Object Selection

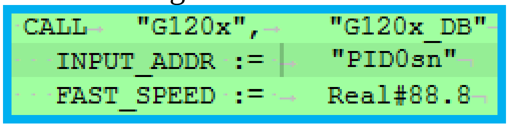
✓ Select Object (1)

Property Selection

Property: Type.FastSpeed

08b_44

The following shows the result.

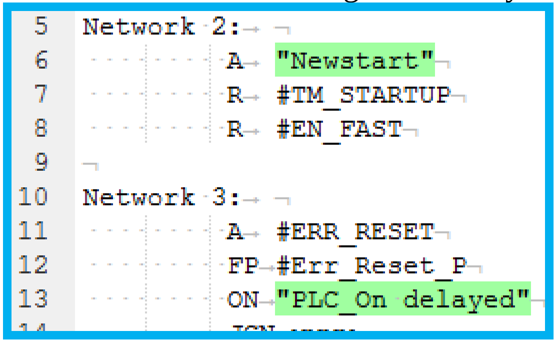


```
CALL    "G120x",    "G120x_DB"
   INPUT_ADDR :=     "PID0sn"
   FAST_SPEED :=     Real#88.8
```

08b_45

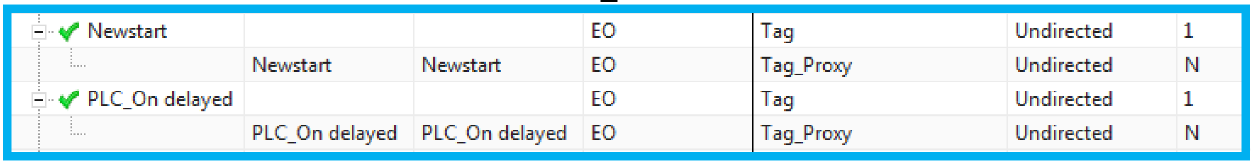
8.5.6. PosDev manual connect to Automation tags

The PosDev automation tags are already connected.



```
Network 2:
      A  "Newstart"
      R  #TM_STARTUP
      R  #EN_FAST

Network 3:
      A  #ERR_RESET
      FP #Err_Reset_P
      ON "PLC_On delayed"
```

08b_46

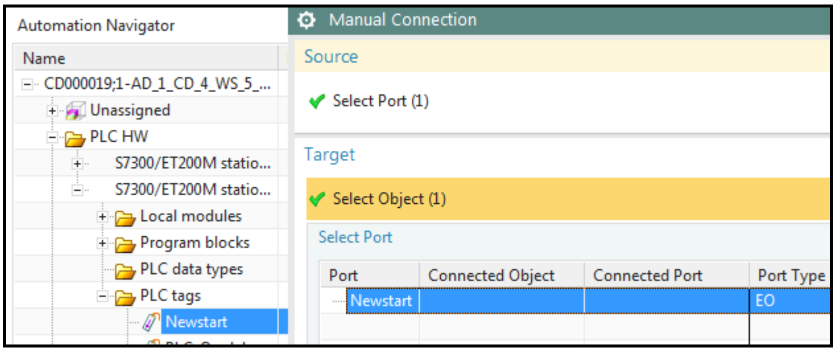


✓ Newstart			EO	Tag	Undirected	1
	Newstart	Newstart	EO	Tag_Proxy	Undirected	N
✓ PLC_On delayed			EO	Tag	Undirected	1
	PLC_On delayed	PLC_On delayed	EO	Tag_Proxy	Undirected	N

08b_47

If they are not connected, the do the following:

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



Automation Navigator

Name

- CD000019;1-AD_1_CD_4_WS_5_...
  - Unassigned
  - PLC HW
    - S7300/ET200M statio...
    - S7300/ET200M statio...
      - Local modules
      - Program blocks
      - PLC data types
      - PLC tags
        - Newstart

Manual Connection

Source

✓ Select Port (1)

Target

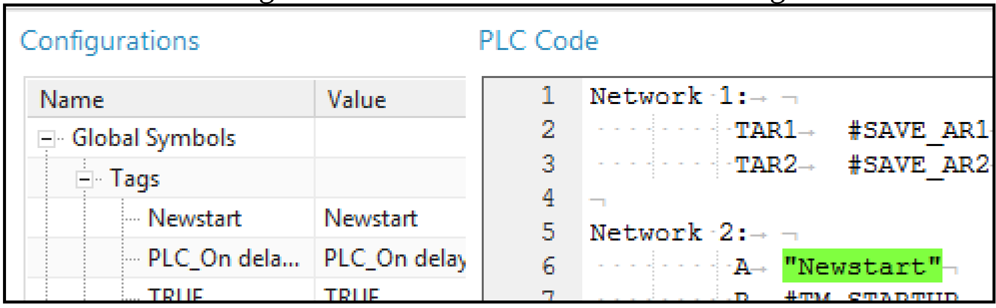
✓ Select Object (1)

Select Port

Port	Connected Object	Connected Port	Port Type
Newstart			EO

08b_48

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



Configurations

Name	Value
Global Symbols	
Tags	
Newstart	Newstart
PLC_On dela...	PLC_On delay
TRUE	TRUE

PLC Code

```
Network 1:
      TAR1  #SAVE_AR1
      TAR2  #SAVE_AR2

Network 2:
      A  "Newstart"
```

08b_49

8.6. Assign software to hardware

TERRY: assigning software and hardware not defined in user guide… maybe this ….

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 hardware** folder collects all the PLC stations of the project. Every PLC station has the following nodes:

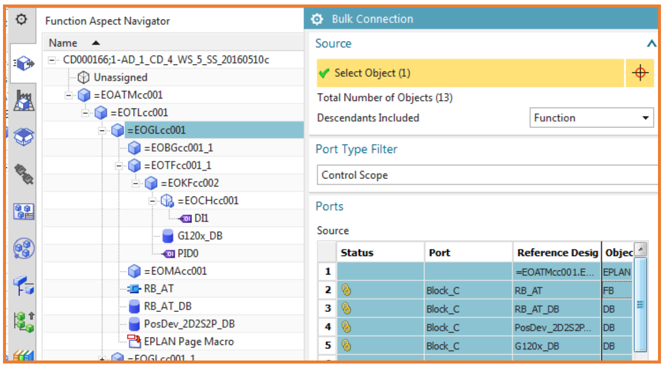
o The **PLC modules** folder collects the PLC station components. It contains the I/O modules of the PLC station, decentralized stations, or field devices. The channels are under their I/O module.

o The **PLC tags** folder collects the tags that belong to the PLC station's control scope.

o The **Program blocks** folder collects the program blocks that belong to the PLC station's control scope.

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.

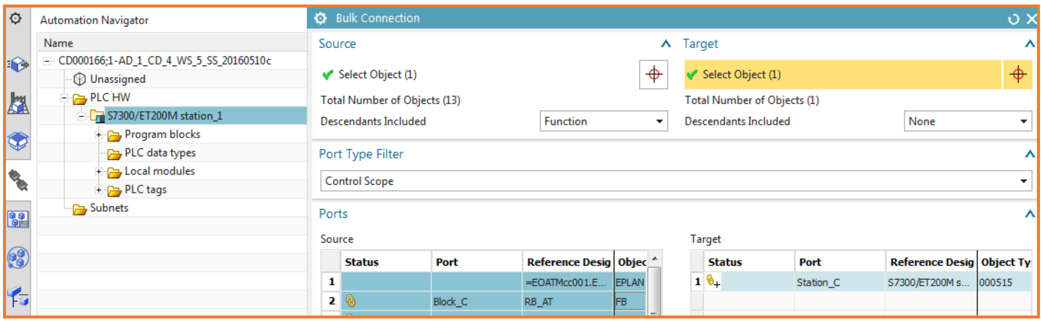
1. Select **Bulk Connection**.
2. For **Source** select **GL**.
3. Select the ports.



	Status	Port	Reference Desig	Objec
1			=EOATMcc001.E...	EPLAN
2		Block_C	RB_AT	FB
3		Block_C	RB_AT_DB	DB
4		Block_C	PosDev_2D2S2P...	DB
5		Block_C	G120x_DB	DB

08b_50

4. Under **Target** select **Select Object**.
5. Select the station.
6. Select the target ports.

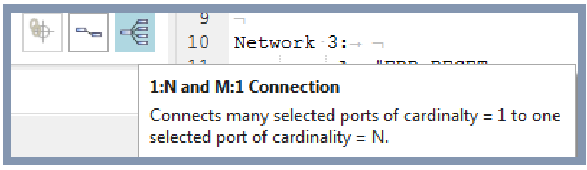


	Status	Port	Reference Desig	Objec
1			=EOATMcc001.E...	EPLAN
2		Block_C	RB_AT	FB

	Status	Port	Reference Desig	Object Ty
1		Station_C	S7300/ET200M s...	000515

08b_51

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

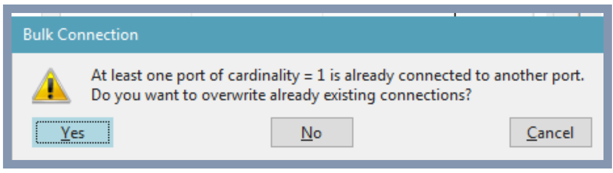


1:N and M:1 Connection

Connects many selected ports of cardinalty = 1 to one selected port of cardinalty = N.

08b_52

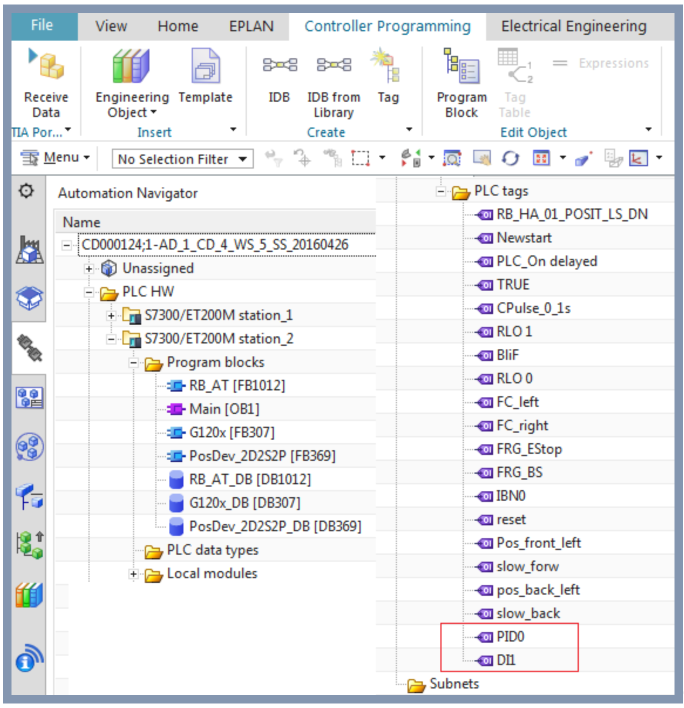
8. The bulk connection message appears. Click **OK**.



At least one port of cardinality = 1 is already connected to another port.
Do you want to overwrite already existing connections?

08b_53

The following shows the result.



- CD000124;1-AD_1_CD_4_WS_5_SS_20160426
  - Unassigned
  - PLC HW
    - S7300/ET200M station_1
    - S7300/ET200M station_2
      - Program blocks
        - RB_AT [FB1012]
        - Main [OB1]
        - G120x [FB307]
        - PosDev_2D2S2P [FB369]
        - RB_AT_DB [DB1012]
        - G120x_DB [DB307]
        - PosDev_2D2S2P_DB [DB369]
      - PLC data types
      - Local modules

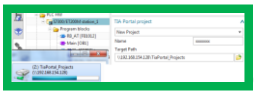
- PLC tags
  - RB_HA_01_POSIT_LS_DN
  - Newstart
  - PLC_On delayed
  - TRUE
  - CPulse_0_1s
  - RLO 1
  - BliF
  - RLO 0
  - FC_left
  - FC_right
  - FRG_EStop
  - FRG_BS
  - IBN0
  - reset
  - Pos_front_left
  - slow_forw
  - pos_back_left
  - slow_back
  - PID0
  - DI1
- Subnets

08b_54

8.7. Send data to TIA Portal

\\192.168.154.128\TiaPortal_Projects\3333\Project1_ohne_startdrive_V13_SP1_V14\Project1_ohne_startdrive_V13_SP1_V14.ap14

Note: Mapped drive to the VM must be connected or get error.



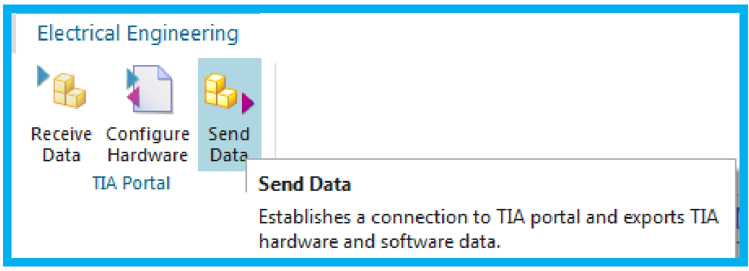
PLC HW
S7300/ET200M station_1
Program blocks
RB_AT [FB1012]
Main [OB1]
TIA Portal project
New Project
Name
Target Path
\\192.168.154.128\TiaPortal_Projects
(Z:) TiaPortal_Projects (\\192.168.154.128)

08b_55

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.

You now want to send the data to TIA portal.

1. Click **Send Data**.

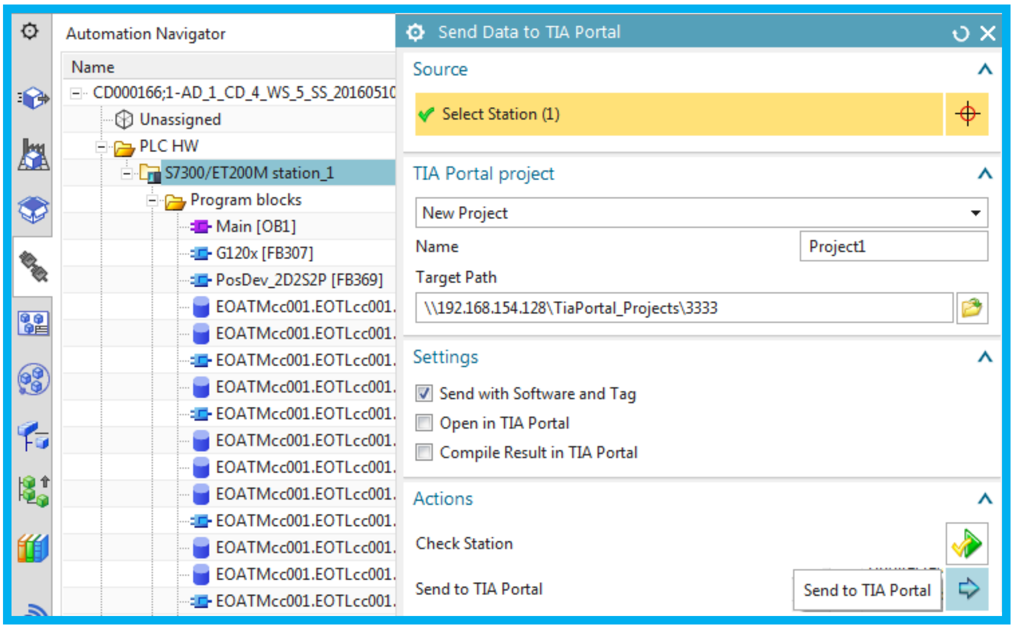


Send Data

Establishes a connection to TIA portal and exports TIA hardware and software data.

08b_56

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**.



CD000166;1-AD_1_CD_4_WS_5_SS_20160510
Unassigned
PLC HW
S7300/ET200M station_1
Program blocks
Main [OB1]
G120x [FB307]
PosDev_2D2S2P [FB369]
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.
EOATMcc001.EOTLcc001.

Send Data to TIA Portal

Source

✓ Select Station (1)

TIA Portal project

New Project

Name: Project1

Target Path: \\192.168.154.128\TiaPortal_Projects\3333

Settings

☑ Send with Software and Tag
□ Open in TIA Portal
□ Compile Result in TIA Portal

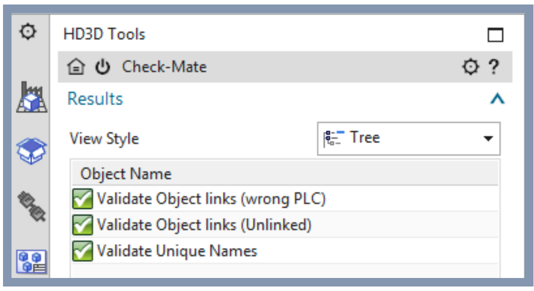
Actions

Check Station

Send to TIA Portal

08b_57

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

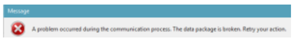


View Style: Tree

Object Name
☑ Validate Object links (wrong PLC)
☑ Validate Object links (Unlinked)
☑ Validate Unique Names

08b_58

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

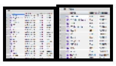
Always get this eror. 

Message

A problem occurred during the communication process. The data package is broken. Retry your action.

Created project. but empty.

Should be something like this (Andreas said this is an Automation Designer error, not mine).

08_100 08_101 08_102  08_103

Part 3. Create/instantiate template

This part describes how to:

9. Template-related concepts.
10. Configure template-ready EPLAN.
11. Configure template-ready TIA Portal. how to add use aspect chain info to software block and tag names.
12. Create/instantiate template. How to create and insert templates.

9. 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.

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.

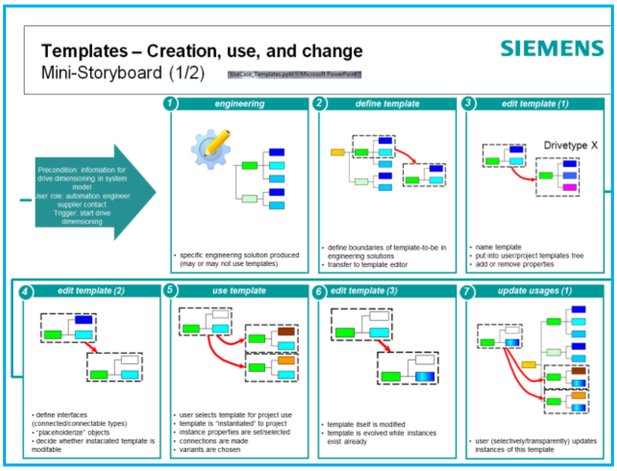
The template definitions are made available to the project engineers in the Solution Library and in the Type Library. In the projects, the project engineers can use each template as many times as required.

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.

Basic template concepts include:

- 9.1. Getting aspect chain of parent Engineering Object using expressions
- 9.2. Getting aspect chain ID of non-parent Engineering Object with 2 ports + link
- 9.3. Getting aspect chain ID outside template with dynamic connection (software only)
- 9.4. Automatic generation of calls for inserted software

See ReqSpec_AD@NX_Templates.docx
20160311 \\debonkl0c19\ADNX\Teams\PRM\Topics\Templates\Material\UseCase_Templates.pptx



Templates – Creation, use, and change
Mini-Storyboard (1/2)

Precondition: information for drive dimensioning in system model
User role: automation engineer supplier contact
Trigger: start drive dimensioning

1 engineering
- specific engineering solution produced (may or may not use templates)

2 define template
- define boundaries of template-to-be in engineering solutions
- transfer to template editor

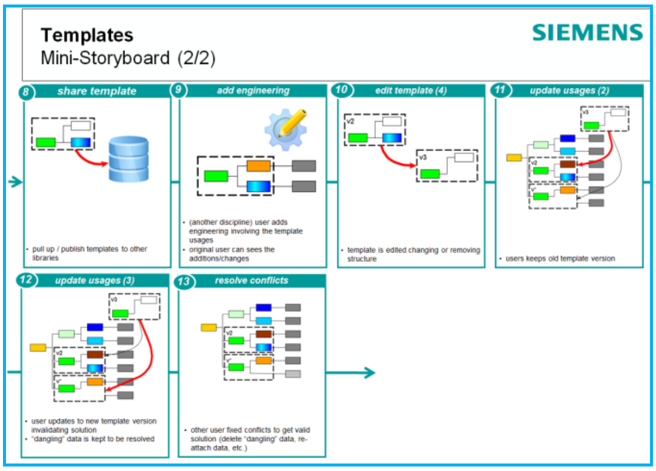
3 edit template (1)
- name template
- put into user/project templates tree
- add or remove properties

4 edit template (2)
- define interfaces (connected/connectable types)
- "placeholderize" objects
- decide whether instaciated template is modifable

5 use template
- user selects template for project use
- template is "instantiated" to project
- instance properties are set/selected
- connections are made
- variants are chosen

6 edit template (3)
- template itself is modified
- template is evolved while instances exist already

7 update usages (1)
- user (selectively/transparently) updates instances of this template



Templates
Mini-Storyboard (2/2)

8 share template
- pull up / publish templates to other libraries

9 add engineering
- (another discipline) user adds engineering involving the template usages
- original user can sees the additions/changes

10 edit template (4)
- template is edited changing or removing structure

11 update usages (2)
- users keeps old template version

12 update usages (3)
- user updates to new template version invalidating solution
- "dangling" data is kept to be resolved

13 resolve conflicts
- other user fixed conflicts to get valid solution (delete "dangling" data, re-attach data, etc.)

9.1. Getting aspect chain ID of parent Engineering Object 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.

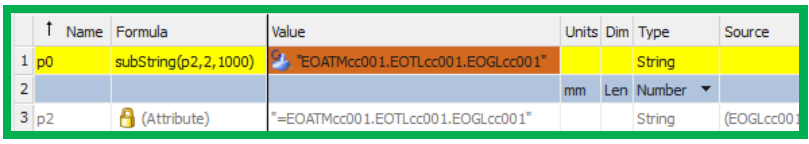
EPLAN

~~In section 7.2 "Add PM250D macro"~~

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

1. Move the parent Engineering Object then the aspect would no longer be valid.
2. Instantiate a template or copy a conveyor with same values, then you will have EPLAN reports with duplicate names.

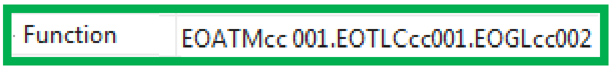
The solution is to use expressions to make the value of the **Function** property depend on the Engineering Object (GL) the macro was placed under in the aspect tree. Thus the property value is unique. This is not required to generate a report, but is required to avoid confusion when actually running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

09b_01

The resulting property value is the following.



Function	EOATMcc 001.EOTLCcc001.EOGLcc002

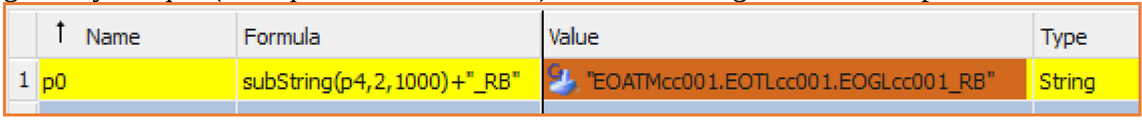
09b_02

TIA Portal (software blocks and tags)

~~In chapter 8 "Configure (non-template) TIA Portal"~~

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 same values, the symbolic name of software blocks and tags will be repeated (which will generate an error in TIA Portal).

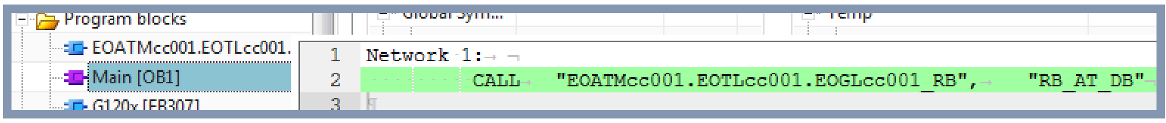
The solution for the software is to use expressions to make the symbolic name of RB_AT FB depend on the Engineering Object (GL) the Function Block was placed under in the aspect tree. Thus the symbolic name is globally unique (as required in TIA Portal). The following shows the expression.



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

09b_03

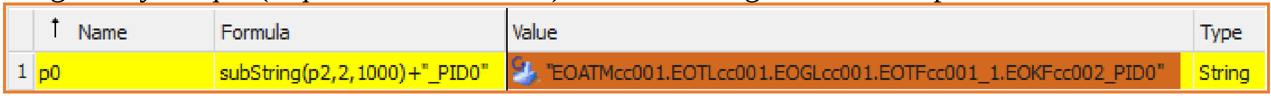
The resulting symbolic name for the Function Block is the following.



```
Network 1:
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",     "RB_AT_DB"
```

09b_04

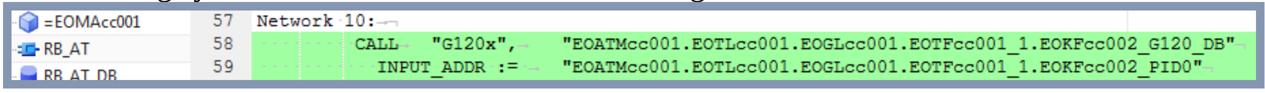
The solution for tags is to use expressions to make the symbolic name of the tags (PID0 and DI1) depend on the Engineering Objects (KF and CH) they were placed under in the aspect tree. Thus the symbolic names are globally unique (required in TIA Portal). The following shows the expression for PID0.



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

09b_06

The resulting symbolic name for PID0 is the following.



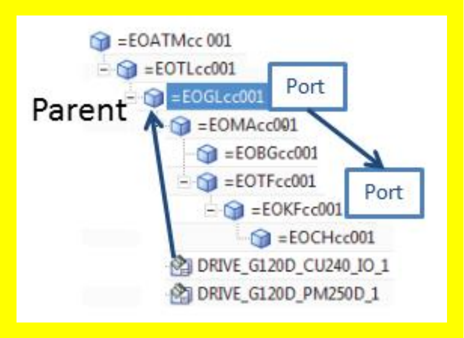
```
Network 10:
        CALL   "G120x",     "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_G120_DB"
          INPUT_ADDR :=     "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_PID0"
```

09b_07

9.2. Getting aspect chain ID of non-parent Engineering Object with 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.



09b_08

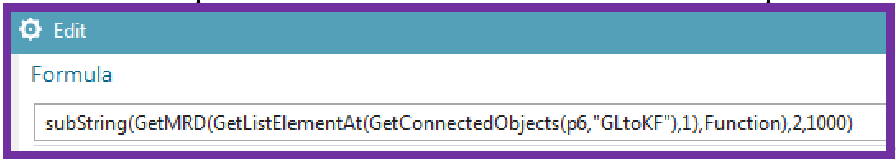
To configure this you do the following:

1. Create a port for parent Engineering Object GL.
2. Create a port for target Engineering Object KF.
3. Link the ports.
4. Reference the parent Engineering Object port name in a property expression.

EPLAN

~~In 7.2 "Add PM250 macro" you simply entered text for device names.~~ 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. ~~In 10.2 “KF01.name ports, link, and expressions”~~ 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).

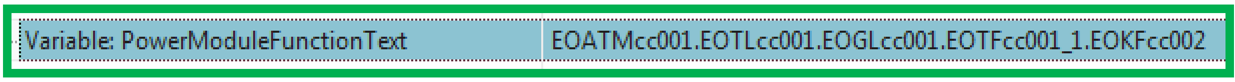


Formula

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

09b_11

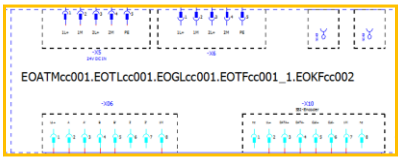
The following shows the resulting property value.



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

09b_13

The following shows the result in the generated macro report.



EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002

09b_14

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.

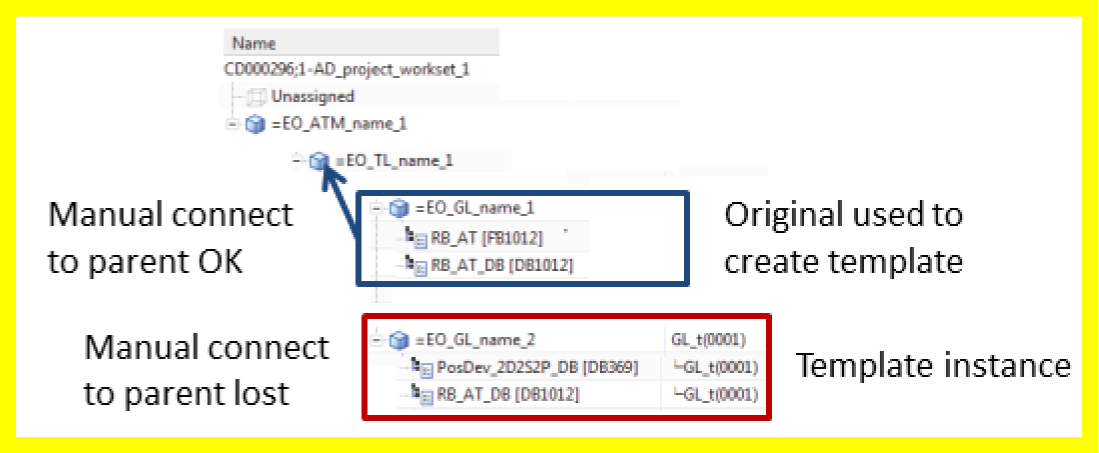
For tags you might want to have the symbolic name for a tag set to the aspect chain of a non-parent Engineering Object. For example, for the DI input tag for the boolean data from the light sensors could use the aspect chain ID of the physical sensor as the symbolic name of the tag. This Getting Started 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).



```
Name
CD000296;1-AD_project_workset_1
  Unassigned
  =EO_ATM_name_1
      =EO_TL_name_1
        =EO_GL_name_1
          RB_AT [FB1012]
          RB_AT_DB [DB1012]
        =EO_GL_name_2                      GL_t(0001)
          PosDev_2D2S2P_DB [DB369]        └GL_t(0001)
          RB_AT_DB [DB1012]               └GL_t(0001)
```

Manual connect to parent OK

Original used to create template

Manual connect to parent lost

Template instance

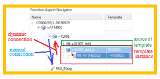
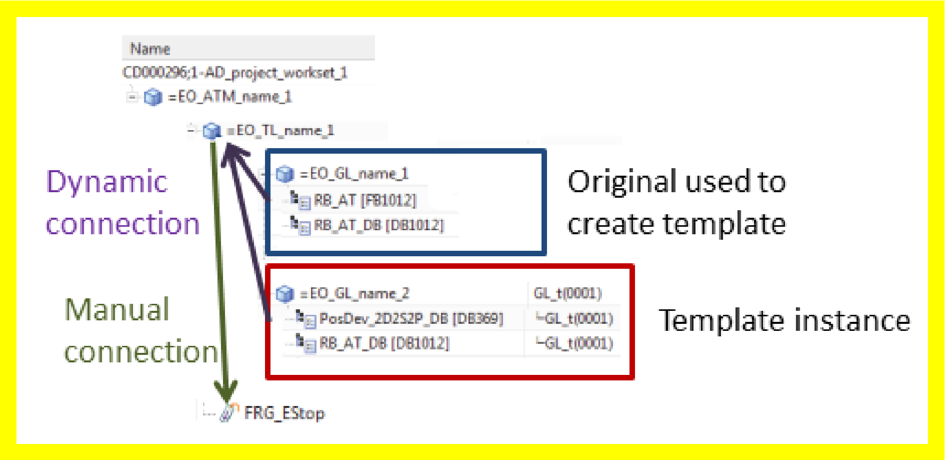
09b_15

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 (11.2)

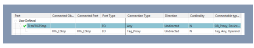
The following diagram shows how with a dynamic expression (red arrow to TL01) the template instance automatically locates the required parent Engineering Object TL. TL has a manual link (blue) to the tag FRG_EStop. Thus the instantiated template can link to the tag (as long as the template instance has TL as parent Engineering Object).

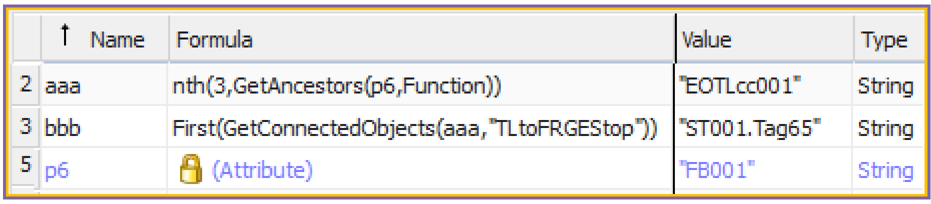
09b_16

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.


09b_17

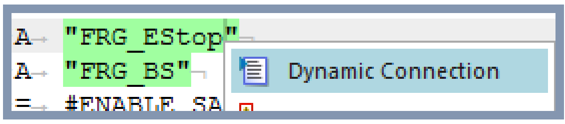
3. Create the dynamic connection from RBAT FB to the tag via the TL port using the following expressions (P6 is RB_AT FB object name).



	↑ 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

09b_18 xxxxx

4. Copy this GL or create a template and instantiate the template.
The result is a dynamic connection to FRG_Estop.



```
A→ "FRG_EStop"
A→ "FRG_BS"
=→ #ENABLE_SA
```

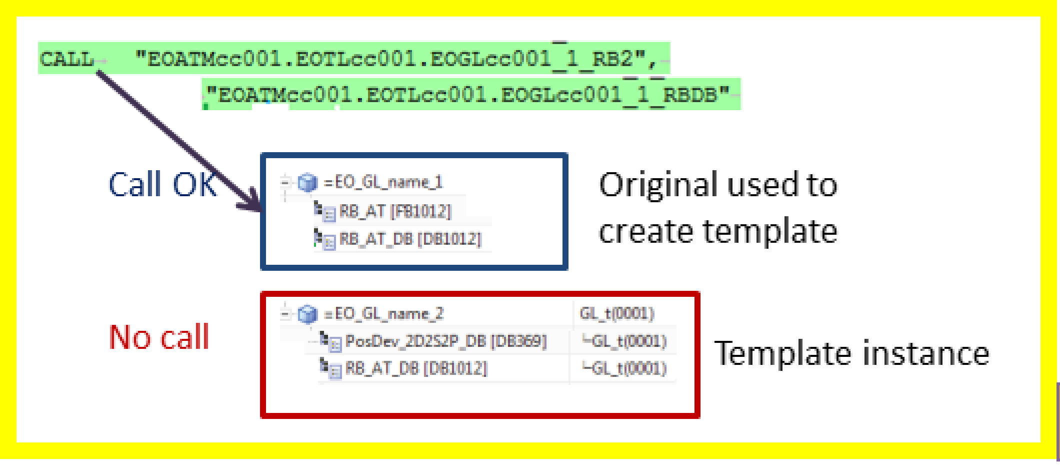
09b_19

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.



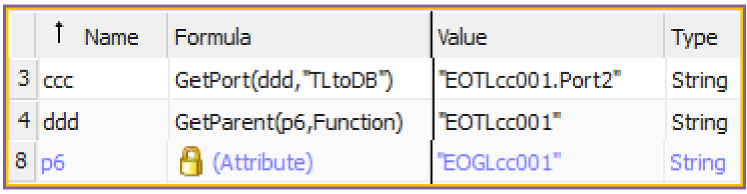
```
CALL  "EOATMcc001.EOTLcc001.EOGLcc001_1_RB2",
      "EOATMcc001.EOTLcc001.EOGLcc001_1_RBDB"
```

09b_20xx

Solution (11.3)

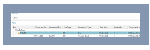
Now you need to enable OB Main to automatically add a call to an added GL. To do this:

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).



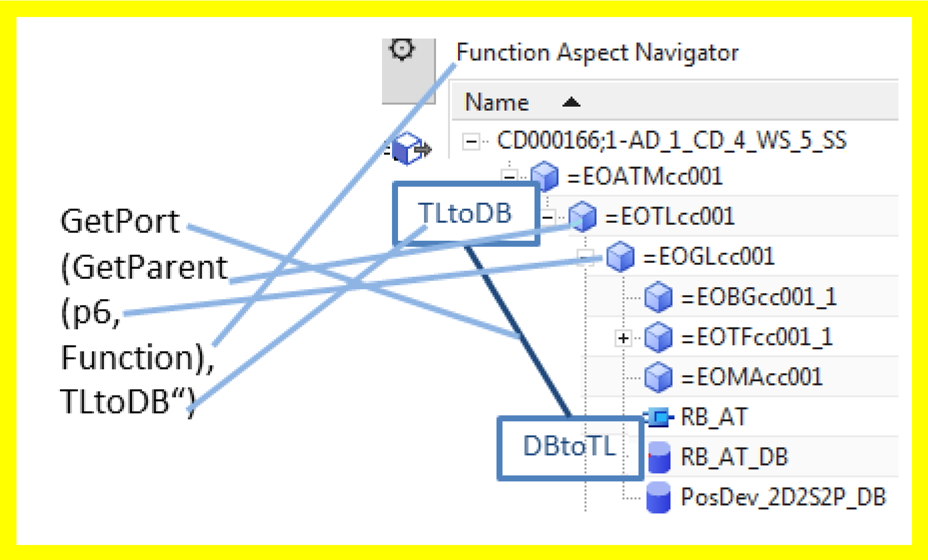
	↑ Name	Formula	Value	Type
3	ccc	GetPort(ddd,"TLtoDB")	"EOTLcc001.Port2"	String
4	ddd	GetParent(p6,Function)	"EOTLcc001"	String
8	p6	🔒 (Attribute)	"EOGLcc001"	String

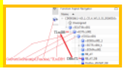
09b_22 xxxxxxx



09b_23

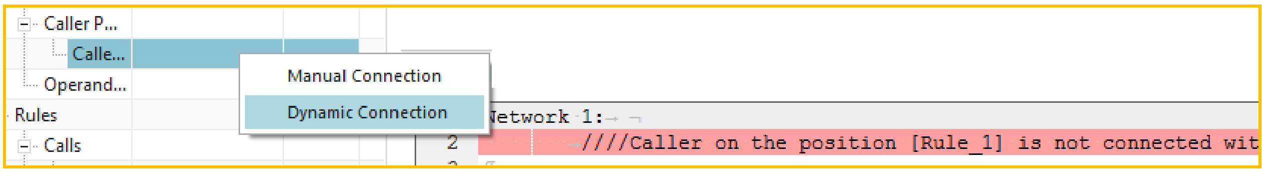
This diagram explains the expression components.





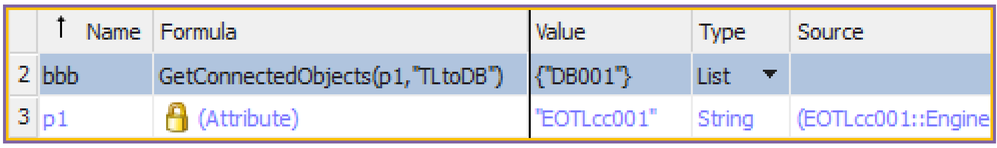
09b_21

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



```
Network 1:
////Caller on the position [Rule_1] is not connected wit
```

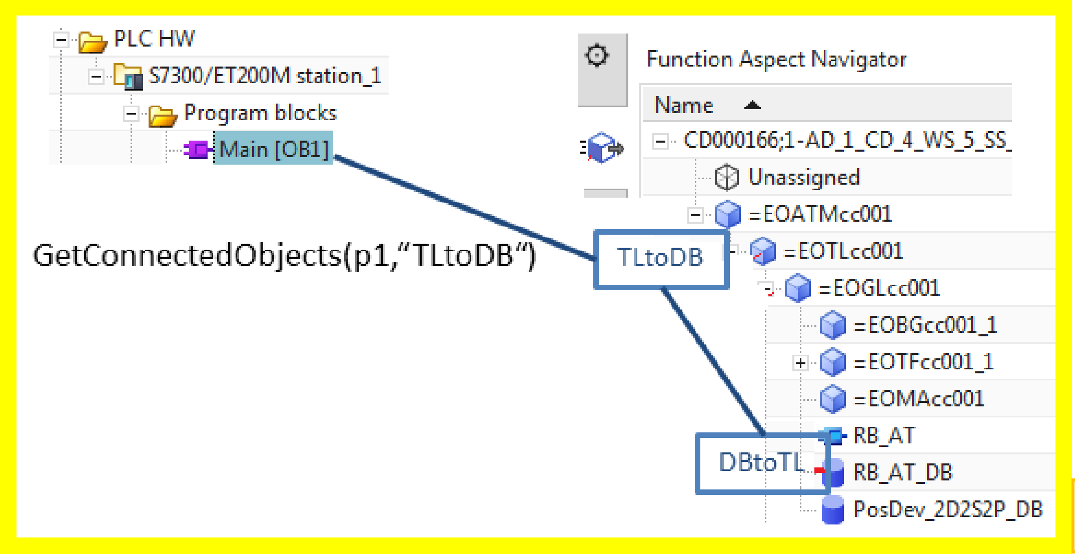
09b_25

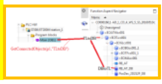


	Name	Formula	Value	Type	Source
2	bbb	GetConnectedObjects(p1,"TLtoDB")	{"DB001"}	List	
3	p1	(Attribute)	"EOTLcc001"	String	(EOTLcc001::Engine

09b_26 xxxxx

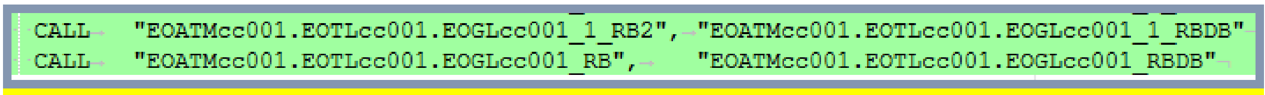
The following diagram shows the connections.





09b_24

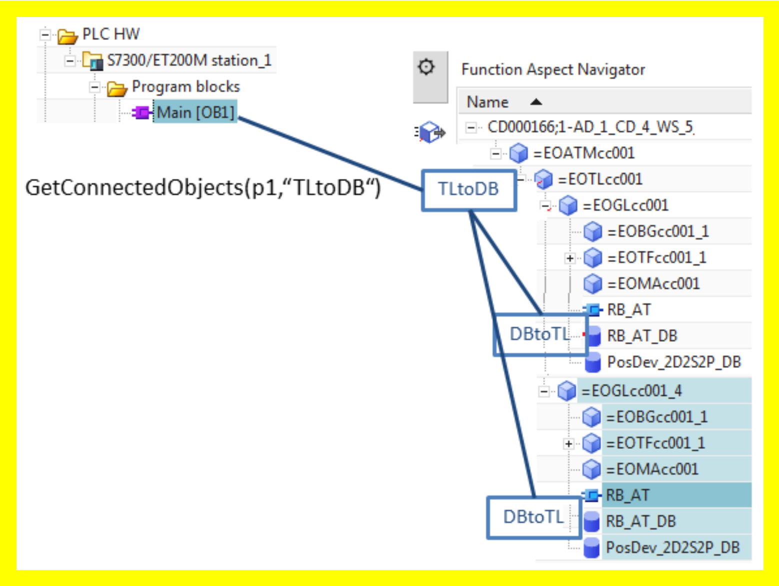
4. When you copy GL or instantiate the template, the OB main calls will automatically update.



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

09b_28

The following diagram shows the connections.



09b_27

10. Configure a template-ready AD project for EPLAN

>>PUT ELSEWHERE: 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. You can use the templates that are shipped with Automation Designer, modify the logic behind them, or create your own adaptive templates.

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

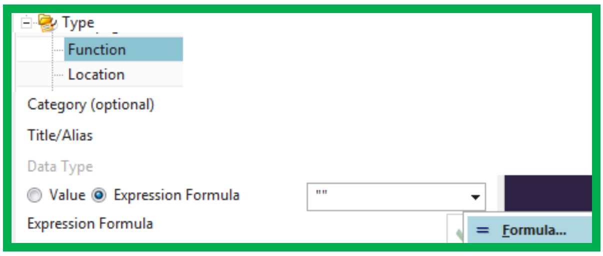
10.2. KF01.name ports, link, and expression

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.

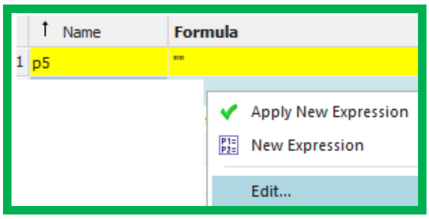
1. First add the expression for the property Function of the macro. Open the properties for the macro. Under **Type** select **Function**. Select **Expression Formula**. Right-click on the drop-down list and select **Formula…**.



Type
Function
Location
Category (optional)
Title/Alias
Data Type
○ Value ● Expression Formula ""
Expression Formula
= Formula...

10b_01

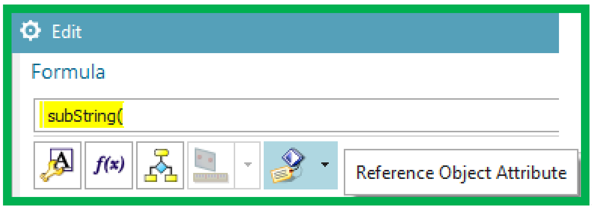
2. Click **Edit**.



Name	Formula
1 p5	""

10b_02

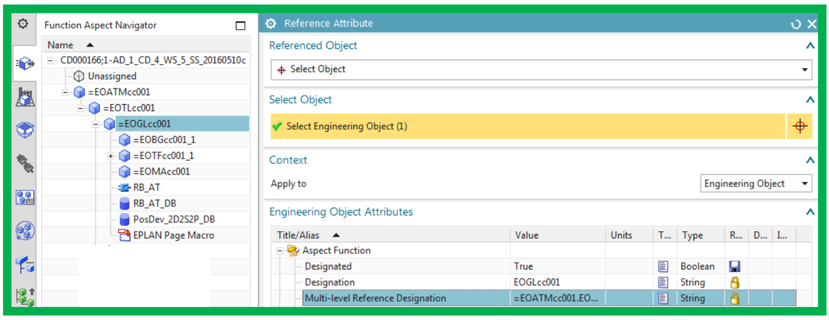
3. Enter "subString(" .



Edit
Formula
subString(

10b_03

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



Apply to: Engineering Object

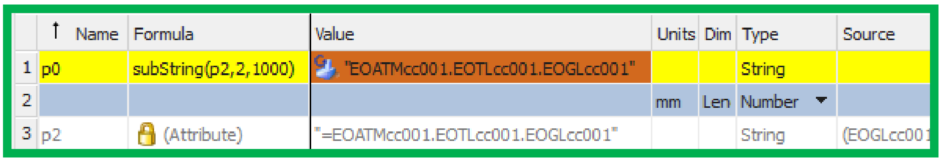
Engineering Object Attributes

Title/Alias	Value	Units	T...	Type	R...	D...	I...
Aspect Function							
Designated	True			Boolean			
Designation	EOGLcc001			String			
Multi-level Reference Designation	=EOATMcc001.EO...			String			

10b_04

5. Add the the formula ",2,1000)". This specifies to create a substring from character 2 and including the first 1000 characters.

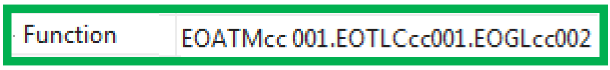
6. Click **OK**.



	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

10b_05

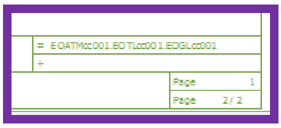
The following shows the result.



Function	EOATMcc 001.EOTLCcc001.EOGLcc002

10b_06

The following shows the value for "Function" in a generated report.



= EOATMcc001.EOTLcc001.EOGLcc001
+
Page 1
Page 2 / 2

10b_23

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

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.

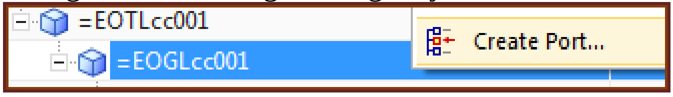
Ports are a means of connecting objects. They are available for Engineering Objects and tags. You connect the port of one object, the source port, to the port of another object, the target port. This creates a bidirectional connection. *Connections* represent port-based relations between objects.

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 conected 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.

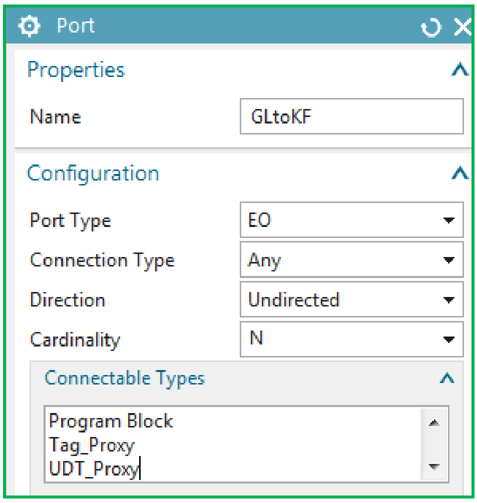
System-defined ports are automatically created by Automation Designer, either when you insert an object in a project, or when you carry out actions that internally require port connections. You can connect or disconnect system-defined ports. You cannot create or delete them, or edit their settings.
User-defined 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.

1. Right-click on Engineering Object GL and select **Create Port…**.



10b_07

2. Enter the GLtoKF port information.



Properties

Name: GLtoKF

Configuration

Port Type: EO

Connection Type: Any

Direction: Undirected

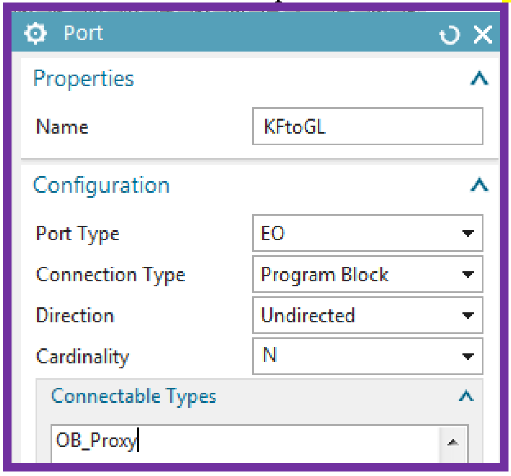
Cardinality: N

Connectable Types

Program Block
Tag_Proxy
UDT_Proxy

10b_08

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**.



Properties

Name: KFtoGL

Configuration

Port Type: EO

Connection Type: Program Block

Direction: Undirected

Cardinality: N

Connectable Types

OB_Proxy

10b_09

6. Click **OK**.

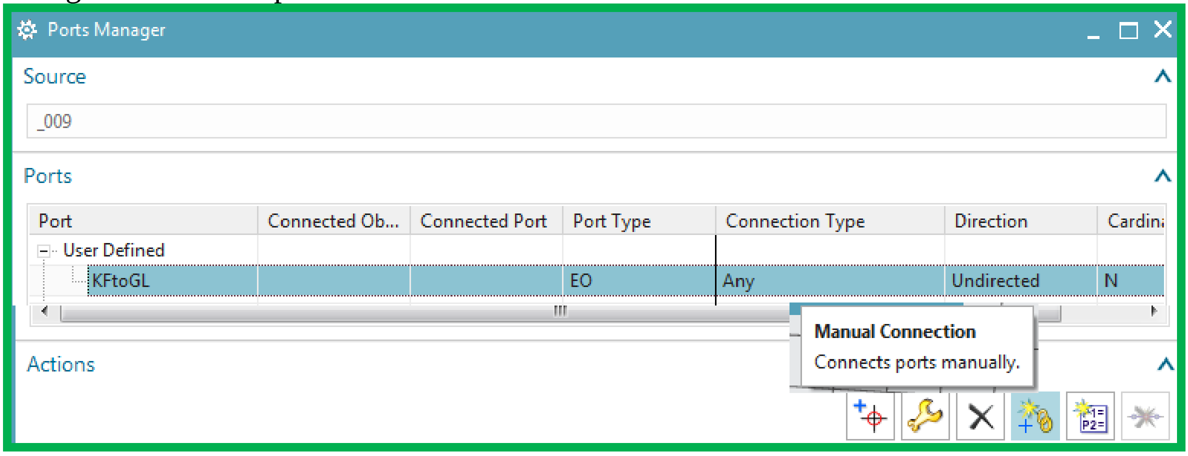
Connection (link)

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**.



Ports Manager

Source

_009

Ports

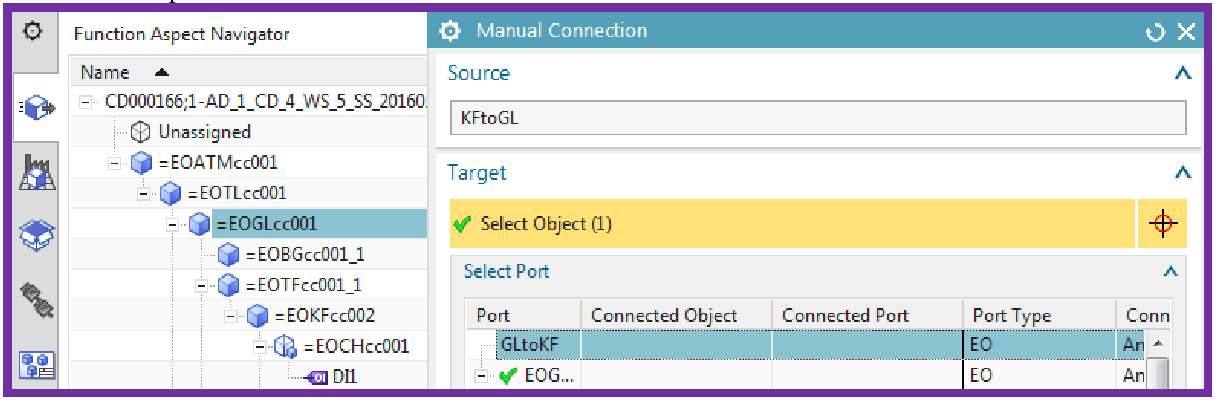
Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardin
User Defined						
KFtoGL			EO	Any	Undirected	N

Actions

Manual Connection
Connects ports manually.

10b_10

3. Select the target Engineering Object GL.
4. Select the port GLtoKF.



Manual Connection

Source

KFtoGL

Target

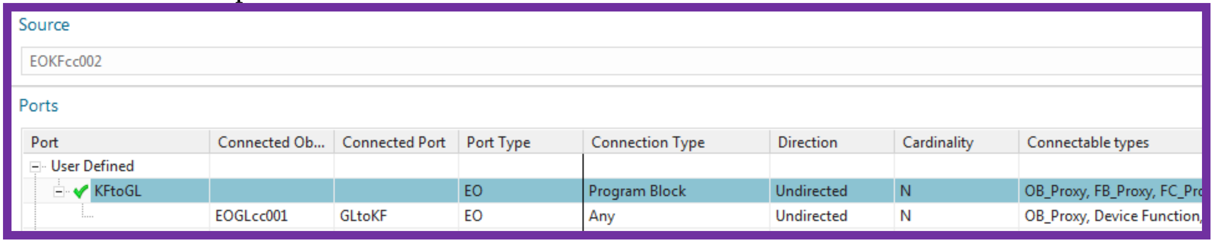
✓ Select Object (1)

Select Port

Port	Connected Object	Connected Port	Port Type	Conn
GLtoKF			EO	An
✓ EOG...			EO	An

10b_11

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



Source

EOKFcc002

Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
✓ KFtoGL			EO	Program Block	Undirected	N	OB_Proxy, FB_Proxy, FC_Pro
	EOGLcc001	GLtoKF	EO	Any	Undirected	N	OB_Proxy, Device Function,

10b_12

Expression

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:

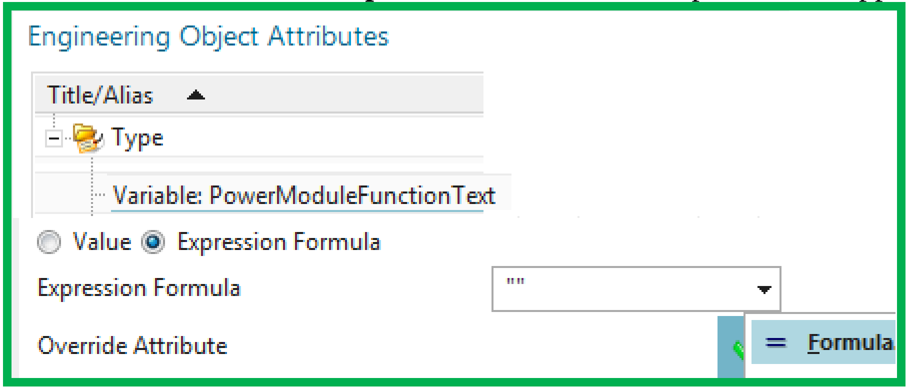
• To set a property value. You can use navigation expressions or an expression that creates an object reference.

• 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.

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.



Engineering Object Attributes

Title/Alias

Type

Variable: PowerModuleFunctionText

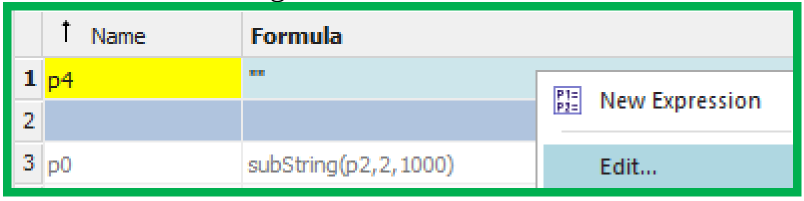
○ Value ● Expression Formula

Expression Formula: ""

Override Attribute

10b_13

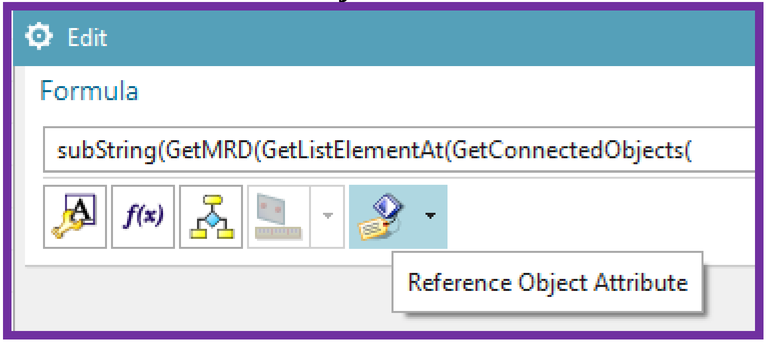
6. Click **Formula**. The Expressions dialog appears.
7. Under **Formula** right-click and select **Edit**.



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

10b_14

8. Enter “subString(GetMulti Reference Designation(GetListElementAt(GetConnectedObjects(“.
9. Click on **Reference Object Attribute**.



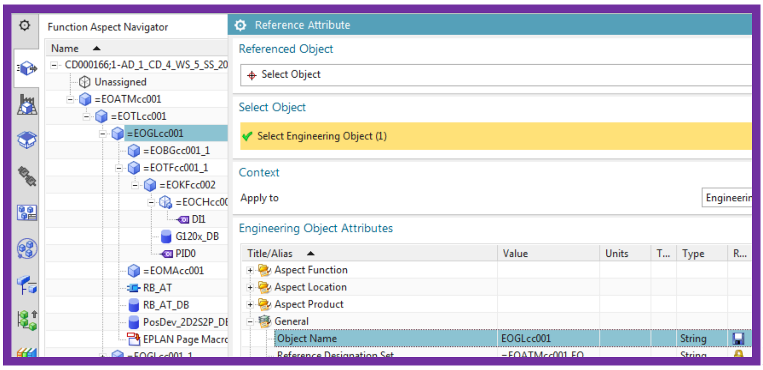
Edit

Formula

subString(GetMRD(GetListElementAt(GetConnectedObjects(

10b_15

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



Function Aspect Navigator

- CD000166;1-AD_1_CD_4_WS_5_SS_20
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
          - =EOKFcc002
            - =EOCHcc0(
              - DI1
            - G120x_DB
            - PID0
        - =EOMAcc001
        - RB_AT
        - RB_AT_DB
        - PosDev_2D2S2P_DI
        - EPLAN Page Macro
      - =EOGLcc001_1

Reference Attribute

Referenced Object

Select Object

Select Object

✓ Select Engineering Object (1)

Context

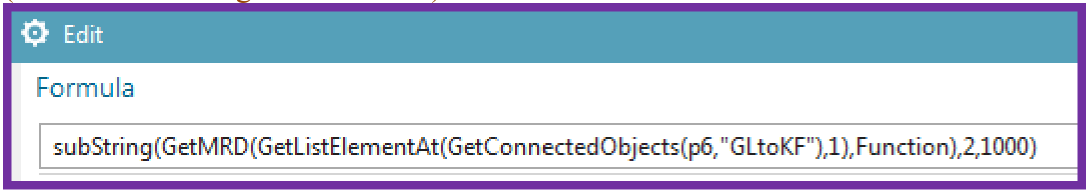
Apply to: Engineerin

Engineering Object Attributes

Title/Alias	Value	Units	T...	Type	R...
Aspect Function					
Aspect Location					
Aspect Product					
General					
Object Name	EOGLcc001			String	
Reference Designation Set	=EOATMcc001.EO			String	

10b_16

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 (wihout the leading “=” character).



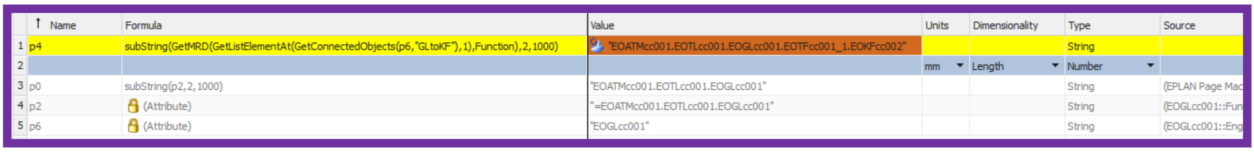
Edit

Formula

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

10b_17

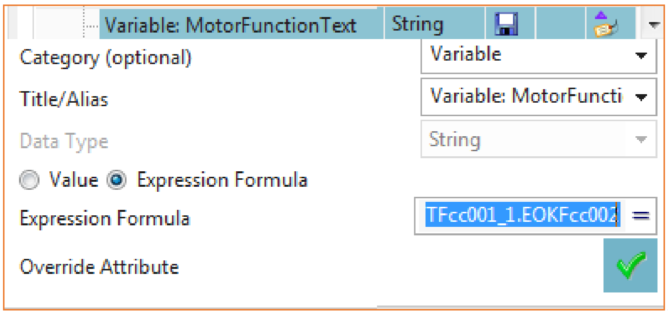
12. Click **OK**.



	Name	Formula	Value	Units	Dimensionality	Type	Source
1	p4	subString(GetMRD(GetListElementAt(GetConnectedObjects(p6,"GLtoKF"),1),Function),2,1000)	"EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002"			String	
2				mm	Length	Number	
3	p0	subString(p2,2,1000)	"EOATMcc001.EOTLcc001.EOGLcc001"			String	(EPLAN Page Mac
4	p2	(Attribute)	"=EOATMcc001.EOTLcc001.EOGLcc001"			String	(EOGLcc001::Fun
5	p6	(Attribute)	"EOGLcc001"			String	(EOGLcc001::Eng

10b_18

13. Click **OK**.



Variable: MotorFunctionText | String

Category (optional): Variable

Title/Alias: Variable: MotorFuncti

Data Type: String

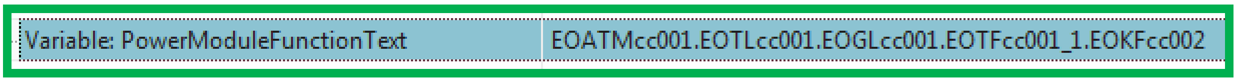
○ Value ● Expression Formula

Expression Formula: TFcc001_1.EOKFcc002

Override Attribute

10b_19

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



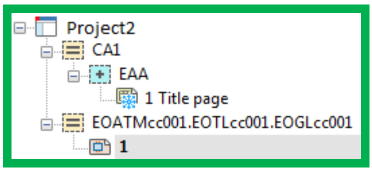
Variable: PowerModuleFunctionText | EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002

10b_20

15. Click **OK**.

10.3. Generate

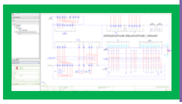
1. Generate EPLAN.

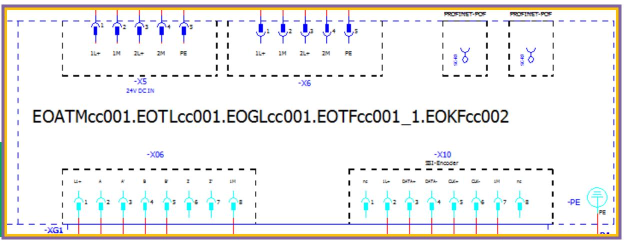


- Project2
  - CA1
    - EAA
      - 1 Title page
  - EOATMcc001.EOTLcc001.EOGLcc001
    - 1

10b_21

Note the property for KF “MotorFunctionText”.





EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002

10b_22 xxxxxxxxxx

11. Configure a template-ready AD project for TIA Portal

You’ve tested basic software generation previously for a project that was not template ready. Now you want to create a template from the GL conveyor.

11b_01

But you must first modify the conveyor. If you copy and paste another GL under TL, many software and tag ID’s (symbolic names) for the 2 GL’s will be the same. You need to make the symbolic names dependent on the aspect chain.

Now will do the following:

11.1. Configure symbolic names
11.2. FRGEStop dynamic connection
11.3. OB Main calls
11.4. Generate

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.

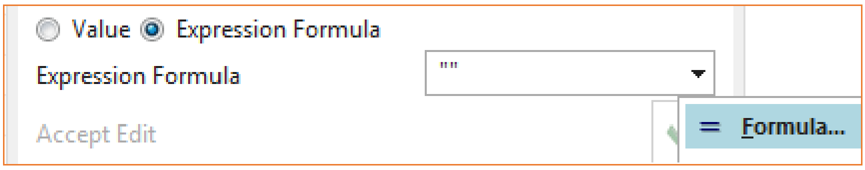
You need to configure symbolic names for the following

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**.

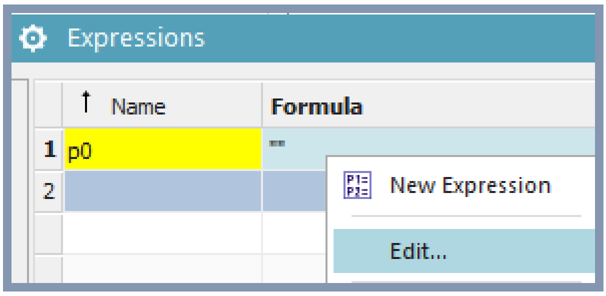


○ Value ● Expression Formula

Expression Formula: ""

11b_02

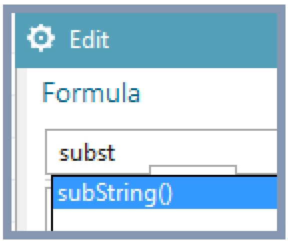
5. Right click in column **Formula**.



	Name	Formula
1	p0	""
2		

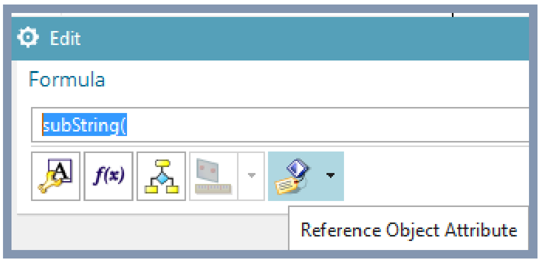
11b_03

6. Enter “subst” and accept the suggestion.



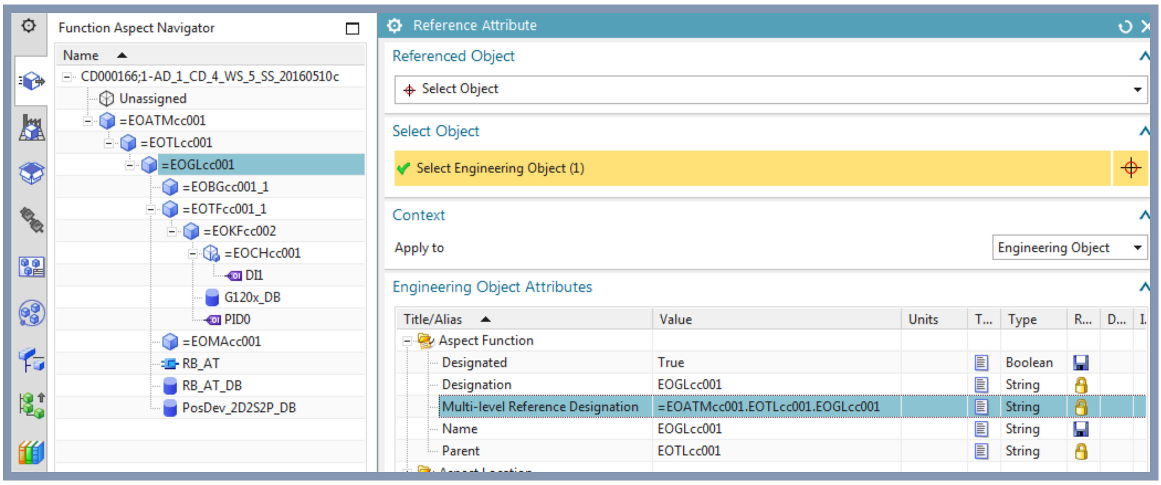
11b_04

7. Click **Reference Object Attribute**.



11b_05

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

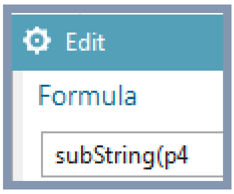


Apply to: Engineering Object

Title/Alias	Value	Units	T...	Type	R...	D...	I.
Aspect Function							
Designated	True			Boolean	💾		
Designation	EOGLcc001			String	🔒		
Multi-level Reference Designation	=EOATMcc001.EOTLcc001.EOGLcc001			String	🔒		
Name	EOGLcc001			String	💾		
Parent	EOTLcc001			String	🔒		

11b_06

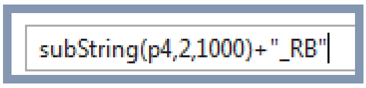
P4 = GL Function Multi Reference Designation.



subString(p4

11b_07

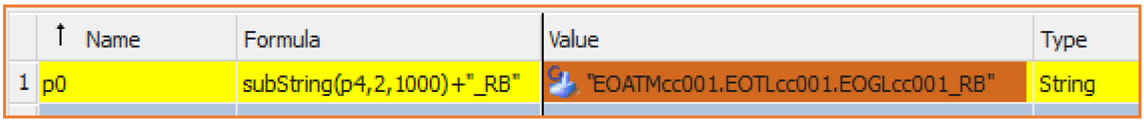
9. Complete the expression.



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

11b_08

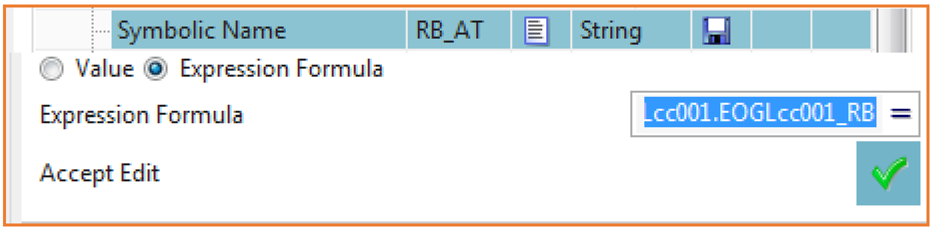
10. Click **OK**. The following shows the result.



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

11b_09

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



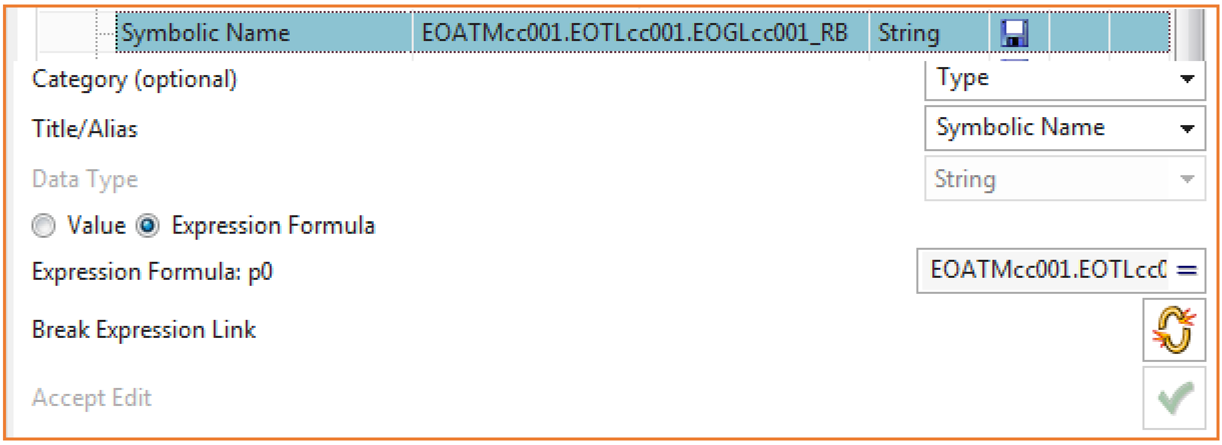
Symbolic Name | RB_AT | String

○ Value ● Expression Formula

Expression Formula: Lcc001.EOGLcc001_RB =

11b_10

12. Click the green arrow.



Symbolic Name | EOATMcc001.EOTLcc001.EOGLcc001_RB | String

Category (optional): Type

Title/Alias: Symbolic Name

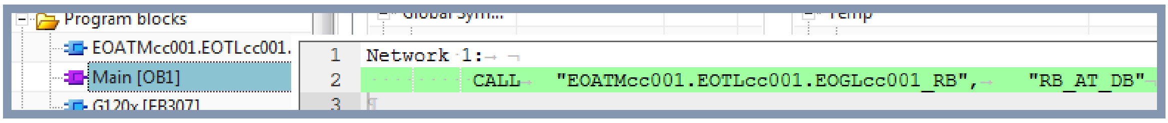
Data Type: String

○ Value ● Expression Formula

Expression Formula: p0: EOATMcc001.EOTLcc(=

11b_11

13. Click **OK**. RB_AT FB now has a globally unique name.



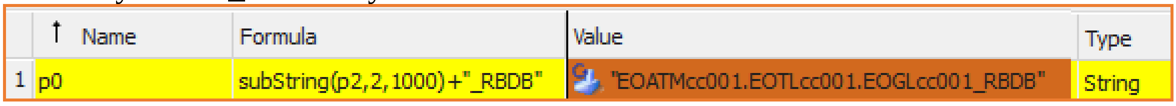
```
Network 1:
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "RB_AT_DB"
```

11b_12

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.

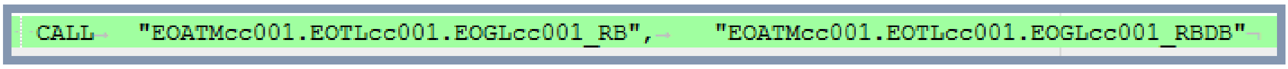
1. Modify the RB_AT IDB symbolic name.



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

11b_13

The following shows the result.

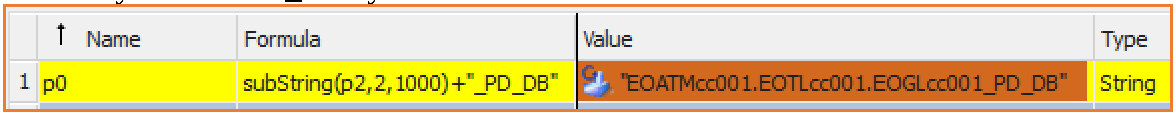


```
CALL  "EOATMcc001.EOTLcc001.EOGLcc001_RB",   "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

11b_14

11.1.3. PosDev_DB symbolic name

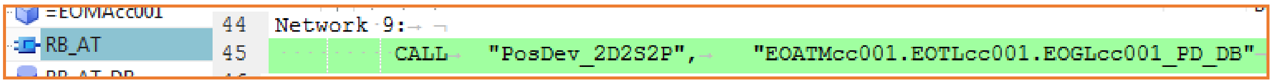
1. Modify the PosDev_DB symbolic name.



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

11b_15

The following shows the result.



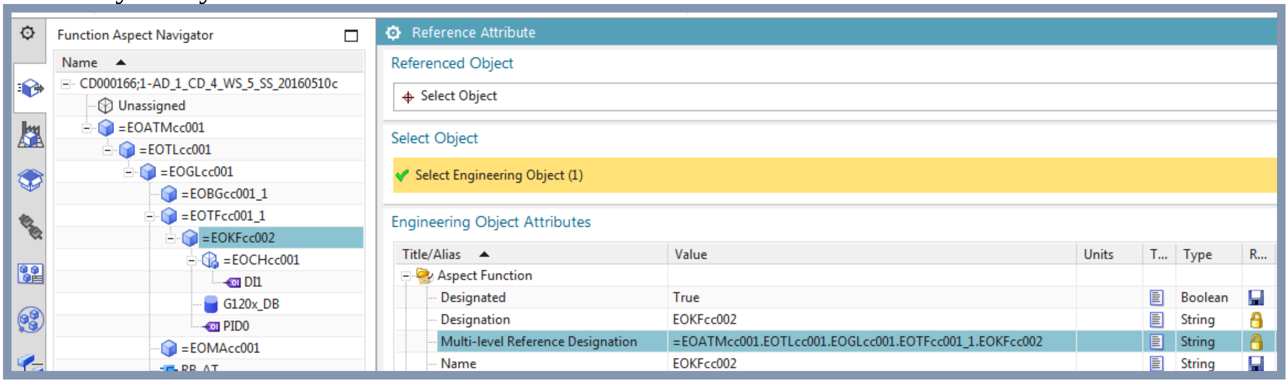
```
44   Network 9:
45         CALL   "PosDev_2D2S2P",   "EOATMcc001.EOTLcc001.EOGLcc001_PD_DB"
```

11b_16

11.1.4. G120 IDB symbolic name

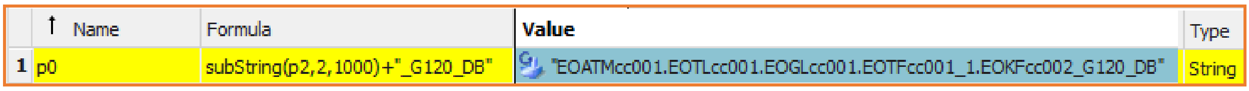
The RB_AT IDB must have a symbolic name that is unique when a template is instantiated. You do this by using an expression to assign a name based on the Function aspect Multi Reference Designation of RB_AT IDB.

1. Modify the symbolic name.



Title/Alias	Value	Units	T...	Type	R...
Aspect Function					
Designated	True			Boolean	
Designation	EOKFcc002			String	
Multi-level Reference Designation	=EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002			String	
Name	EOKFcc002			String	

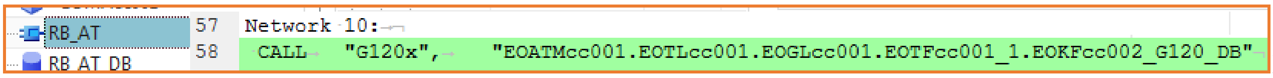
11b_17



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

11b_18

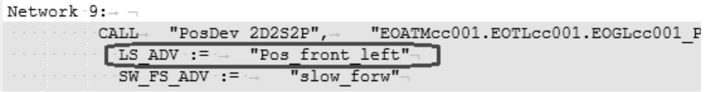
The following shows the result.



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

11b_19

>>>>>>>>>>>>>>>> TERRY… did I forgot to do this previously???

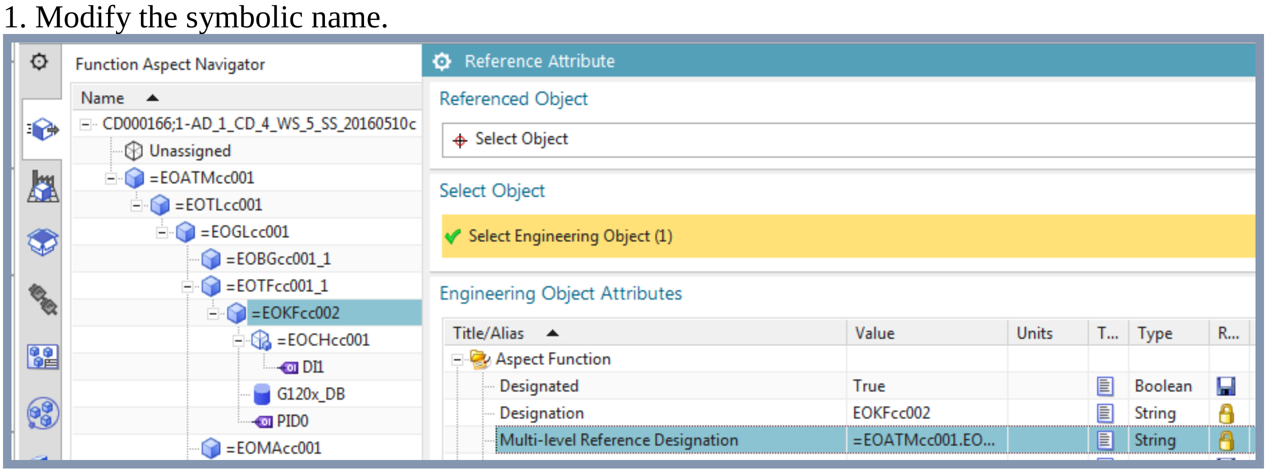


```
Network 9:
        CALL   "PosDev 2D2S2P",   "EOATMcc001.EOTLcc001.EOGLcc001_P
          LS_ADV :=    "Pos_front_left"
          SW_FS_ADV :=    "slow_forw"
```

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.



Function Aspect Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_SS_20160510c
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
          - =EOKFcc002
            - =EOCHcc001
              - DI1
            - G120x_DB
            - PID0
        - =EOMAcc001

Reference Attribute

Referenced Object

Select Object

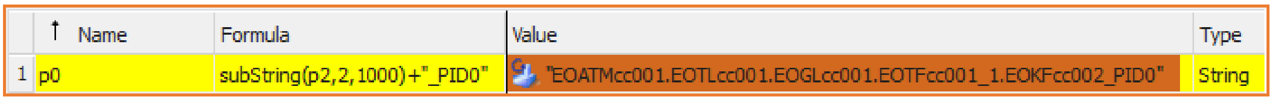
Select Object

✓ Select Engineering Object (1)

Engineering Object Attributes

Title/Alias	Value	Units	T...	Type	R...
Aspect Function					
Designated	True			Boolean	
Designation	EOKFcc002			String	
Multi-level Reference Designation	=EOATMcc001.EO...			String	

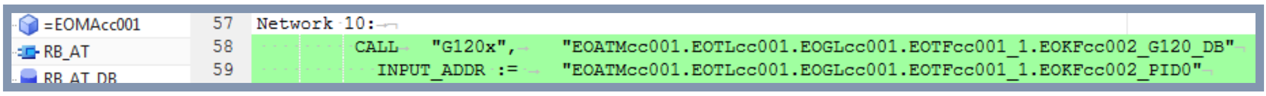
11b_20



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

11b_21

The following shows the result.



=EOMAcc001
RB_AT
RB_AT_DB

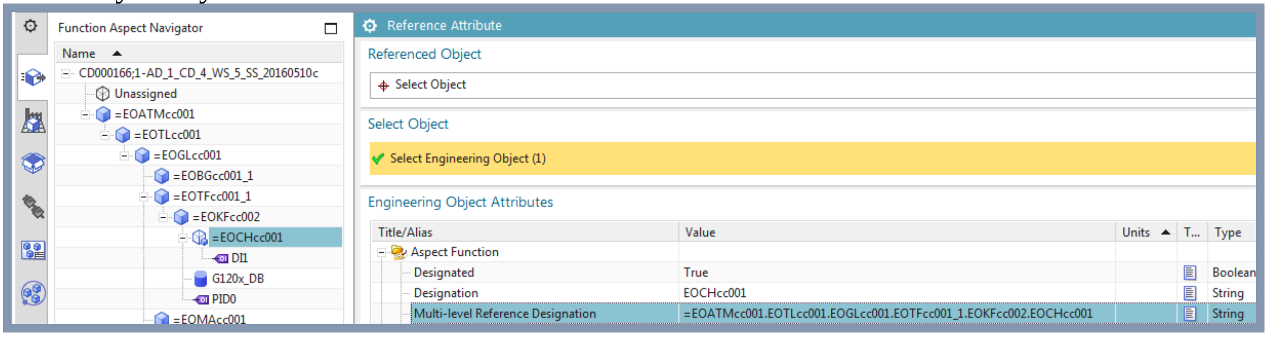
```
Network 10:
      CALL   "G120x",      "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_G120_DB"
        INPUT_ADDR :=      "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_PID0"
```

11b_22

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.



Function Aspect Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_SS_20160510c
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
          - =EOKFcc002
            - =EOCHcc001
              - DI1
            - G120x_DB
            - PID0
        - =EOMAcc001

Reference Attribute

Referenced Object

Select Object

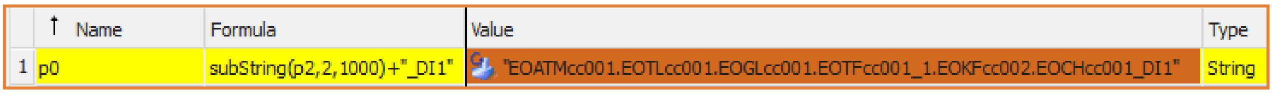
Select Object

✓ Select Engineering Object (1)

Engineering Object Attributes

Title/Alias	Value	Units	T...	Type
Aspect Function				
Designated	True			Boolean
Designation	EOCHcc001			String
Multi-level Reference Designation	=EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002.EOCHcc001			String

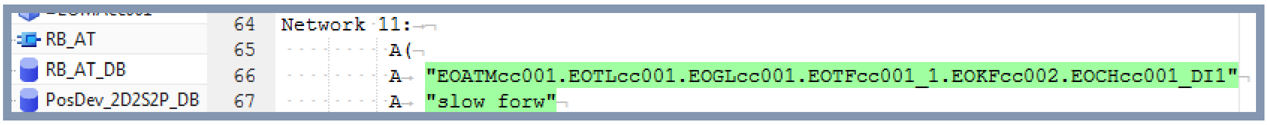
11b_23



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

11b_24

The following shows the result.



RB_AT
RB_AT_DB
PosDev_2D2S2P_DB

```
Network 11:
        A(
        A   "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002.EOCHcc001_DI1"
        A   "slow forw"
```

11b_25

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.

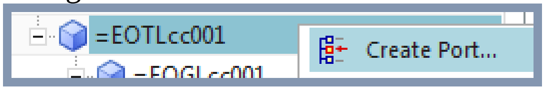
A *dynamic connection* is a connection that you link to a navigation expression. The navigation expression returns the target port.

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.

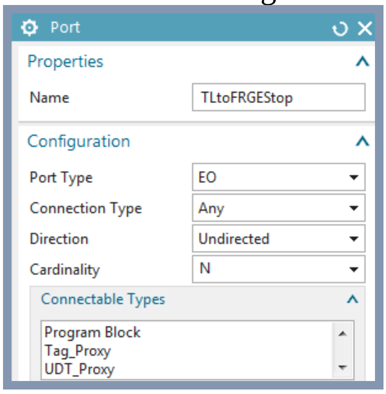
First create a port from TL to the tag.

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



11b_26

2.Enter the following.


Port

Properties

Name: TLtoFRGEStop

Configuration

Port Type: EO

Connection Type: Any

Direction: Undirected

Cardinality: N

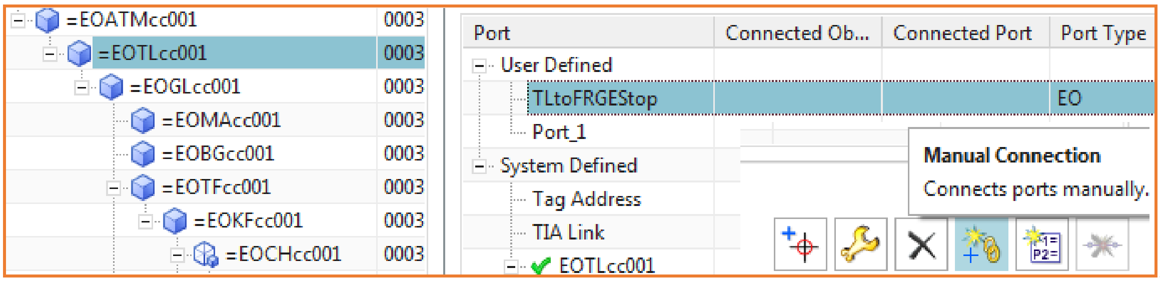
Connectable Types

Program Block
Tag_Proxy
UDT_Proxy

11b_27

3. Click **OK**. The port is created.

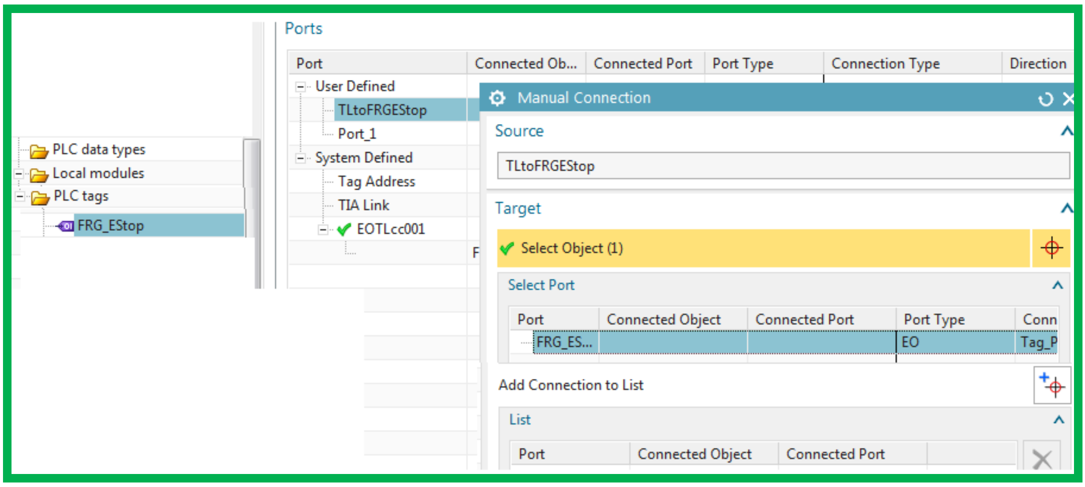
4. Create a manual connection from TL to the tag. The tag does not require a port. Note that a manual connection is good enough, since TL is not in the template.


=EOATMcc001 0003
=EOTLcc001 0003
=EOGLcc001 0003
=EOMAcc001 0003
=EOBGcc001 0003
=EOTFcc001 0003
=EOKFcc001 0003
=EOCHcc001 0003

Port	Connected Ob...	Connected Port	Port Type
User Defined			
TLtoFRGEStop			EO
Port_1			
System Defined			
Tag Address			
TIA Link			
EOTLcc001			

Manual Connection
Connects ports manually.

11b_28



Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction
User Defined					
TLtoFRGEStop					
Port_1					
System Defined					
Tag Address					
TIA Link					
EOTLcc001					

Source

TLtoFRGEStop

Target

✓ Select Object (1)

Select Port

Port	Connected Object	Connected Port	Port Type	Conn
FRG_ES...			EO	Tag_P

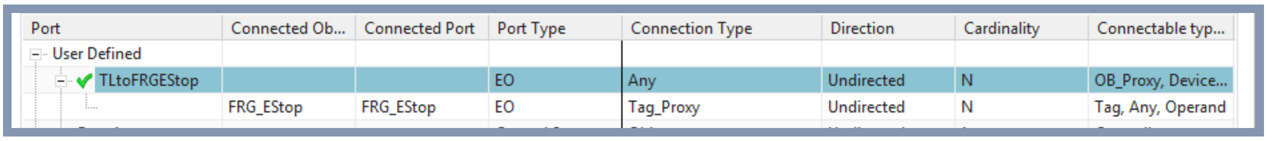
Add Connection to List

List

Port	Connected Object	Connected Port

11b_29

The following shows the result.

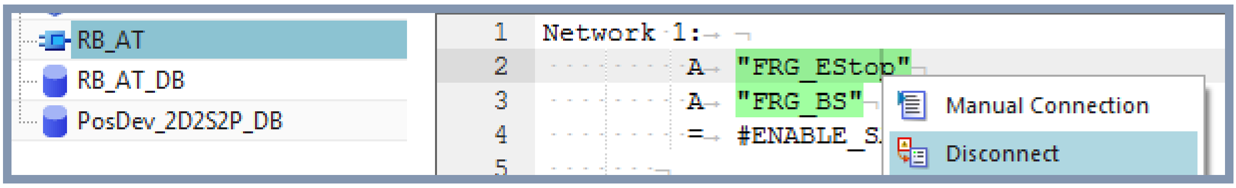


Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable typ...
User Defined							
✓ TLtoFRGEStop			EO	Any	Undirected	N	OB_Proxy, Device...
	FRG_EStop	FRG_EStop	EO	Tag_Proxy	Undirected	N	Tag, Any, Operand

11b_30

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

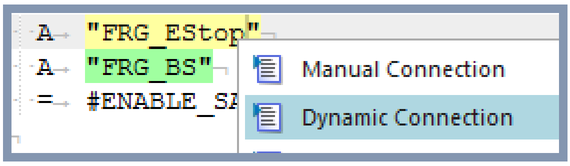
5. Disconnect the manual connection.



```
Network 1:
      A    "FRG_EStop"
      A    "FRG_BS"
      =    #ENABLE_S
```

11b_31

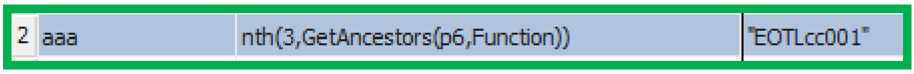
6. Create a dynamic connection.



```
A    "FRG_EStop"
A    "FRG_BS"
=    #ENABLE_SA
```

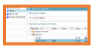
11b_32

7. Add an expression to get TL (P6 is RB_AT FB object name).

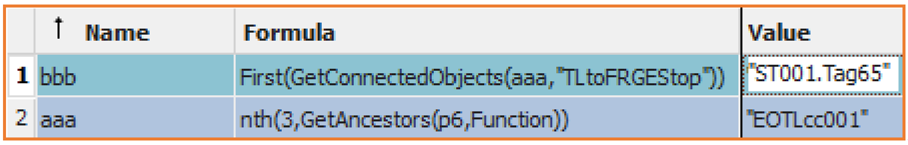


2	aaa	nth(3,GetAncestors(p6,Function))	"EOTLcc001"

11b_33

11b_34

8. Add an expression to get the tag.

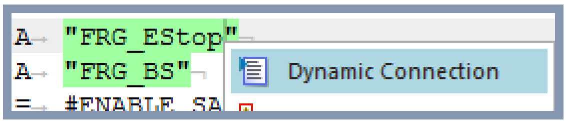


	↑ Name	Formula	Value
1	bbb	First(GetConnectedObjects(aaa,"TLtoFRGEStop"))	"ST001.Tag65"
2	aaa	nth(3,GetAncestors(p6,Function))	"EOTLcc001"

11b_35 xxxxxx

Note that value is blank. This is a bug.

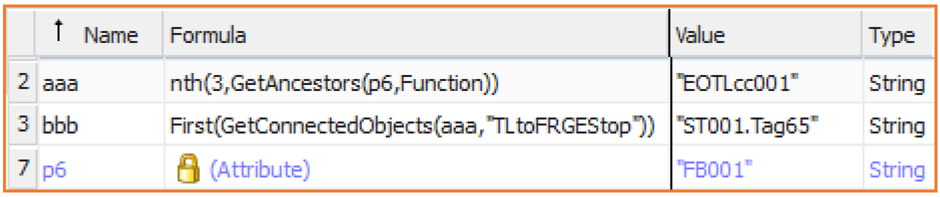
9. Click **OK**. FRG_EStop is green because the connection was established.



```
A    "FRG_EStop"
A    "FRG_BS"
=    #ENABLE_SA
```

11b_36

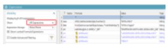
Note also that the expression not show. another nasty confusing bug.



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

11b_37xxxxxx

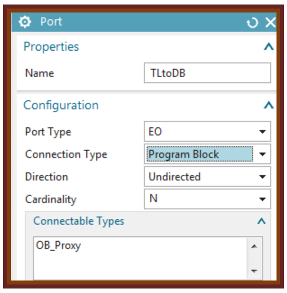
10. Show all.

11b_38

11.3. OB Main calls

Now you need to enable OB Main to automatically add a call to an added GL. The following is my guess at how to do this, Not sure if it's the official way.

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



Port

Properties

Name: TLtoDB

Configuration

Port Type: EO

Connection Type: Program Block

Direction: Undirected

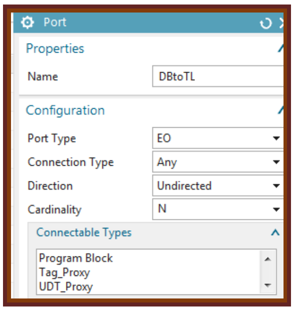
Cardinality: N

Connectable Types

OB_Proxy

11b_39

2. Create a port on RBAT IDB.



Port

Properties

Name: DBtoTL

Configuration

Port Type: EO

Connection Type: Any

Direction: Undirected

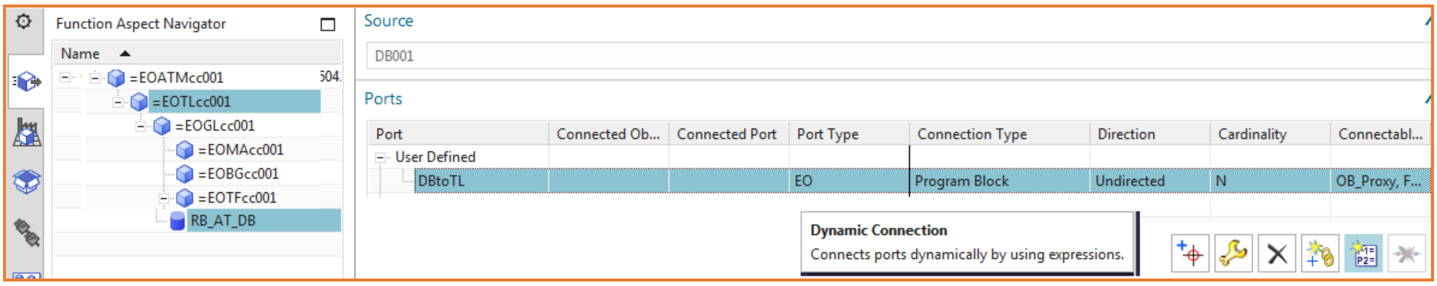
Cardinality: N

Connectable Types

Program Block
Tag_Proxy
UDT_Proxy

11b_40

3. Create a dynamic connection by doing a **getPort** from RB_AT IDB (p6 = conveyor **Object name**).



Function Aspect Navigator

Name

- =EOATMcc001 504.
  - =EOTLcc001
    - =EOGLcc001
      - =EOMAcc001
      - =EOBGcc001
      - =EOTFcc001
        - RB_AT_DB

Source

DB001

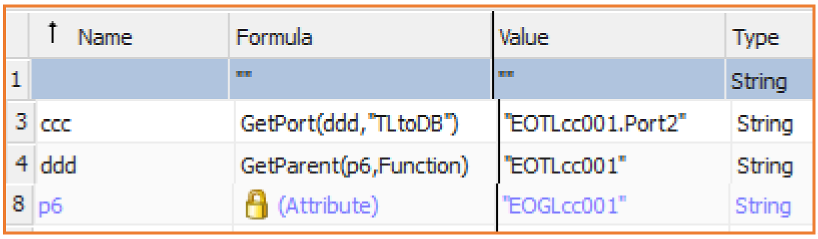
Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectabl...
User Defined							
DBtoTL			EO	Program Block	Undirected	N	OB_Proxy, F...

Dynamic Connection

Connects ports dynamically by using expressions.

11b_41 xxxxxx

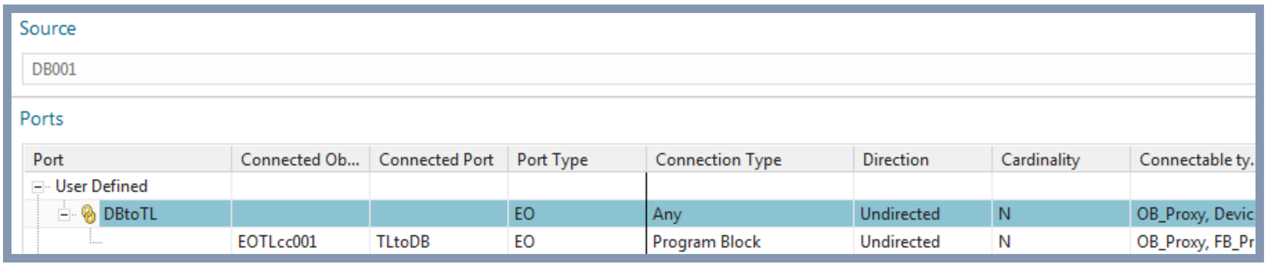


	Name	Formula	Value	Type
1		""	""	String
3	ccc	GetPort(ddd,"TLtoDB")	"EOTLcc001.Port2"	String
4	ddd	GetParent(p6,Function)	"EOTLcc001"	String
8	p6	(Attribute)	"EOGLcc001"	String

11b_42

11b_43

The following shows the result.



Source

DB001

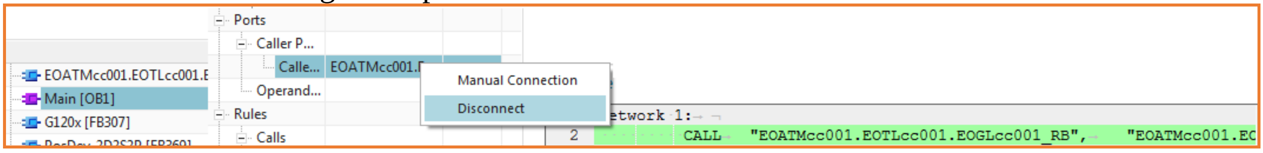
Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable ty.
User Defined							
DBtoTL			EO	Any	Undirected	N	OB_Proxy, Devic
	EOTLcc001	TLtoDB	EO	Program Block	Undirected	N	OB_Proxy, FB_Pr

11b_44

Now change the OB main to RBAT DB call port.

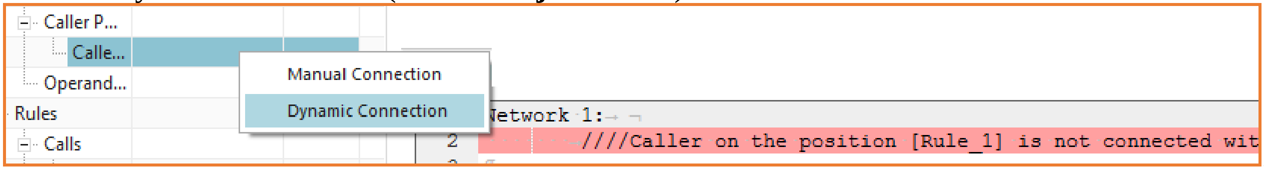
4. Disconnect the existing caller port.



```
etwork 1:
2     CALL  "EOATMcc001.EOTLcc001.EOGLcc001_RB",   "EOATMcc001.EC
```

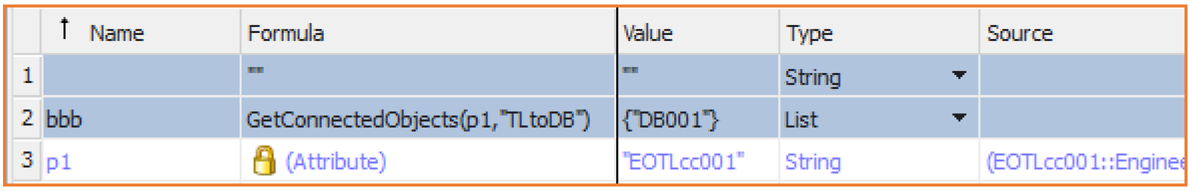
11b_45

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



```
Network 1:
2     ////Caller on the position [Rule_1] is not connected wit
```

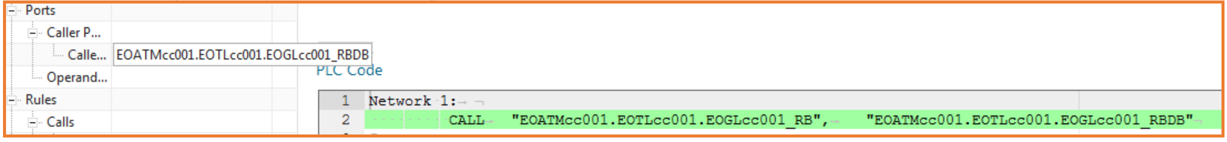
11b_46



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

11b_47

The following shows the resulting OB main call.



```
1   Network 1:
2       CALL  "EOATMcc001.EOTLcc001.EOGLcc001_RB",   "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

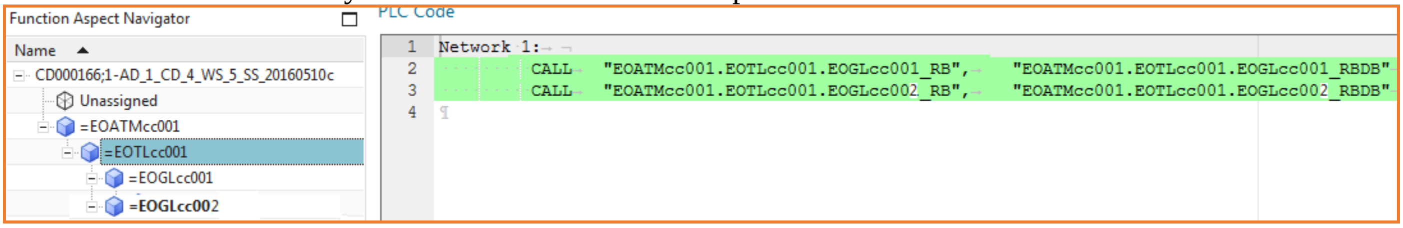
11b_48

11.4. Generate

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

TERRY: Several major errors. To finish this Getting Started, probably just have to fake things.

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.



Function Aspect Navigator
Name
CD000166;1-AD_1_CD_4_WS_5_SS_20160510c
Unassigned
=EOATMcc001
=EOTLcc001
=EOGLcc001
=EOGLcc002
PLC Code

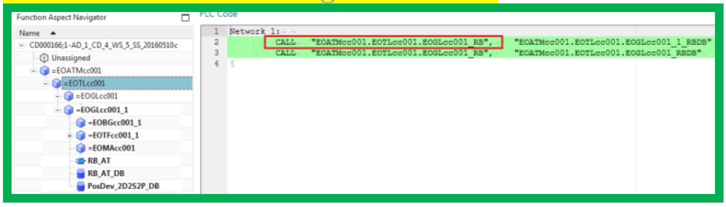
```
Network 1:
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",   "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc002_RB",   "EOATMcc001.EOTLcc001.EOGLcc002_RBDB"
```

11b_49 xxxxx

11.2.1. OB main bug

1. Right-click on GL.
2. select copy.
3. Right-click on TL.
4. Select Paste.

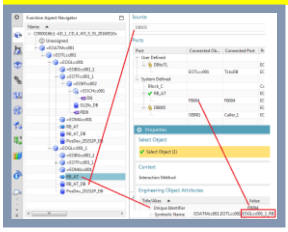
Note the error. The wrong FB is listed.



```
Network 1:
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",   "EOATMcc001.EOTLcc001.EOGLcc001_1_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",   "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

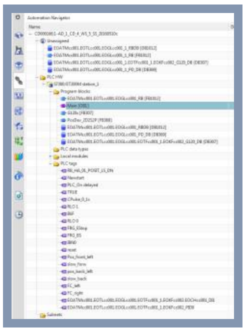
11b_49

But its actually linked to the correct FB. You can verify this by looking at the IDB ports. Connects to the correct FB. The name is simply not being updated.



11b_50

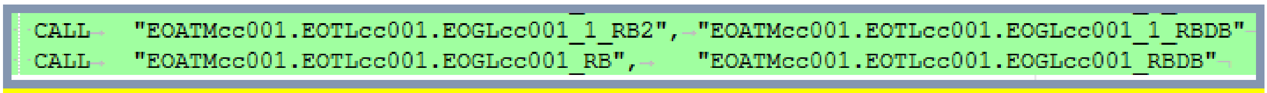
5. connect software (left: before, right: after) No effect.



11b_51

11b_52

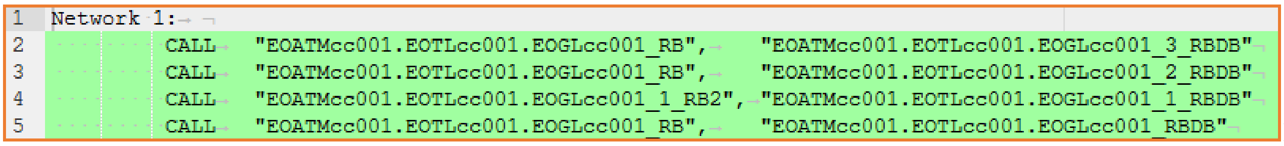
6. change the FB symname (add “2” on the end). This causes an update.



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

11b_53

7. Copy 2 more GL’s. same problem. Talked with Amir, it’s a bug.



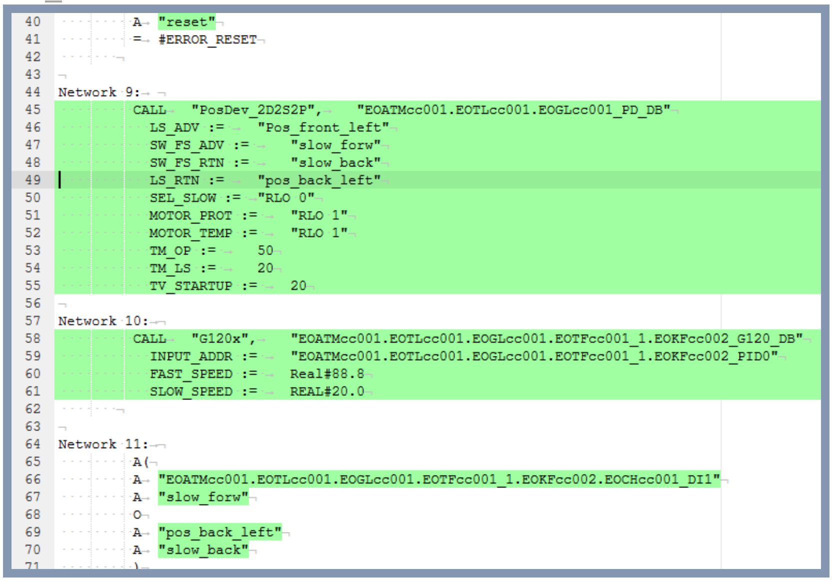
```
Network 1:
        CALL    "EOATMcc001.EOTLcc001.EOGLcc001_RB",      "EOATMcc001.EOTLcc001.EOGLcc001_3_RBDB"
        CALL    "EOATMcc001.EOTLcc001.EOGLcc001_RB",      "EOATMcc001.EOTLcc001.EOGLcc001_2_RBDB"
        CALL    "EOATMcc001.EOTLcc001.EOGLcc001_1_RB2",  "EOATMcc001.EOTLcc001.EOGLcc001_1_RBDB"
        CALL    "EOATMcc001.EOTLcc001.EOGLcc001_RB",      "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

11b_54

11.2.2. RB_AT mistakes

conveyors 0 and 1 have been connnected software. 2 and 3 have not. Note the differences below. Amir says this is not an error. Seems strange to me.

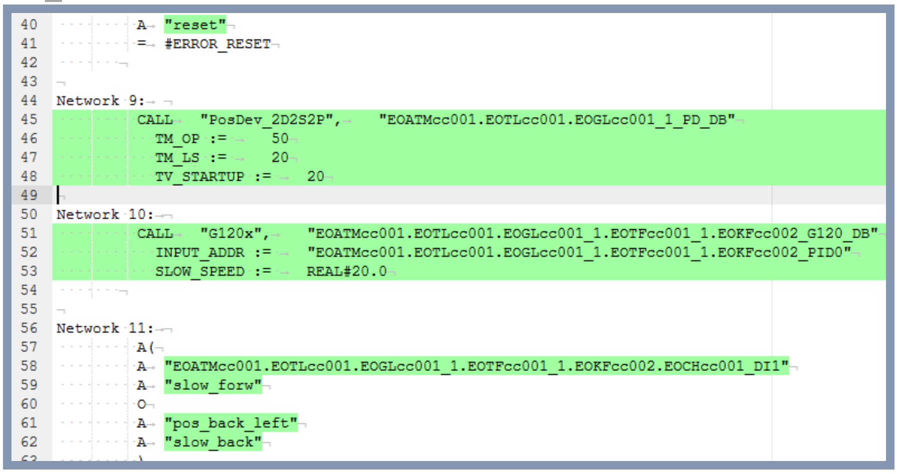
RB_AT 0



```
        A  "reset"
        =  #ERROR_RESET

Network 9:
        CALL  "PosDev_2D2S2P",   "EOATMcc001.EOTLcc001.EOGLcc001_PD_DB"
          LS_ADV :=    "Pos_front_left"
          SW_FS_ADV :=    "slow_forw"
          SW_FS_RTN :=    "slow_back"
          LS_RTN :=    "pos_back_left"
          SEL_SLOW :=  "RLO 0"
          MOTOR_PROT :=    "RLO 1"
          MOTOR_TEMP :=    "RLO 1"
          TM_OP :=    50
          TM_LS :=    20
          TV_STARTUP :=    20

Network 10:
        CALL  "G120x",    "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_G120_DB"
          INPUT_ADDR :=    "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002_PID0"
          FAST_SPEED :=    Real#88.8
          SLOW_SPEED :=    REAL#20.0


Network 11:
        A(
        A  "EOATMcc001.EOTLcc001.EOGLcc001.EOTFcc001_1.EOKFcc002.EOCHcc001_DI1"
        A  "slow_forw"
        O
        A  "pos_back_left"
        A  "slow_back"
        )
```

11b_55

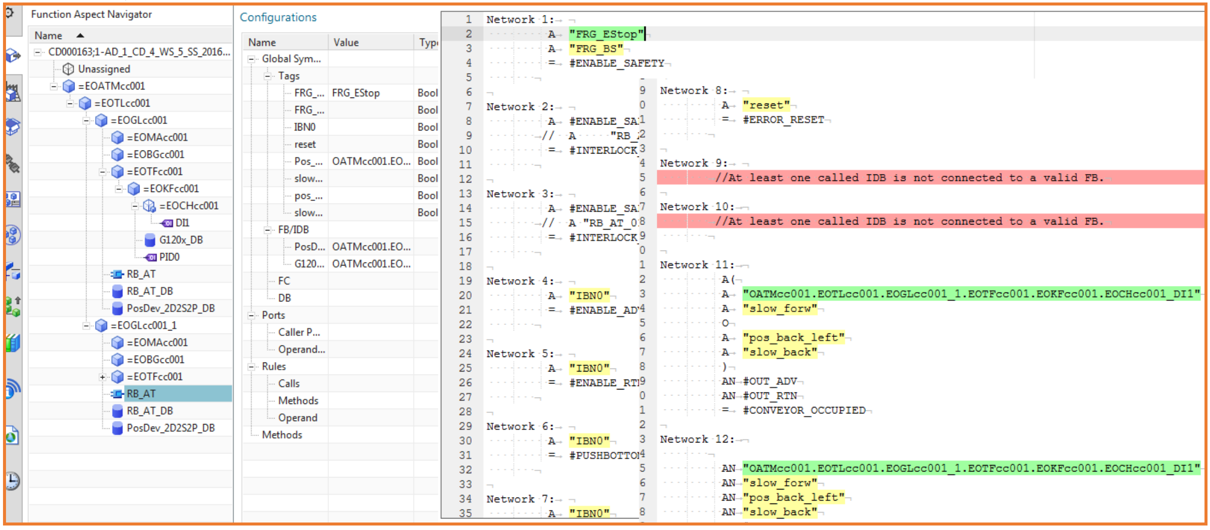
RB_AT 1



```
        A  "reset"
        =  #ERROR_RESET

Network 9:
        CALL  "PosDev_2D2S2P",    "EOATMcc001.EOTLcc001.EOGLcc001_1_PD_DB"
          TM_OP :=    50
          TM_LS :=    20
          TV_STARTUP :=    20

Network 10:
        CALL  "G120x",    "EOATMcc001.EOTLcc001.EOGLcc001_1.EOTFcc001_1.EOKFcc002_G120_DB"
          INPUT_ADDR :=    "EOATMcc001.EOTLcc001.EOGLcc001_1.EOTFcc001_1.EOKFcc002_PID0"
          SLOW_SPEED :=    REAL#20.0


Network 11:
        A(
        A  "EOATMcc001.EOTLcc001.EOGLcc001_1.EOTFcc001_1.EOKFcc002.EOCHcc001_DI1"
        A  "slow_forw"
        O
        A  "pos_back_left"
        A  "slow_back"
        )
```

11b_56

RB_AT 2,3



Name	Value	Typ
Global Sym...		
Tags		
FRG_...	FRG_EStop	Bool
FRG_...		Bool
IBN0		Bool
reset		Bool
Pos_...	OATMcc001.EO...	Bool
slow...		Bool
pos_...		Bool
slow...		Bool
FB/IDB		
PosD...	OATMcc001.EO...	
G120...	OATMcc001.EO...	
FC		
DB		
Ports		
Caller P...		
Operand...		
Rules		
Calls		
Methods		
Operand		
Methods		

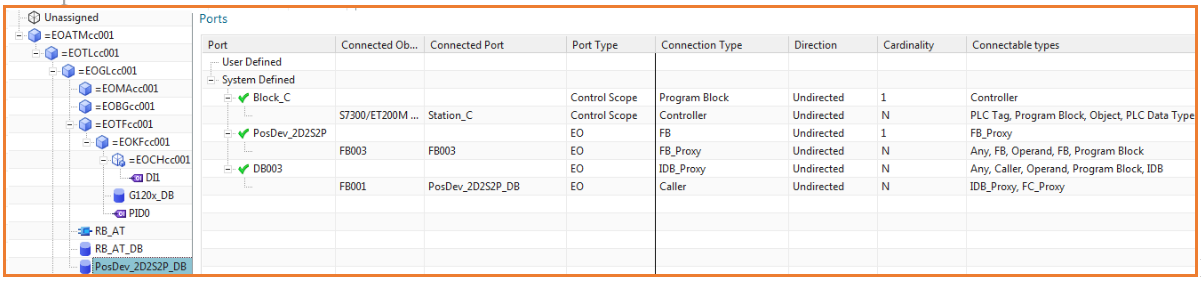
```
Network 1:
        A  "FRG_EStop"
        A  "FRG_BS"
        =  #ENABLE_SAFETY


Network 2:
        A  #ENABLE_SA
      //  A     "RB_
        =  #INTERLOCK


Network 3:
        A  #ENABLE_SA
      //  A "RB_AT_0
        =  #INTERLOCK


Network 4:
        A  "IBN0"
        =  #ENABLE_AD


Network 5:
        A  "IBN0"
        =  #ENABLE_RT


Network 6:
        A  "IBN0"
        =  #PUSHBOTTO


Network 7:
        A  "IBN0"
```

```
Network 8:
        A  "reset"
        =  #ERROR_RESET


Network 9:
        //At least one called IDB is not connected to a valid FB.

Network 10:
        //At least one called IDB is not connected to a valid FB.


Network 11:
        A(
        A  "OATMcc001.EOTLcc001.EOGLcc001_1.EOTFcc001.EOKFcc001.EOCHcc001_DI1"
        A  "slow_forw"
        O
        A  "pos_back_left"
        A  "slow_back"
        )
        AN #OUT_ADV
        AN #OUT_RTN
        =  #CONVEYOR_OCCUPIED

Network 12:

        AN "OATMcc001.EOTLcc001.EOGLcc001_1.EOTFcc001.EOKFcc001.EOCHcc001_DI1"
        AN "slow_forw"
        AN "pos_back_left"
        AN "slow_back"
```

11b_57

Following shows how I previously manually fixed this.

The ports.

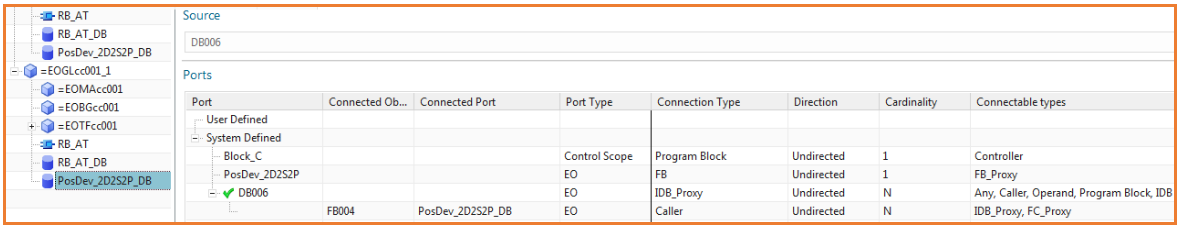


Unassigned
=EOATMcc001
=EOTLcc001
=EOGLcc001
=EOMAcc001
=EOBGcc001
=EOTFcc001
=EOKFcc001
=EOCHcc001
DI1
G120x_DB
PID0
RB_AT
RB_AT_DB
PosDev_2D2S2P_DB

Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
System Defined							
✓ Block_C			Control Scope	Program Block	Undirected	1	Controller
	S7300/ET200M ...	Station_C	Control Scope	Controller	Undirected	N	PLC Tag, Program Block, Object, PLC Data Type
✓ PosDev_2D2S2P			EO	FB	Undirected	1	FB_Proxy
	FB003	FB003	EO	FB_Proxy	Undirected	N	Any, FB, Operand, FB, Program Block
✓ DB003			EO	IDB_Proxy	Undirected	N	Any, Caller, Operand, Program Block, IDB
	FB001	PosDev_2D2S2P_DB	EO	Caller	Undirected	N	IDB_Proxy, FC_Proxy

11b_58



RB_AT
RB_AT_DB
PosDev_2D2S2P_DB
=EOGLcc001_1
=EOMAcc001
=EOBGcc001
=EOTFcc001
RB_AT
RB_AT_DB
PosDev_2D2S2P_DB

Source

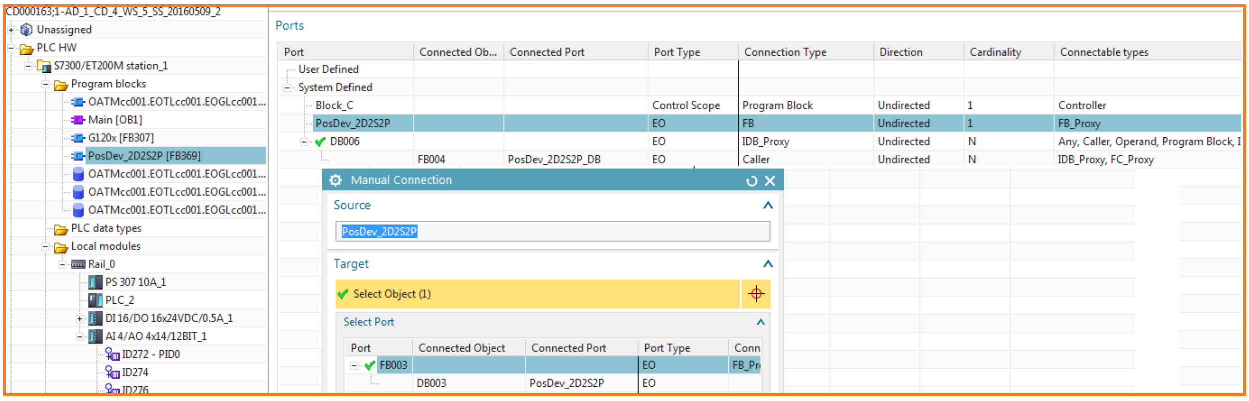
DB006

Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
System Defined							
Block_C			Control Scope	Program Block	Undirected	1	Controller
PosDev_2D2S2P			EO	FB	Undirected	1	FB_Proxy
✓ DB006			EO	IDB_Proxy	Undirected	N	Any, Caller, Operand, Program Block, IDB
	FB004	PosDev_2D2S2P_DB	EO	Caller	Undirected	N	IDB_Proxy, FC_Proxy

11b_59

1. manually connect the ports.



CD000163;1-AD_1_CD_4_WS_5_SS_20160509_2
Unassigned
PLC HW
S7300/ET200M station_1
Program blocks
OATMcc001.EOTLcc001.EOGLcc001...
Main [OB1]
G120x [FB307]
PosDev_2D2S2P [FB369]
OATMcc001.EOTLcc001.EOGLcc001...
OATMcc001.EOTLcc001.EOGLcc001...
OATMcc001.EOTLcc001.EOGLcc001...
PLC data types
Local modules
Rail_0
PS 307 10A_1
PLC_2
DI 16/DO 16x24VDC/0.5A_1
AI 4/AO 4x14/12BIT_1
ID272 - PID0
ID274
ID276

Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
System Defined							
Block_C			Control Scope	Program Block	Undirected	1	Controller
PosDev_2D2S2P			EO	FB	Undirected	1	FB_Proxy
✓ DB006			EO	IDB_Proxy	Undirected	N	Any, Caller, Operand, Program Block, I
	FB004	PosDev_2D2S2P_DB	EO	Caller	Undirected	N	IDB_Proxy, FC_Proxy

Manual Connection

Source

PosDev_2D2S2P

Target

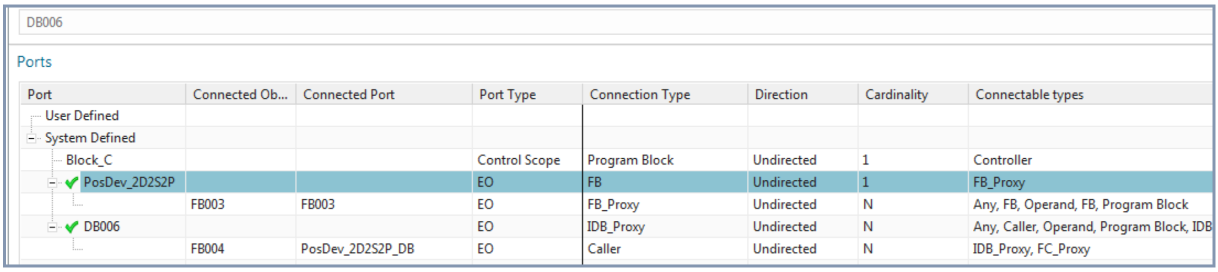
✓ Select Object (1)

Select Port

Port	Connected Object	Connected Port	Port Type	Conn
✓ FB003			EO	FB_Pr
	DB003	PosDev_2D2S2P	EO	

11b_60

Result.

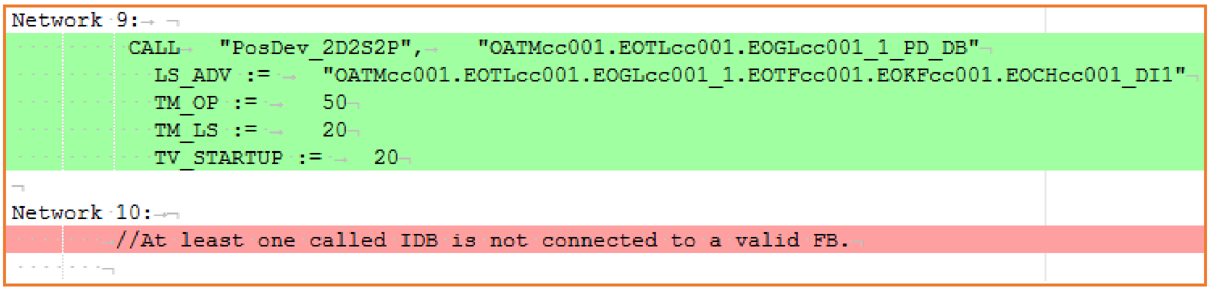


DB006

Ports

Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
System Defined							
Block_C			Control Scope	Program Block	Undirected	1	Controller
✓ PosDev_2D2S2P			EO	FB	Undirected	1	FB_Proxy
	FB003	FB003	EO	FB_Proxy	Undirected	N	Any, FB, Operand, FB, Program Block
✓ DB006			EO	IDB_Proxy	Undirected	N	Any, Caller, Operand, Program Block, IDB
	FB004	PosDev_2D2S2P_DB	EO	Caller	Undirected	N	IDB_Proxy, FC_Proxy

11b_61



```
Network 9:
          CALL  "PosDev_2D2S2P",   "OATMcc001.EOTLcc001.EOGLcc001_1_PD_DB"
            LS_ADV := "OATMcc001.EOTLcc001.EOGLcc001_1.EOTFcc001.EOKFcc001.EOCHcc001_DI1"
            TM_OP :=  50
            TM_LS :=  20
            TV_STARTUP :=  20

Network 10:
        //At least one called IDB is not connected to a valid FB.
```

11b_62

So all that needs to be done is to manually connect the ports. I think this is an error. In any case, something the user will not like.

12. Create/instantiate template

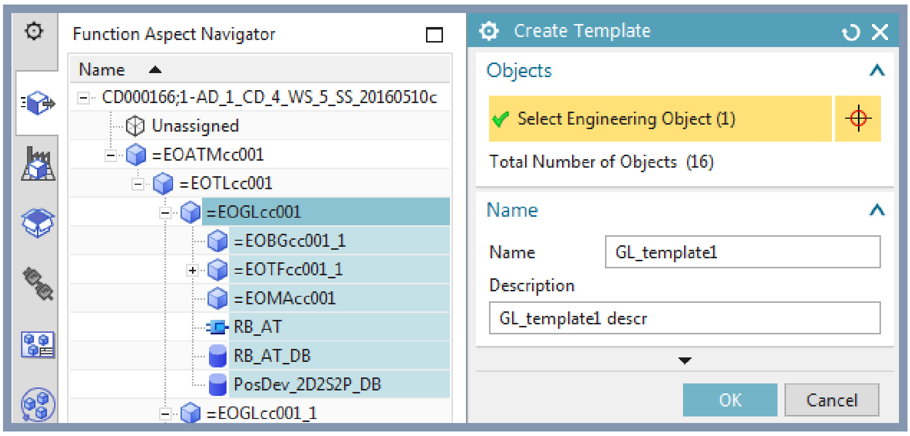
The goal of Automation Designer is to create templates that allow you to easily create software and EPLAN for items in the factory that are very repititious. For example, a factory with 1000 conveyors that are more or less identical. Rather than having to think of unique global names for elements in 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 you 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**.



Function Aspect Navigator

Name

- CD000166;1-AD_1_CD_4_WS_5_SS_20160510c
  - Unassigned
  - =EOATMcc001
    - =EOTLcc001
      - =EOGLcc001
        - =EOBGcc001_1
        - =EOTFcc001_1
        - =EOMAcc001
        - RB_AT
        - RB_AT_DB
        - PosDev_2D2S2P_DB
      - =EOGLcc001_1

Create Template

Objects

✓ Select Engineering Object (1)

Total Number of Objects (16)

Name

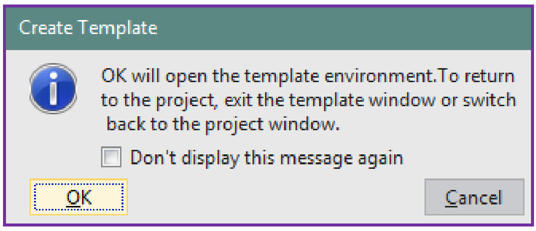
Name: GL_template1

Description: GL_template1 descr

OK | Cancel

12b_01

3. Click **OK**.



Create Template

OK will open the template environment.To return to the project, exit the template window or switch back to the project window.

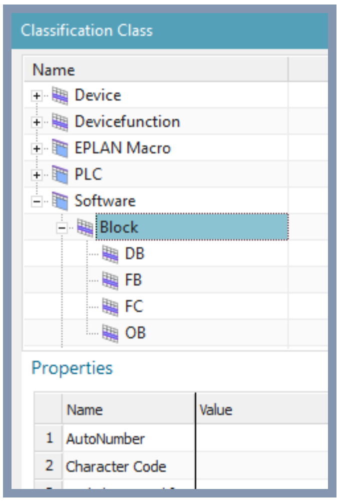
□ Don't display this message again

OK | Cancel

12b_02

4. Click **OK**.

5. For **Classification** select **Software / block**.



Classification Class

Name

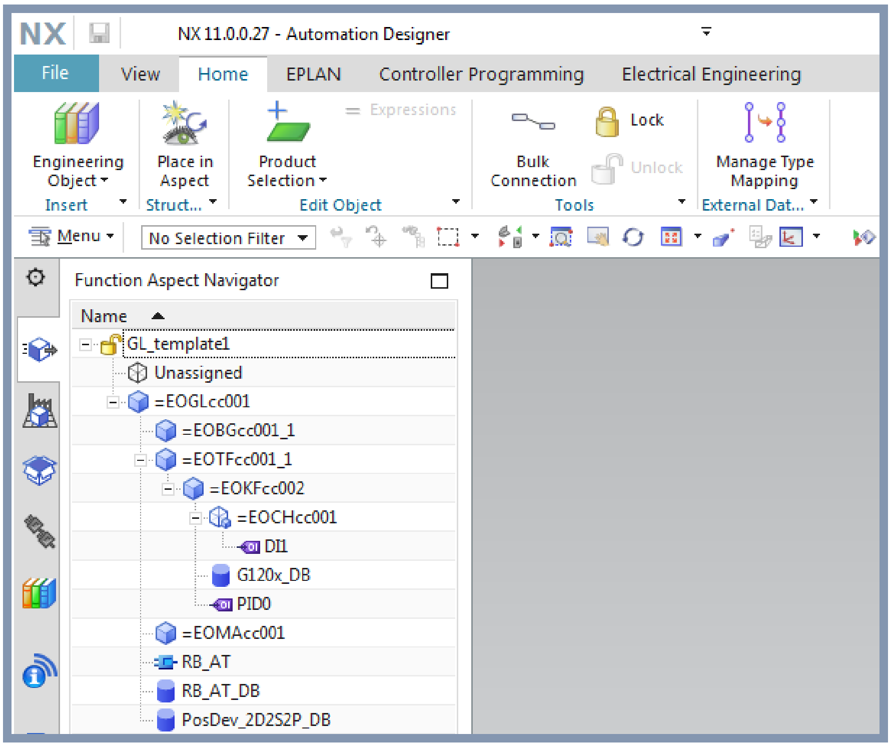
- Device
- Devicefunction
- EPLAN Macro
- PLC
- Software
  - Block
    - DB
    - FB
    - FC
    - OB

Properties

	Name	Value
1	AutoNumber	
2	Character Code	

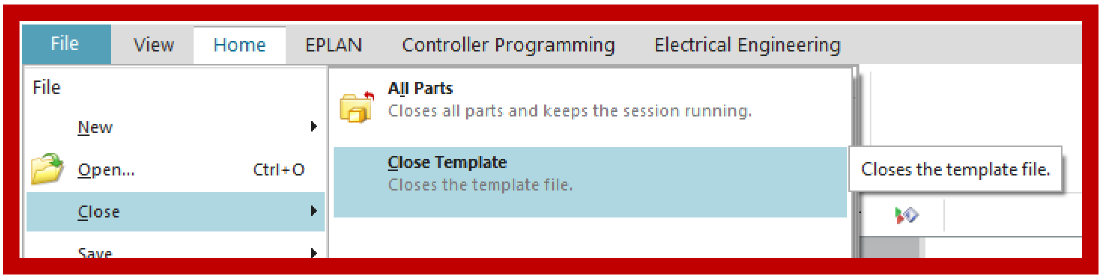
12b_03

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



12b_04

7. Choose **File→Close→Close template**.



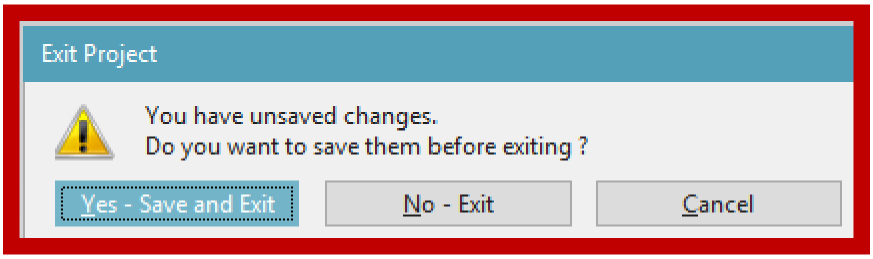
All Parts: Closes all parts and keeps the session running.

Close Template: Closes the template file.

Closes the template file.

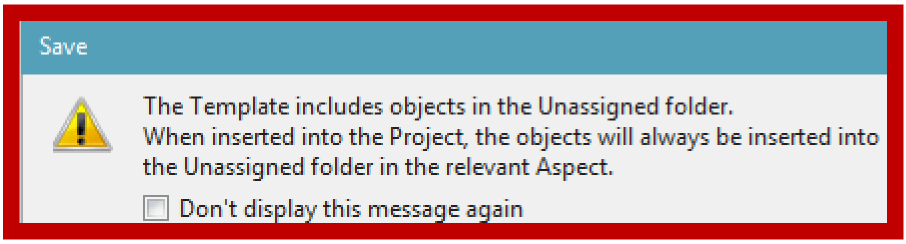
12c_01

8. Choose **Yes – Save and Exit**.


Exit Project

You have unsaved changes.
Do you want to save them before exiting ?

12c_02


Save

The Template includes objects in the Unassigned folder.
When inserted into the Project, the objects will always be inserted into the Unassigned folder in the relevant Aspect.

□ Don't display this message again

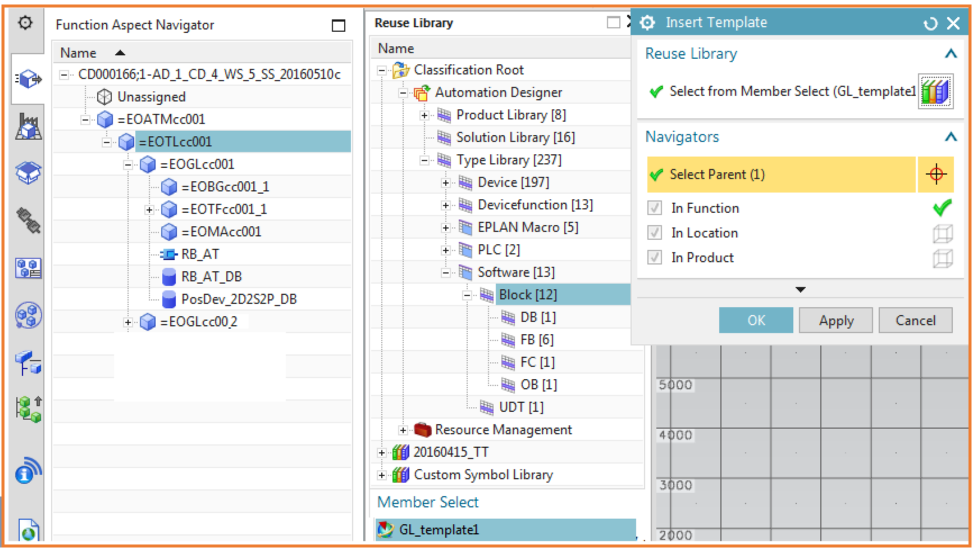
12c_03

12.2. Insert template

This is the goal of Automation Designer for TIA Portal. It doesn’t work.

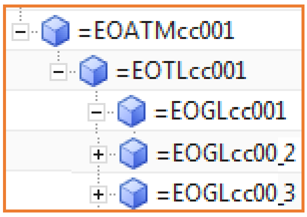
12.2.1. Drag&drop template

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



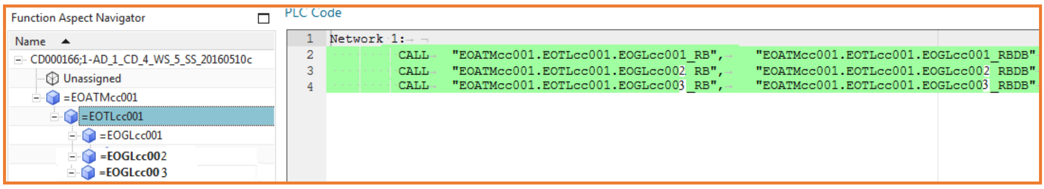
12b_06xxxxx

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.



12b_07xxxxxx

The following shows the result for OB Main.



```
Network 1:
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc002 RB",    "EOATMcc001.EOTLcc001.EOGLcc002 RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc003_RB",    "EOATMcc001.EOTLcc001.EOGLcc003_RBDB"
```

12b_17b

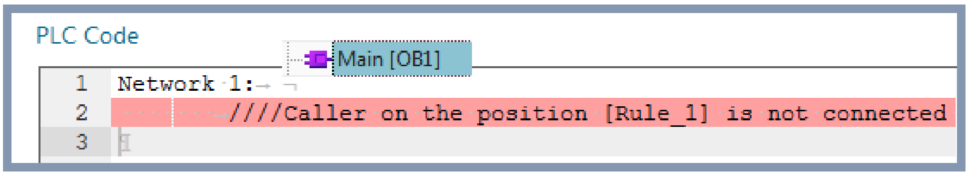
12b_08

RBAT FB:

12b_09

xxx12.2.2. Fix errors

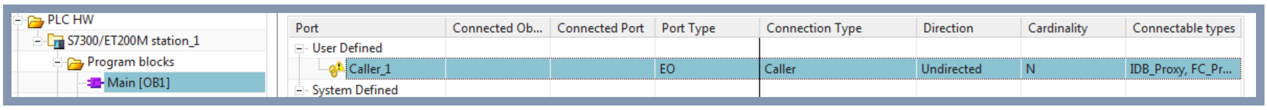
1. Do a bulk connect. OB main still not fixed:



```
Network 1:
    ////Caller on the position [Rule_1] is not connected
```

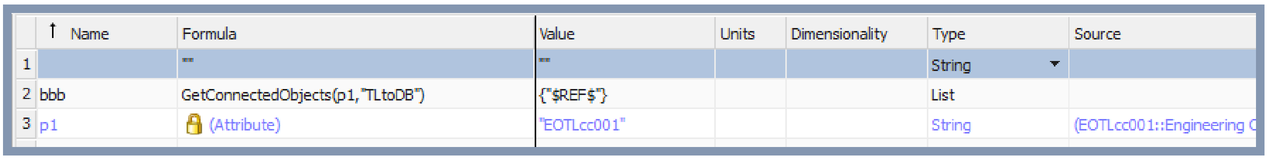
12b_10

2. Open the port



Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
Caller_1			EO	Caller	Undirected	N	IDB_Proxy, FC_Pr...
System Defined							

12b_11

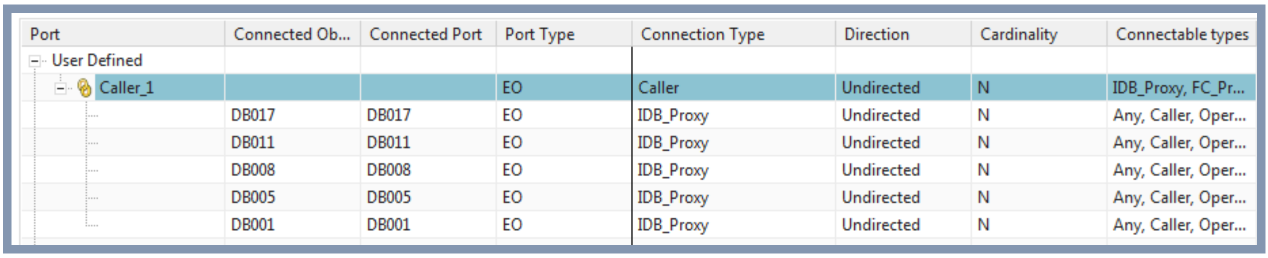


	Name	Formula	Value	Units	Dimensionality	Type	Source
1		""	""			String	
2	bbb	GetConnectedObjects(p1,"TLtoDB")	{"REF"}			List	
3	p1	(Attribute)	"EOTLcc001"			String	(EOTLcc001::Engineering C

12b_12

3. Select bbb.

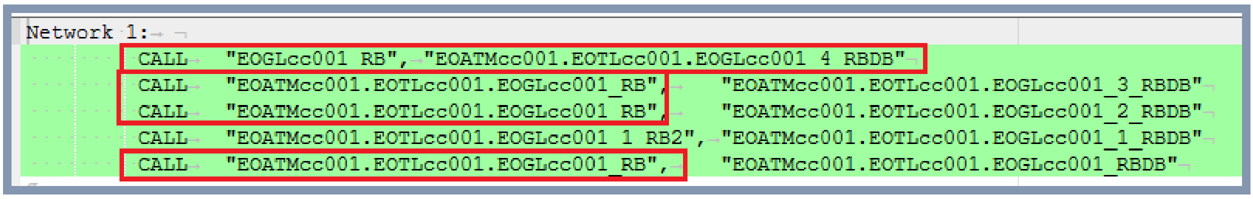
4. Click ok. The 5 conveyors appear.



Port	Connected Ob...	Connected Port	Port Type	Connection Type	Direction	Cardinality	Connectable types
User Defined							
Caller_1			EO	Caller	Undirected	N	IDB_Proxy, FC_Pr...
	DB017	DB017	EO	IDB_Proxy	Undirected	N	Any, Caller, Oper...
	DB011	DB011	EO	IDB_Proxy	Undirected	N	Any, Caller, Oper...
	DB008	DB008	EO	IDB_Proxy	Undirected	N	Any, Caller, Oper...
	DB005	DB005	EO	IDB_Proxy	Undirected	N	Any, Caller, Oper...
	DB001	DB001	EO	IDB_Proxy	Undirected	N	Any, Caller, Oper...

12b_13

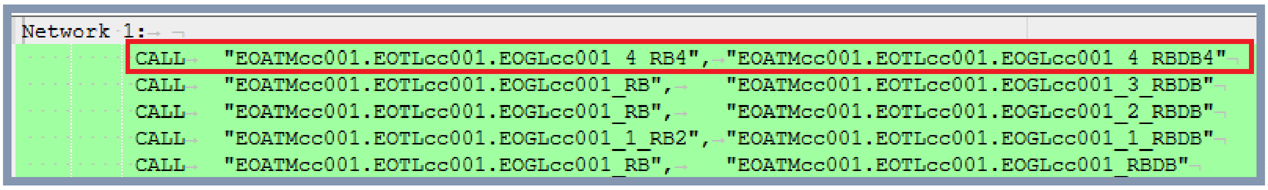
Note the errors in main.



```
Network 1:
        CALL   "EOGLcc001 RB", "EOATMcc001.EOTLcc001.EOGLcc001 4 RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "EOATMcc001.EOTLcc001.EOGLcc001_3_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001 RB",    "EOATMcc001.EOTLcc001.EOGLcc001_2_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001 1 RB2", "EOATMcc001.EOTLcc001.EOGLcc001_1_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

12b_14

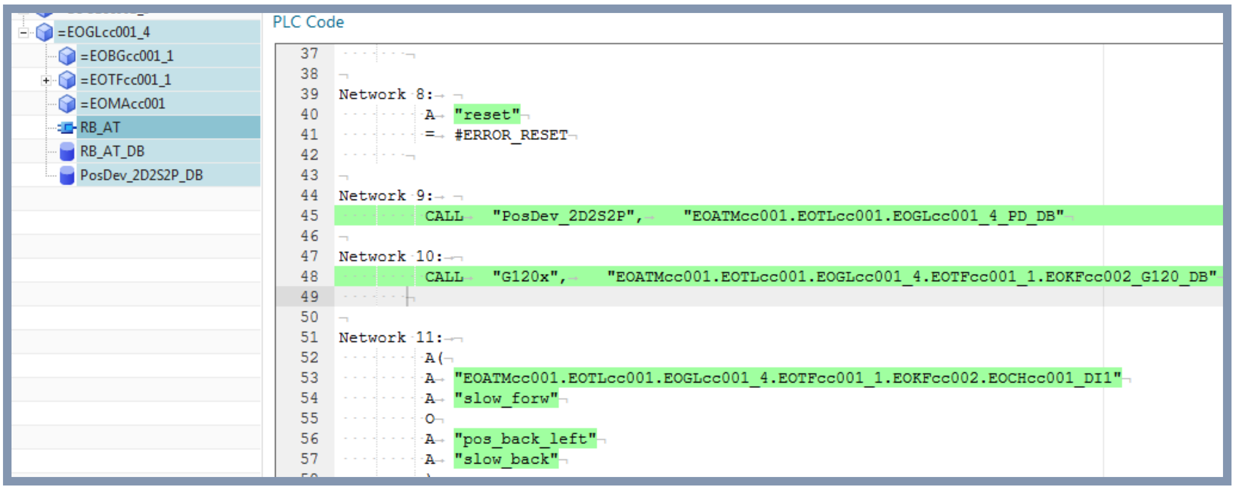
5. Change the RBAT FB symname. Now its updated. The others did not update.



```
Network 1:
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001 4 RB4", "EOATMcc001.EOTLcc001.EOGLcc001 4 RBDB4"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "EOATMcc001.EOTLcc001.EOGLcc001_3_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "EOATMcc001.EOTLcc001.EOGLcc001_2_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_1_RB2", "EOATMcc001.EOTLcc001.EOGLcc001_1_RBDB"
        CALL   "EOATMcc001.EOTLcc001.EOGLcc001_RB",    "EOATMcc001.EOTLcc001.EOGLcc001_RBDB"
```

12b_15

Note in the instantiated template that the RB_AT calls to PosDev and G120x have no variables.



```

Network 8:
        A   "reset"
        =   #ERROR_RESET


Network 9:
        CALL   "PosDev_2D2S2P",    "EOATMcc001.EOTLcc001.EOGLcc001_4_PD_DB"

Network 10:
        CALL   "G120x",    "EOATMcc001.EOTLcc001.EOGLcc001_4.EOTFcc001_1.EOKFcc002_G120_DB"


Network 11:
        A(
        A   "EOATMcc001.EOTLcc001.EOGLcc001_4.EOTFcc001_1.EOKFcc002.EOCHcc001_DI1"
        A   "slow_forw"
        O
        A   "pos_back_left"
        A   "slow_back"
```

12b_16

12.3. Synchronize changes

TERRY: just leave this section as simple text.

If no conveyors are added in Line Designer, the 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**.